The Role of Visual Grounding in Visual Question Answering Generalization and Shortcut Learning

Zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften
des Fachbereich 3
der Universität Bremen

vorgelegte

Dissertation

von

Daniel Reich

Tag des Kolloquiums:	06.06.2024
Erste Gutachterin:	Prof. Dr.-Ing. Tanja Schultz
Zweiter Gutachter:	Prof. Dr.-Ing. Rainer Stiefelhagen

Deutsche Zusammenfassung

Visual Question Answering (VQA) ist die Beantwortung natürlichsprachlicher Fragen zu Bildinhalten. VQA-Modelle verarbeiten zwei Eingabemodalitäten, um eine Antwort zu erzeugen: Vision (das Bild) und Language (die Frage). Die Antwort eines VQA-Modells wird als *Visually Grounded* (VG) bezeichnet, wenn sie auf den für die Frage relevanten Teilen des Bildes basiert. Dieser Prozess wird manchmal intuitiv mit *"right for the right reasons"* beschrieben. Auf konzeptioneller Ebene ist die Notwendigkeit von VG naheliegend und seine Rolle offensichtlich. In der Praxis aber sind VQA-Modelle berüchtigt für ihren Mangel an VG, und das obwohl ihre Leistung sich stetig verbessert. Es stellt sich also die Frage, wie VQA-Modelle ohne VG überhaupt richtig funktionieren können, und welche Rolle VG in VQA tatsächlich einnimmt.

Wie in vielen, wenn nicht gar allen Bereichen des Maschinellen Lernens, basieren moderne Modelle in VQA fast ausschließlich auf Deep Learning (DL). Während DL in allen Bereichen des maschinellen Lernens, inklusive VQA, zu noch nie dagewesenen Leistungen beitragen, sind DL-Modelle mit einer erheblichen Einschränkung konfrontiert: dem sogenannten *Shortcut-Learning*. Ein Shortcut kann als unbeabsichtigter Lösungsweg eines gegebenen Problems beschrieben werden. Zur Veranschaulichung können wir Parallelen zu der menschlichen Strategie ziehen, das Einmaleins auswendig zu lernen (unbeabsichtigter Lösungsweg), anstatt das zugrundeliegende Konzept der Multiplikation zu erlernen (beabsichtigter Lösungsweg). Der unbeabsichtigte Lösungsweg, d.h. der Shortcut, funktioniert perfekt für die Zahlen in der gelernten Tabelle. Sobald wir jedoch mit anderen Zahlen konfrontiert werden, funktionieren diese Abkürzungen nicht mehr und nur der beabsichtigte Lösungsweg kann konstant zur richtigen Lösung führen. Analog zu diesem Beispiel sind Shortcuts in DL-Modellen dadurch gekennzeichnet, dass sie in Standard-Benchmarks gut funktionieren, aber sich nicht für eine Anwendung unter anspruchsvolleren Konditionen verallgemeinern lassen. In diesem Zusammenhang kann Shortcut-Learning demnach auch als eine der *Generalisierung* entgegengesetzte Kraft interpretiert werden.

In VQA wurden Shortcut-Learning und Generalisierung vor allem im Zusammenhang mit Datensatz-Bias und dem notorischen Mangel an VG in modernen VQA-Modellen untersucht. Einer der bekanntesten und einfachsten Wege, diesen Mangel an VG aufzudecken, sind sogenannte Out-of-Distribution Tests, die so konzipiert sind, dass sie Fragen mit anderen Antwortverteilungen enthalten als in den Trainingsdaten gegeben sind. Damit wird versucht, die Notwendigkeit von VG als Teil des beabsichtigten Lösungswegs hervorzuheben, von welchem erwartet wird, dass er sich gut auf diese Tests verallgemeinern lässt.

Trotz der offensichtlich wichtigen Bedeutung von VG für VQA auf konzeptioneller Ebene, sowie der klaren Auswirkungen seines Fehlens auf Generalisierungsszenarien, wurde die Rolle von VG in VQA in der Praxis noch nicht klar genug herausgearbeitet. In dieser Dissertation untersuchen wir die Rolle von VG im Kontext von VQA Generalisierung und Shortcut Learning. Unsere Beiträge zum Forschungsgebiet umfassen die folgenden Teilaspekte:

1. Wir stellen ein neuartiges System namens "VQA by Lattice-based Retrieval" (VLR) vor, dessen Ziel es ist, den beabsichtigten Lösungsweg des VQA Tasks zu approximieren, indem VG für die Erzeugung der Antwort explizit eingefordert wird.

2. Wir stellen eine neuartige Metrik zur Messung von Visual Grounding in VQA Systemen vor.

3. Wir geben einen umfassenden Überblick über die Qualität von VG in einer Vielzahl von VQA-Architekturen.

4. Wir decken problematische Evaluierungsmethodiken in der VG Forschung auf, welche das Potenzial haben, die allgemeine Wahrnehmung der Relevanz von VG für VQA nachhaltig zu beeinflussen.

5. Wir entwickeln ein neues Konzept namens "Visually Grounded Reasoning" (VGR), welches VG und VQA Reasoning formal als Schlüsselkomponenten etabliert, die die VQA-Leistung im Kontext der VQA-Generalisierung und des Shortcut Learnings beeinflussen.

Wir beginnen die Reise dieser Dissertation mit einer konzeptionellen Vorstellung davon, was die Rolle VGs und seine Einflüsse in VQA sein *sollten*, aber erkennen schon bald, dass es abgesehen von den Beeinträchtigung durch Shortcut Learning auch noch andere fundamentale Einschränkungen in der gegenwärtigen VG-Forschung gibt, wenn es darum geht VG in VQA zu messen und VGs Einflüsse zu bewerten, was einer gründlichen Analyse im Weg steht. Wir adressieren diese Einschränkungen mit Hilfe unserer oben beschriebenen

Beiträge und erlangen dabei ein viel klareres Verständnis dafür warum der Einfluss von VG auf die VQA-Leistung sich so schwer erfassen lässt, und wie VGs Rolle durch entsprechend ausgearbeitete Evaluierungsszenarien in den Vordergrund gerückt werden kann. Unsere gesammelten Erkenntnisse und Untersuchungen führen uns dann im letzten Kapitel schließlich zur Einführung eines theoretischen Modells, das die Rolle von VG im Rahmen von VQA-Generalisierung und Shortcut Learning klar beschreibt, was das Ende unserer Reise markiert.

Abstract

Visual Question Answering (VQA) is the task of answering natural language questions about image contents. VQA models process two input modalities to produce an answer: vision (the image) and language (the question). A VQA model's answer is called *Visually Grounded* (VG), if it is based on question-relevant parts of the image. This process is sometimes more intuitively described as being *right for the right reasons*. On a conceptual level, the necessity of Visual Grounding is clear and its role seems obvious, but in practice, VQA models are notorious for their lack of Visual Grounding, while still achieving increasingly better performances. So the question is, how would VQA even be able to function properly without Visual Grounding and what role does it really play in VQA?

Like in many, if not all, of today's machine learning fields, modern models in VQA are based almost exclusively on Deep Learning (DL). While delivering unprecedented achievements throughout all areas of machine learning, including VQA, DL models face a significant limitation called *shortcut learning*. A shortcut can be characterized as an unintended solution to a given problem. As an analogy, let us consider the human strategy of learning the results of the multiplication table by heart (unintended solution) instead of learning the underlying mathematical concept of multiplication (intended solution). The unintended solution, i.e., the shortcut, works perfectly fine for numbers in the learned table. However, as soon as we are presented with different numbers, these shortcuts fail and the intended solution thrives in comparison. Similarly, shortcuts in DL models are characterized by working well in standard benchmarks, but fail to generalize to more challenging conditions. In this vein, shortcut learning can also be seen as an opposing force to *generalization*.

In VQA, shortcut learning and generalization have been investigated in the context of dataset biases and the notorious lack of Visual Grounding in modern VQA models. One of the most prominent and straightforward ways to expose this lack of Visual Grounding have been so-called Out-of-Distribution tests, which are designed to contain questions with different answer distributions than encountered in the training data, thereby attempting to emphasize

the need for Visual Grounding to be part of the intended solution that will generalize well to these tests.

Despite the obvious importance of Visual Grounding for VQA on a conceptual level and the obvious implications of its lack for generalization scenarios, the role of Visual Grounding in VQA in practice is still not clearly understood. In this thesis, we thoroughly investigate the role of Visual Grounding in VQA generalization and shortcut learning. Our contributions include the following achievements:

1. We introduce a novel system called "VQA by Lattice-based Retrieval", or VLR, whose goal is to align with the intended solution for the VQA task by explicitly necessitating Visual Grounding for answer inference.
2. We propose a novel metric for measuring Visual Grounding in VQA systems.
3. We report a large-scale overview of Visual Grounding quality across a wide variety of VQA architectures.
4. We uncover problematic evaluation methodologies in Visual Grounding research that have the potential to interfere with the general perception of how important Visual Grounding is for VQA.
5. We develop a novel concept called "Visually Grounded Reasoning", or VGR, that formally establishes Visual Grounding and VQA Reasoning as two key components that influence VQA performance in the context of VQA generalization and shortcut learning.

We set out on this thesis' journey with a conceptual idea of what Visual Grounding's role and impact in VQA *should* be, but learn soon that — apart from the interference of shortcut learning with these expectations — there are also other fundamental limitations of contemporary Visual Grounding research, when it comes to measuring Visual Grounding in VQA and properly evaluating its impact, that impede an in-depth analysis. We address these limitations with our contributions and in the process gain a much clearer understanding about why the impact of Visual Grounding on VQA performance has been so difficult to grasp and how its role can be better highlighted with appropriately designed evaluation scenarios. Finally, in the last chapter, this thesis culminates in the definition of a theoretical model (VGR) that clearly describes the role of Visual Grounding in the context of VQA generalization and shortcut learning, thereby marking the end of our journey.

Contents

List of Figures

List of Tables

Glossary of Terms, Abbreviations and Acronyms

VG Visual Grounding

VQA Visual Question Answering

OOD Out-of-Distribution

ID In-Distribution

DNN Deep Neural Network

DL Deep Learning

DET Detected (Features)

ORA Oracle (Features)

INF Infusion (Features)

FPVG Faithful and Plausible Visual Grounding

VLR VQA by Lattice-based Retrieval

VGR Visually Grounded Reasoning

TVG True Visual Grounding (Test Subset)

SC Shortcut (Learning)

GQA-AUG Augmented Test Set based on GQA

GQA Dataset for compositional VQA on scene graphs

IoU Intersection over Union (Metric)

LOO Leave-One-Out

SG Scene Graph

SGG Scene Graph Generator

OD Object Detector

V+L Vision and Language

CP Changing Priors

Chapter 1

Introduction

1.1 Motivation

1.1.1 The Field of Visual Question Answering

Visual Question Answering (VQA) (Antol et al., 2015) is the task of answering natural language questions about image contents (see Figure 1.1). With the integration of multi-modal input consisting of language and vision, VQA stands at the intersection of multiple major fields of computer science. As such, VQA has garnered considerable research interest in recent years, most notably from communities of Computer Vision (CV), Natural Language Processing (NLP) and Artificial Intelligence (AI).

VQA-related approaches are applicable to various potential use-cases in human-computer interactions (HCI) and robotics. Systems leveraging VQA-

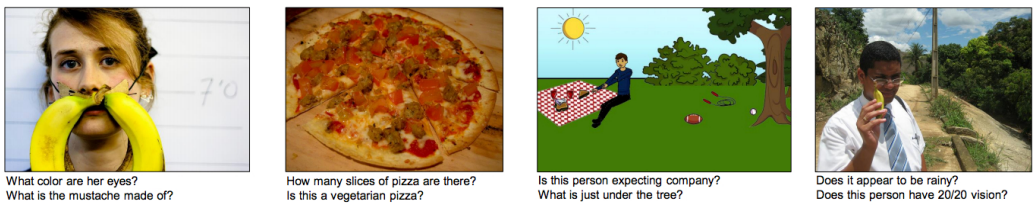

Figure 1.1 – Examples of the Visual Question Answering task. Image taken from Antol et al. (2015)

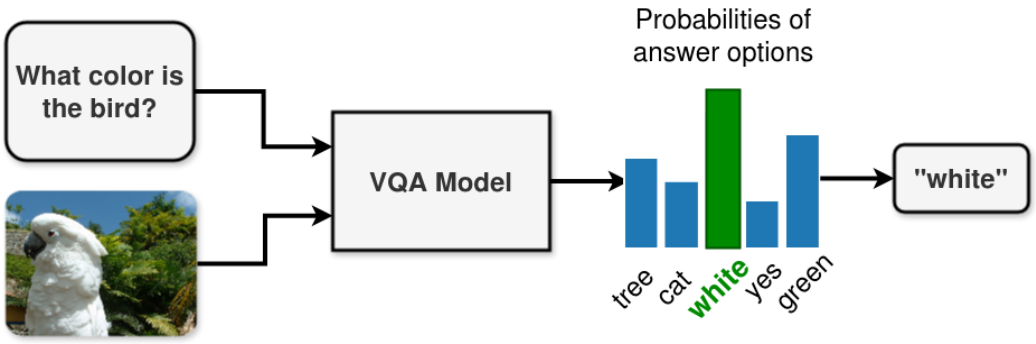

Figure 1.2 – Structure of a typical VQA system. Two input modalities, vision and language, are merged during processing by the model, which finally produces a probability distribution over answer alternatives. The most likely option is then picked as the model's answer to the given question about the image.

driven advances can improve independent living for the visually impaired (Chen et al., 2022) and establish more intuitive interfaces for media retrieval (Liu et al., 2021). Robotic designs can benefit from VQA-inspired methods by fusing a detailed visual scene representation (Chang et al., 2021) with language processing as the most natural way for humans to provide instructions or describe their surroundings (Kenfack et al., 2020).

The challenges that VQA addresses have been framed as Visual Turing Tests for Artificial Intelligence (Geman et al., 2015; Malinowski et al., 2015), thereby associating progress made in this field as highly relevant milestones on the road towards believable, human-like AI.

1.1.2 VQA and Visual Grounding

The VQA Model Blueprint. VQA models are characterized by two input modality streams: vision (image representation) and language (question representation). The VQA task is typically cast as a classification problem. Accordingly, a model's desired output – a textual answer to the question – is commonly realized by a probability distribution over predetermined answer alternatives defined by a given dataset (e.g., Goyal et al. (2017); Hudson and Manning (2019)). Figure 1.2 illustrates the typical design of today's VQA systems.

VQA as an Information Retrieval Task. While classification is by far the more popular design choice in VQA, the task of VQA can also

be framed as an information retrieval problem, with the vision modality carrying the knowledge base and the language modality representing a query to it. When interpreting VQA in this manner, we notice a fundamental characteristic that is intuitively understood by humans: successful production of the correct answer necessarily involves extracting relevant, query-specified piece(s) of information from the knowledge base. In VQA context, this intrinsic characteristic of involving relevant pieces of visual information in the extraction and production of the answer is called **Visual Grounding** (VG). The successful manifestation of VG during answer production is sometimes described as *being right for the right reasons* (Ross et al., 2017; Ying et al., 2022), although, it should be noted that correctness of the returned answer is not a necessary consequence of proper VG. Proper VG is simply characterized by a model's meaningful reliance on question-relevant visual information during answer inference, regardless of the produced answer.

Visual Grounding and Shortcut Learning in Deep Neural Networks (DNNs). Contrary to the framing of VQA as an information retrieval task, where VG assumes an obvious and central role without which a correct answer is improbable, we find that the role of VG in modern VQA systems is not as easily understood, and its impact – or even necessity – substantially less obvious. This is owed to the fact that most successful VQA models today are not conceptualized as information retrieval systems utilizing human-defined decision rules, but are conceived as powerful classifiers based on complex DNNs that are increasingly often trained end-to-end. Common practices of discriminative training for complex DNN-based classifiers have been shown to regularly result in so-called shortcut learning in Computer Vision (Beery et al., 2018) and Natural Language Processing (Niven and Kao, 2019). Shortcuts can be described as unintended solutions for a given task that do not transfer well to certain generalization conditions (Geirhos et al., 2020). It can be argued that shortcut learning is an influential contributor to DNNs tremendous success in task-specific benchmarks in various fields. The reason for this lies in the way benchmarks are typically constructed: Common benchmarks are created under data assumptions called "independently and identically distributed", or i.i.d., which refers to train and test samples being drawn from the same distribution (i.e., a common source dataset). Tests created under these conditions are also referred to as In-Distribution (ID) tests. As ID tests stem from the same distribution as train sets, there is a good chance that any shortcuts that might have been learned in training can also be exploited in ID tests. Since benchmarks represent the most straightforward way of comparing model performances and tracking research progress, the extent

of a model's shortcut exploitations can often go unnoticed without explicit efforts to uncover them. As a result, such inconspicuous contributions of shortcut learning to benchmark success can give rise to a misplaced sense of confidence in high-scoring models which regularly perform poorly when evaluated in real-life scenarios falling outside the i.i.d. dataset. Scenarios to uncover shortcut exploitation can be simulated by creating diverse test variations known as Out-of-Distribution (OOD) tests. Performance on such OOD tests are generally characterized by their significant challenge for any model trained under i.i.d. conditions without taking special precautions. Such precautions may include explicit enforcement of human-intended decision rules to solve a given task. In VQA, we identify VG as one of such intended decision rules a model should absorb based on human intuition (but regularly does not).

1.1.3 The Impact of VG in VQA

Influence of VG. In VQA, high-scoring results in the primary performance metric of *answer accuracy*[1] in ID testing can give the impression that a model must have learned all relevant human-intended decision rules, including VG. Perhaps it is this misconception that contributes to the fact that VG itself is rarely attempted to be explicitly quantified when new benchmark-topping VQA systems are introduced. The omission of such investigations into the involvement of a known axiomatic decision rule like VG and the resulting lack of insight into a VQA model's decision making process, however, unnecessarily hurts a model's predictability. Consequently, under-evaluated models are more likely to surprise us with unexpected behavior when challenged in OOD scenarios designed to test their generalization capabilities and uncover shortcut exploitation. While accuracy in specially designed OOD tests can be a good indicator for shortcut exploitation, in the case of VG, designing tests that successfully and cleanly isolate VG's influence on accuracy is a challenging task in itself. This is because in the absence of capable VQA reasoning, VG on its own is not expected to affect answer accuracy in any significant capacity. When illustrating this point within the framing of VQA as a retrieval task we find that we cannot expect a correct answer, if the system does not understand what type of information the query is requesting from the knowledge base. Hence, measuring VG directly instead of via proxies like accuracy tests (even OOD) is the preferable and more definitive method for confirming to what degree it has manifested in the model. Nevertheless, model

[1] Answer accuracy is calculated as the percentage of correctly returned answers, unless specified otherwise.

Figure 1.3 – Over-reliance on language causing image content to be ignored when producing the answer. Figure taken from Agrawal et al. (2018).

behavior (seemingly) revealing a lack of VG has regularly been illustrated via accuracy-based proxy tests in pertinent VQA literature, of which we describe two notable examples in the following.

Observations of VQA model behavior attributed to a lack of VG. A VQA system is considered "well-grounded" if it infers an answer to a given question by relying on image regions that are relevant to the question and plausible to humans when considering the inference process necessary to resolve said question. Hence, visually grounded inference in VQA can be broken down into two aspects: (1) The model's answering process is impacted by the contents of the input image in a principle capacity. (2) The model's answering process relies on *relevant* image content.

Various works have reported observations of problematic behavior in VQA systems that are linked to missing support for the aforementioned properties:

1. Many VQA models have been shown to suffer large performance drops in OOD testing where relevant image contents tied to the answer deviate from contents predominantly seen in training for the same question types. Figure 1.3 shows Q/A examples of a VQA model ("SAN" (Yang et al., 2016)) seemingly ignoring visual evidence in the image and instead selecting an answer that dominates the sample distribution for the given question in training. This behavior can be attributed to an over-reliance on the language modality (Goyal et al., 2017; Agrawal et al., 2018, 2016), and therefore points to a lack of property (1).

2. Gupta et al. have observed different answers to the same question after manipulation of image regions that are plausibly irrelevant to the given

question. Figure 1.4 illustrates this phenomenon. The depicted behavior suggests that while the model's decision is meaningfully influenced by visual information, it appears to not be based on information that is relevant to the question, which points to a lack of property (2).

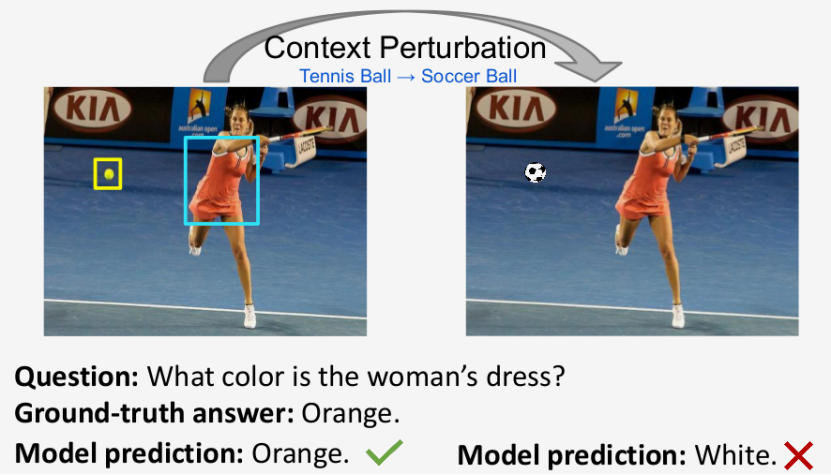

Figure 1.4 – Alterations of irrelevant image contents (ball, right) impact the model's answer. Figure taken from Gupta et al. (2022).

These findings illustrate how shortcut learning can manifest itself in VQA models and how insufficient VG might hurt a model's capacity to provide consistent and reliable performances outside of commonly benchmarked ID accuracy.

Attempts at VG-focused Remedies. Solutions for improving OOD generalization in VQA have been a topic of great interest in the community. Following observations made in Example 1 above, one line of research attempts to provide remedies to shortcut exploitation based on the assumption that wrong answers given in OOD tests must mean that models do not accurately ground their reasoning in the image, i.e., do not rely on (relevant) image content when inferring an answer. Hence, improving VG is being motivated as a natural solution for improving OOD generalization in VQA. Surprisingly, however, prominent methods like Selvaraju et al. (2019) and Wu and Mooney (2019), which were designed to strengthen VG in model training, have been shown to achieve accuracy improvements on OOD tests when trained with both meaningful, as well as, nonsensical visual region relevance cues (Shrestha et al., 2020), raising doubts about the true significance of VG for answer performance, even in OOD scenarios. Moreover, Ying et al. (2022) have found no advantage of using existing VG metrics to predict OOD performance instead

of doing so with ID accuracy, adding to the obscurity of VG's involvement in VQA generalization even further. Consequently, the importance of VG's role in generalization scenarios has yet to be conclusively determined and demonstrated, a task we accomplish in Part III of this thesis.

The role of VG in VQA. Motivated by these conflicting accounts of empirical reports implying VG's lack of utility, and the described theoretical necessity of VG's involvement in reliable VQA, we formulate the goal of this work as follows. The purpose of this thesis is to shed light onto VG's mystery role in VQA and unveil its long suspected value in generalization settings in practice. To support this endeavor, we develop a variety of methods and empirical procedures. In the course of this thesis, we uncover that VG does in fact play a crucial role in generalization scenarios.

1.2 Contributions

In this thesis, we seek to develop new insights into the role of VG in VQA in general, and its importance in OOD settings in particular. The contributions of this work are summarized as follows:

1. *Visually Grounded VQA:* A modular and transparent VQA system developed under an information retrieval paradigm. Our system largely separates reasoning, Visual Grounding and answer accuracy, thereby enabling us to isolate and analyze each part involved in a rule-based VQA system. As this system represents an approximation of shortcut-free VQA in the sense that it is specifically designed to make use of human-intended decision rules to solve the task, it constitutes a valuable tool and baseline system for our investigations.
   We further develop a number of new VQA generalization tasks that demonstrate our system's advantages in the context of VQA generalization. We share these tasks with the research community[2] to encourage development in this direction.

2. *Measuring Visual Grounding:* The lack of a general, unified, straightforward metric to quantify VG capabilities across a wide variety of VQA model architectures further complicates the proper assessment of VG's impact in practice. We propose a VG metric that is both *faithful* and *plausible* in its explanations and can be applied to most VQA model architectures without significant adaptation efforts.

[2] https://github.com/dreichCSL/GQA_generalization_splits

We share our implementation of FPVG with the research community[3] for easier adoption of our new metric.

3. *Information Infusion with Symbolic Features:* Standard sub-symbolic visual features used for image representation in VQA are black boxes w.r.t. their informational payload, making it difficult to understand and control the information carried by the visual modality. We propose a method we call "Information Infusion" and use it in combination with symbolic feature representations to fully control the visual information flow, which opens up a number of options for in-depth VG analysis.

4. *Efficacy of VG-methods:* VG-methods employed with the goal of improving OOD performance have come under scrutiny in related work (Shrestha et al., 2020; Ying et al., 2022). We show that problems in the commonly used evaluation scheme hinder a clear assessment of their impact on the model, which is adding to the confusion about VG's utility.
   We share our implementation with the research community[4] to facilitate reproduction of the involved experiments and encourage adoption of the introduced "True VG" methodology in VG analysis.

5. *Visually Grounded Reasoning:* We propose a theoretical model that defines VQA inference as a co-dependency involving VG and VQA Reasoning. On the back of this model, we investigate the common practice of using OOD testing as a way of assessing inherent VQA model characteristics including VG and VQA Reasoning strength. We show why typically used OOD tests are not ideal to fully understand and measure the involvement of either of the two and propose a test design that does accentuate their involvement. Finally, our investigations lead us to new insights of the relationship between VG and VQA Reasoning, as well as their combined effect on model accuracy.

1.3 Structure of this Thesis

The remainder of this thesis is structured as follows:

Part I — Background

Chapter 2: Background on VQA, VG and shortcut learning.

[3] https://github.com/dreichCSL/FPVG
[4] https://github.com/dreichCSL/TrueVG

Part II — Methods: Novel approaches that highlight and measure Visual Grounding in VQA

Chapter 3: VLR: Description of our information retrieval-based VQA system.

Chapter 4: FPVG: Description of our VG metric.

Chapter 5: Symbolic features & Information Infusion: Our technique for manipulating image representations in VQA input feature space.

Part III — Investigations & Insights: Findings of our investigations into the role of Visual Grounding in VQA generalization and shortcut learning

Chapter 6: Overview of VG strength and OOD performance in a large variety of common VQA architectures. Initial attempts at understanding the connection between VG and OOD performance.

Chapter 7: Investigations into the full potential of VG-boosting methods in VQA.

Chapter 8: Definition of the concept of Visually Grounded Reasoning in VQA and the introduction of a test scenario that necessitates involvement of VG.

Part IV — Conclusion

Chapter 9: Summary of this thesis and closing remarks.

Part I

Background

Introduction

In Part I of this thesis, we provide background information on the field of Visual Question Answering, Visual Grounding and shortcut learning.

In Chapter 2.1 and Chapter 2.2, we formally introduce the task of Visual Question Answering and give an overview of influential milestone modeling approaches proposed in recent years.

Chapter 2.3 describes prominent datasets used to track progress in the field of VQA.

Chapter 2.4 gives an introduction of the concept of shortcut learning in Deep Neural Networks and discusses its relationship to generalization.

In Chapter 2.5, we introduce the concept of Visual Grounding and establish its connection to shortcut learning, generalization and OOD testing. We further describe a number of existing datasets and methods involved in research about shortcut exploitation in VQA as well as VG research.

Chapter 2

Background: Visual Question Answering, Visual Grounding and Shortcut Learning

2.1 Introduction to Visual Question Answering Modeling

In contemporary research, the task of Visual Question Answering (VQA) is typically cast as a classification problem. It is under this modeling paradigm that VQA research has achieved the most substantial progress in recent years.

Conceptually, VQA models are characterized by two input modality streams: Vision (image representation) and language (question representation). The answer is delivered as a single class output of the model. As such, the problem of VQA can be formalized as the following function transformation:

$$F : (Q, I) \mapsto A \tag{2.1}$$

The function transformation F is realized by a VQA model, mapping its input questions Q and images I to answers A. Concretely, for a given VQA model, we have

$$f_{VQA}(q, i) = P(a|q, i), \tag{2.2}$$

with $f_{VQA} \in F$ being an instantiated VQA model with inputs $q \in Q$ and $i \in I$. By far the most common way VQA models determine an output answer

is by selecting the most likely candidate a in a probability distribution P over a set of pre-defined answer alternatives A:

$$a = \underset{a \in A}{argmax}\ P(a|q,i) \tag{2.3}$$

2.1.1 Vision and Language Representations

Most contributions to contemporary VQA systems share commonalities in aspects of input pre-processing for vision and language.

Visual representation. Most modern VQA models are not designed to process raw (digitized) images as visual input representation. Instead, powerful image processing models are used in a decoupled pre-processing step to provide a sophisticated, featurized image representation for the VQA model to leverage. These image processing models are typically trained independently on a large number of images (one million and more) to perform more general tasks like image or object recognition based on popular datasets like ImageNet (Deng et al., 2009) and MS COCO (Lin et al., 2014). The exact makeup of an image representation that acts as input to a VQA model varies in structure and ranges from grid-based visual features (Yang et al., 2016; Jiang et al., 2020) over object-based bag of vectors (Anderson et al., 2018) to highly structured scene-graph representations (Hudson and Manning, 2019; Hu et al., 2019). Attempts at replacing external image processing modules altogether follow a more holistic modeling approach that integrates training for task-appropriate image processing of the raw image into the VQA model. For instance, Kim et al. (2021) proposes to process the input image within the VQA model as small image patches (Dosovitskiy et al., 2021) with promising results. In practice, however, object-based representations extracted from an object detector still are the de-facto standard for visual representations in VQA since their popularization by Anderson et al. (2018).

Question representation. The input question is originally given as text in natural language, which requires a transformation into numerical vectors prior to processing. In most cases, VQA models leverage an independently optimized transformation process to attain meaningful mappings of text into numerical space. The question is then represented as a sequence of so-called word embeddings. These word embeddings are a vectorized, distributed representation of a word that is characterized by the word's semantic context. Embeddings acting as input to a VQA model are usually obtained from independently trained models such as word2vec (Mikolov et al., 2013), GloVE (Pennington et al., 2014) or BERT (Devlin et al., 2019). These models have

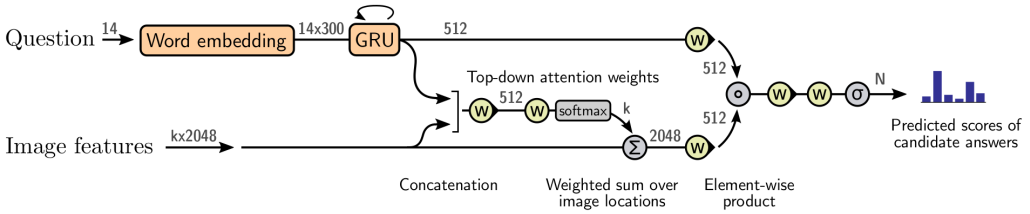

Figure 2.1 – Single-hop inference in "Bottom-Up Top-Down", where a one-time attention operation (single-hop) is employed to determine the weight (i.e., question-relevance) of each visual object. The two VQA input modalities, vision and language, are then merged accordingly. The model finally produces a probability distribution over answer alternatives from which the most likely option is picked as answer to the question. Image taken from Anderson et al. (2018).

usually been trained in a self-supervised fashion on vast amounts of general text data consisting of billions of words.

2.2 Mechanisms of Inference

VQA systems have showcased impressive gains in answer accuracy performance in recent years and the pursuit of which has spawned a wide variety of modeling approaches. In the follwing, we give an overview of some of the most influential mechanisms of inference in recent VQA model design.

2.2.1 Question-guided Attention

Intuitively, the inference process of a VQA system can be understood as a navigation over image content. In its simplest form, this process can be realized in a VQA system by a question-guided attention mechanism (Bahdanau et al., 2015) over the visual input representation. The idea behind this mechanism is to empower the model with the ability to focus on question-relevant parts of the visual input representation while discounting other parts.

Single-hop inference. In the classic Bottom-Up Top-Down (UpDn) model (Anderson et al., 2018), depicted in Figure 2.1, question-guided attention ("Top-down attention weights" in the figure) is realized as a one-time operation to determine the weight — or question-relevance — of each element in the visual input (represented as a bag of visual objects) w.r.t. the given question (represented as a single question vector assembled by an RNN that processes

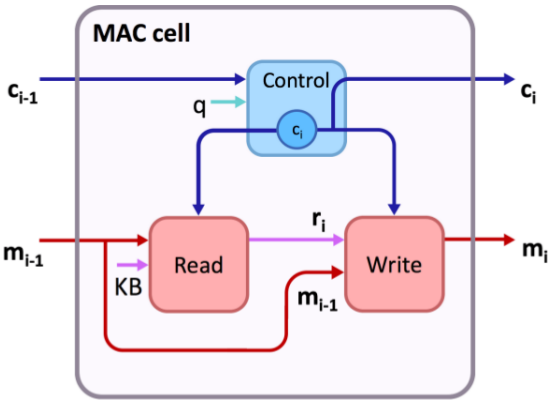

Figure 2.2 – The MAC cell (image taken from Hudson and Manning (2018)). The Control unit attends to different aspects of the question in each cell. Image information relevant to the currently processed aspects of the question is extracted in the Read unit and then integrated by the Write unit into the memory state, which informs the next inference step.

the question). This type of mechanism is called *single-hop inference*, as the entire question-guided navigation over image content is captured by a single attention operation.

Multi-hop inference. Questions such as "What color is the dog to the right of the man wearing a bowtie?" require humans to take more than just a single glance at the image before answering. Hence, such questions might be better handled with a modeling mechanism that allows for *multi-hop inference* to incorporate all information laid out as part of the question. The MAC network Hudson and Manning (2018) implements such a mechanism by stringing together RNN-based cells called "Memory, Attention, Composition", or MAC cells (pictured in Figure 2.2). The MAC network decomposes the question into a fixed number of inference steps that each involve an attention operation over image content w.r.t. different aspects of the question.

2.2.2 Bilinear Pooling

Extensive pairwise interactions between all feature elements of the two input modalities in VQA can be modeled with bilinear models to obtain a rich joint input representation. An efficient way to realize this mechanism in DNNs is by low-rank *bilinear pooling* (Kim et al., 2016), which significantly reduces the number of parameters otherwise needed to be learned in a straightforward

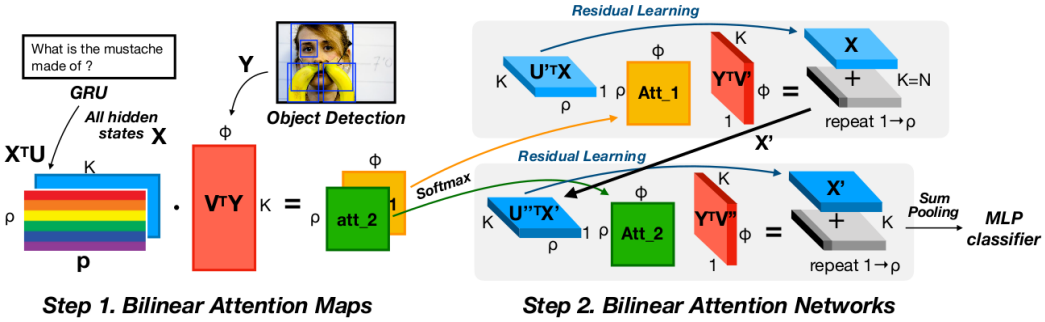

Figure 2.3 – Multi-head attention over rich, bilinearly mapped joint representations of vision and language is the key contribution of BAN (image from Kim et al. (2018)). Step 1 shows "two-glimpse" attention (i.e., two-headed) which produces two bilinear attention maps from an input of ϕ visual features and ρ question features. Step 2 pictures the creation of BAN's joint input representation that is the basis for VQA classification.

bilinear operation[1]. The Bilinear Attention Network (BAN, Kim et al. (2018)) pictured in Figure 2.3 combines this concept of low-rank bilinear pooling with multiple attention heads (i.e., attention mechanisms over the same input but involving different weight matrices per head).

2.2.3 Self-attention and Co-attention

While MAC applies an additional attention mechanism over the question to highlight different aspects in its interactions with the image in a small capacity, BAN manages to model the exhaustively rich interaction between all question and image feature dimensions via bilinear pooling. Both approaches, however, do not model interactions between elements within the same modality. Therefore, MCAN (Modular Co-Attention Networks) (Yu et al., 2019b) proposes to use Transformer-inspired (Vaswani et al., 2017) multi-headed *self-attention* layers to model intra-modality interactions (Figure 2.4, left), producing much richer individual modality representations. The resulting representations are then combined with a form of question-guided attention (Figure 2.4, right) over the image representation, where each (self-attended) question element is used to identify (self-attended) image elements relevant to it. This type of inter-modal interaction between all elements in both modalities is called *co-attention*.

[1]In an unoptimized bilinear Neural Network layer, the size of the involved weight matrix is dependent on the result of the Kroneker product between two input modality matrices and thus consists of an impractically large number of learnable parameters for VQA tasks.

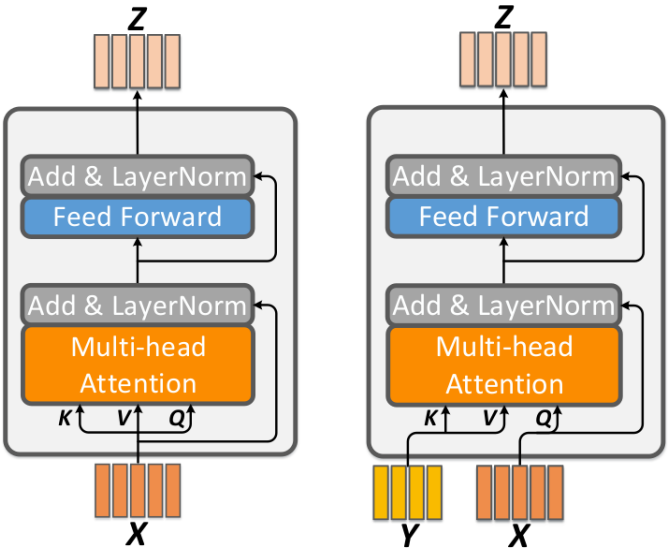

Figure 2.4 – Transformer-inspired multi-headed self-attention (left) and guided-attention units (right) are the building blocks of MCAN (image taken from Yu et al. (2019b)). Additional details in Chapter 2.2.3.

2.2.4 Large-Scale Pre-Training with Transformers

As seen with MCAN, leveraging Transformer-based concepts of multi-headed self-attention and the Transformer's capacity to stack layers easily have shown great potential in learning rich representations for VQA through profuse intra- and inter-modality interaction modeling. Expanding in this direction, the LXMERT model (Learning Cross-Modality Encoder Representations from Transformers), shown in Figure 2.5, adopts the strength of Transformers and BERT-like (Devlin et al., 2019) training regiments to learn rich multi-modal representations for VQA. LXMERT (and other similar Transformer-based Vision+Language (V+L) models like ViLBERT (Lu et al., 2019) and OSCAR (Li et al., 2020)) is inspired by the uni-modal language model BERT (Devlin et al., 2019), a Transformer-based modeling milestone in the NLP field. BERT is designed to learn rich language representations by pre-training with large amounts of text data. The resulting model is fine-tuned with smaller datasets for solving specific downstream (NLP-related) tasks. LXMERT follows this kind of pre-training/fine-tuning scheme for V+L tasks and additionally includes vision and cross-modality encoders alongside a language encoder. Similar to MCAN, cross-modality representations involve both self-attention layers for intra-modal interaction modeling as well as vision and language-guided attention for inter-modal interaction modeling. One of LXMERT's key differences to MCAN is the learning scheme which

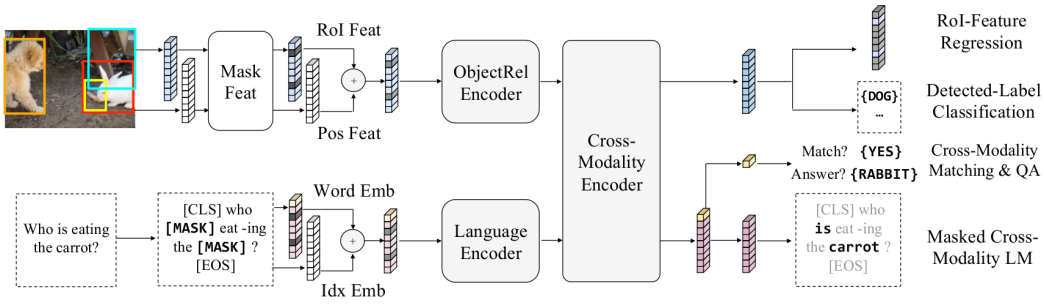

Figure 2.5 – LXMERT: Transformer-based model pre-training picturing several pre-training objectives (right) to learn rich cross-modality representations for V+L tasks and VQA in particular (image from Tan and Bansal (2019)).

involves pre-training the encoders with several datasets consisting of various V+L tasks (i.e., in addition to VQA-specific datasets which MCAN uses). Its pre-training procedure includes a number of uni-modal and multi-modal learning objectives, including masked language modeling and a standard VQA task loss (w.r.t. answer correctness). After the lengthy pre-training process on large amounts of data, LXMERT's cross-modality encoding is leveraged in training (also called "fine-tuning" in this context) a comparably simple classification head for VQA tasks using only task-specific data.

2.2.5 Relation Modeling with Scene Graphs

Powerful Transformer-based models like LXMERT provide the means for learning rich joint-modality representations, but do not offer intuitive interpretations regarding their inference process. Multi-hop inference, such as seen for MAC, offers a sensible and interpretable way to model the process of resolving a question that implicitly involves interpreting question-referred relations between objects in an image in a sequential fashion. The common bag-of-objects image representation in VQA, however, typically provides only rudimentary spatial information in the shape of location coordinates of the input objects and lacks explicit information about semantic relationships between objects in the scene. As a result, cases where questions explicitly refer to semantic relationships between objects, models with multi-hop inference may not reap the full benefits from their sequential processing. Scene graph representations of images (Zitnick and Parikh, 2013; Yang et al., 2018; Chang et al., 2021), where objects are represented as nodes and relationships as edges, can help remedy this shortcoming, but require the adoption of dedicated modeling techniques such as proposed in Graph Attention Networks (Veličković et al., 2018), to process them effectively. Various models have adopted inference over

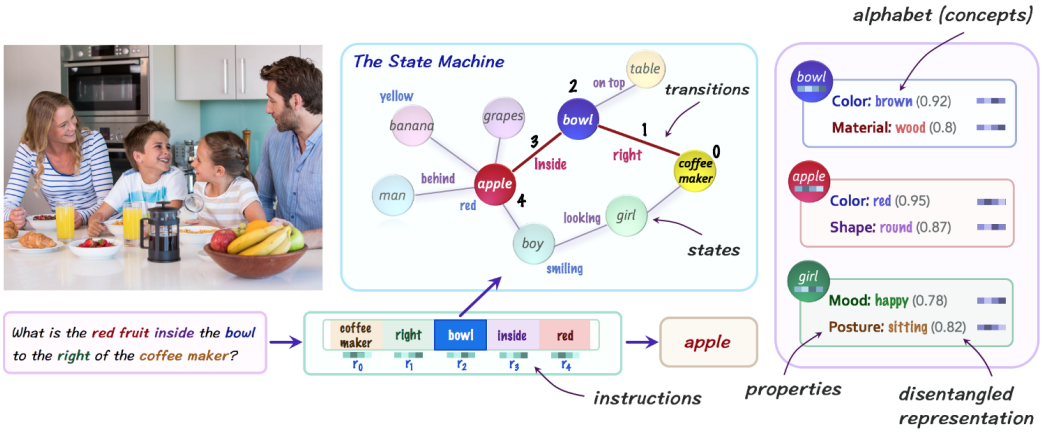

Figure 2.6 – NSM's traversal process over a symbolic scene graph guided by sequentially constructed inference instructions from the input question (image from Hudson and Manning (2019)).

scene graphs with success in particular for compositional VQA, i.e., the type of questions requiring correct interpretation of relations (Liang et al., 2021; Li et al., 2019b; Hu et al., 2019). A notable example for a model that implements sequential multi-hop inference by iterative traversal of scene graphs is the Neural State Machine, or NSM (Hudson and Manning, 2019). NSM's key contribution is its graph traversal-based inference over an intricately rich symbolic representation of the visual modality, which compiles object names, their attributes and relationships in a probabilistic scene graph. A demonstration of its functionality is shown in Figure 2.6.

2.2.6 Disentangled Inference

So far, the discussed inference mechanisms target modality fusion as a central aspect of modeling inference in VQA. Among them, NSM takes a special place as a model that focuses more on actual *navigation* of the given image information than on a cross-modal *transformation* into a joint representation which inherently encapsulates inference. In this sense, NSM can be interpreted as trying to close the gap between classification-based and retrieval-based VQA, where modalities remain disentangled throughout inference and answers are retrieved from the visual knowledge base rather than selected from a global answer set via answer classification. Another neuro-symbolic model that more fully embraces modality disentanglement and retrieval-like answer production is proposed with DFOL (Differentiable First-Order Logic) in Amizadeh et al. (2020c). An illustration of the system is shown in Figure 2.7. Informed by a symbolic image representation similar in content richness to

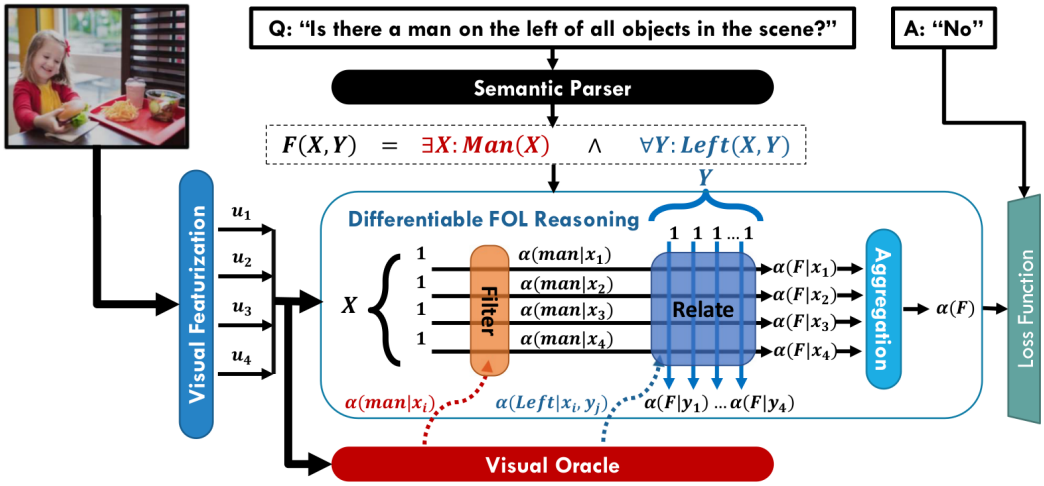

Figure 2.7 – Retrieval-like VQA by evaluation of first-order logic-based scene descriptions in DFOL (image taken from Amizadeh et al. (2020c)). See Chapter 2.2.6 for a detailed description.

NSM's scene graph ("Visual Oracle" in Figure 2.7), DFOL evaluates first-order logic-based (FOL) reformulations of the input question generated by an independently trained semantic parser. These reformulations embody a kind of visual description of plausible answers to the question. Questions with a binary answers, such as the one pictured in Figure 2.7, are evaluated via queries to the Visual Oracle according to each partial description in the FOL formulation. Figure 2.7 illustrates how DFOL aggregates probabilities of each object in the Visual Oracle of being a "man" (first FOL description in red) and being positioned to the left of all other objects in the image (second FOL description in blue), which then informs the answer. More challenging query-type questions, such as "What color is the chair?" (not pictured here), require evaluation of one FOL reformulation per color answer possibility. Essentially, this question would be represented by FOL formulations such as "there is a red, blue, brown, ... chair", which are each evaluated to determine the likeliest statement, thereby implying the answer. The evaluation of a FOL description, i.e. DFOL's central inference procedure, relies heavily on queries to specific information content in the image, which are processed in a sequential multi-hop manner. Importantly, vision and language remain disentangled throughout the entirety of DFOL's inference.

We note that in Chapter 3 of this thesis, we introduce our own concept for a retrieval-based VQA system.

Dataset	#imgs	#questions	#unique answers	image type	QA source
VQAv1	205k+50k	614k+150k	>23k	real-world + abstract scenes	crowd-sourced
VQAv2	205k	1.1m	>23k	real-world	crowd-sourced
GQA	113k	22m	1.88k	real-world	template-based
GQA balanced	86k	1.08m	1.84k	real-world	template-based
CLEVR	100k	1m	28	rendered	template-based

Table 2.1 – Statistics of VQA datasets. All numbers were taken from respective publications. Exception: Numbers for GQA balanced represent only data points with fully released annotations (i.e., excludes benchmark tests).

2.2.7 Interpretable Inference in Monolithic Models

Finally, the Meta Module Network (MMN) proposed in Chen et al. (2021) marries two concepts seen in earlier architectures: Dedicated question parsing and DNN-based multi-hop inference. An externally trained question parser transforms the question into a straightforward sequence of instructions (also called a "program") to inform navigation over the visual input. These programs essentially replace the original question as input to MMN's main DNN-based multi-hop inference process. MMN's inference transparency benefits greatly from this setup, as the question parser produces explicit, interpretable inference steps, which is unlike other similar DNN-based architectures employing multi-hop inference such as MAC or NSM. Furthermore, in contrast to DFOL, which also parses the question into interpretable functions, MMN leverages the generated programs as question input to a powerful DNN-based model with Transformer-inspired neural modules to achieve strong performances in compositional VQA tasks.
A notable aspect of MMN's training process is its auxiliary objective function that optimizes the model's ability to identify visual objects relevant to each processed inference step as a secondary task. The inclusion of such an objective function can be interpreted as an attempt to improve the model's plausibility in its Visual Grounding, which is a concept we will discuss in depth in Chapter 2.5.

2.3 Datasets for Visual Question Answering

In this section, we introduce three datasets that are routinely used to track progress in VQA research. A summary of their traits and statistics is given in Table 2.1.

Figure 2.8 – Questions about real-world images and abstract scenes (third from left) in the VQA dataset. Image from Antol et al. (2015).

2.3.1 The VQA Dataset

The VQA dataset proposed in Antol et al. (2015) consists of questions about real-world images from the MS COCO image database (Lin et al., 2014), as well as additional images depicting abstract scenes. MS COCO itself is a popular benchmark for object detection and object recognition tasks. Figure 2.8 shows a few representative examples of images and questions in the VQA dataset.

The VQA dataset consists of a mixture of real-world (205k images) and abstract images (50k images). In total, 764k questions and about 10 million answers[2] were collected from human subjects: Three questions per image and 10 answers for each question (one answer per subject). Each question can have up to 10 unique answers, depending on the agreement between subjects. The total collected number of *unique* answers for real-world images exceeds 23k, i.e., many of the collected answers occur multiple times. In evaluations, this set is usually truncated (e.g., only the most frequent 1k answers are considered). Per the authors' recommendation for evaluating a model's answer accuracy, a predicted answer should evaluate as correct if at least three subjects agreed with it. The formal specification of this recommendation is given in relevant parts of this thesis (Chapter 7.8.1).

The VQA dataset aims to represent a mixture of both Information Retrieval-based VQA and commonsense-based VQA. The former is represented by questions that can be accurately answered by *extracting and validating* relevant image content ("Is the *man* wearing *shorts* and eating *ice-cream*?"), while the latter requires additional commonsense knowledge to help *interpret* relevant image content (e.g., "Is the man *feeling warm*?"). Both types of questions represent significant use-cases in practical VQA.

[2]This number includes additionally collected answers from subjects when not looking at images (used for analytical purposes).

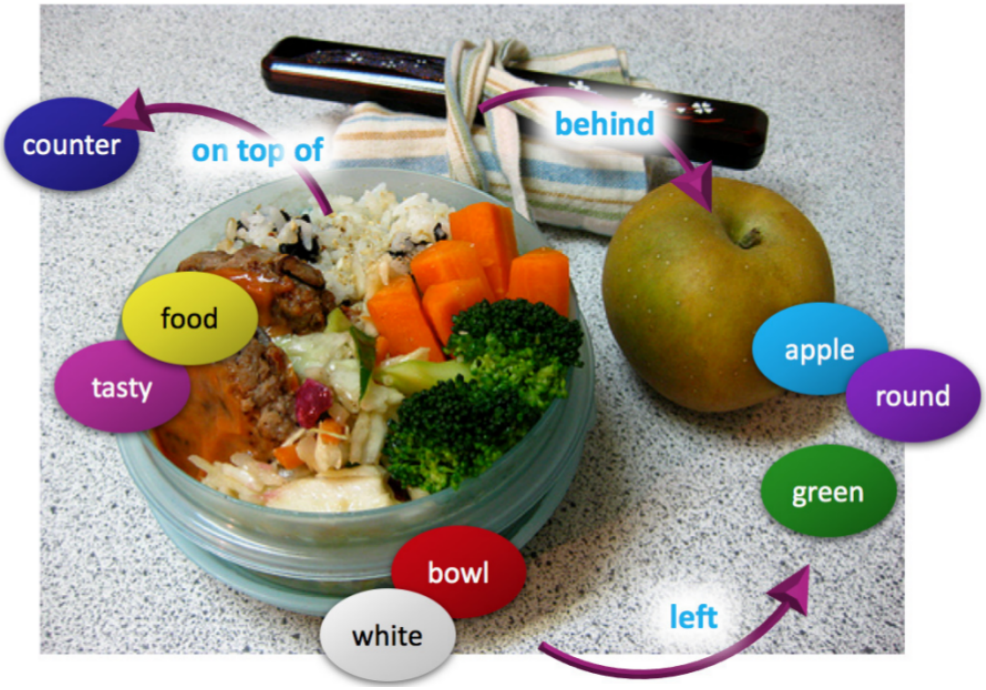

Figure 2.9 – Illustration of a scene graph from the GQA dataset, which serves as foundation for generating questions and answers for the dataset. GQA focuses on retrieval-based, compositional VQA (image from Hudson and Manning (2019)).
Example questions:
Is the bowl to the right of the green apple?
What type of fruit in the image is round?

In response to reports of shortcut exploitation (definition provided in Chapter 2.4) facilitated by the dataset's sample distributions in the first version of the VQA dataset (=VQAv1) (Zhang et al., 2016; Kafle and Kanan, 2017; Agrawal et al., 2016), a second iteration to this dataset, VQAv2, was introduced in Goyal et al. (2017) to improve on this issue. VQAv2 adds a considerable number of new QA-Image triplets (a new total of 1.1m questions) to re-balance the dataset's biased prior distribution and counteract shortcut exploitation by VQA models. We discuss this phenomenon in more detail in Chapter 2.5.1.

2.3.2 The GQA Dataset

The GQA dataset (Hudson and Manning, 2019) was proposed for compositional VQA and represents a retrieval-based VQA benchmark with focus on visual scene understanding. GQA consists of real-world images taken from

Visual Genome (Krishna et al., 2016), a large-scale crowd-sourced database for various image processing tasks. Similar to the VQA dataset, Visual Genome also uses images from MS COCO, and therefore GQA and the VQA dataset have some overlap in this regard.

GQA consists of 113k real-world images and over 22 million questions. Each image is accompanied by an annotated scene graph (Johnson et al., 2015) that provides information about objects, attributes and relations in the image. Unlike in the VQA dataset, questions in GQA are not crowd-sourced but generated using an extensive linguistic grammar, i.e., templates, in conjunction with image information provided by the given scene graphs. The Q/A templates themselves were either manually constructed or derived from questions in VQAv1.
Each question is further accompanied by a functional program consisting of a number of reasoning steps to be executed like navigational instructions on the respective scene graph to answer the question. A large majority of questions also list pointers to visual evidence in the image that is relevant for producing the correct answer (we call these "relevance annotations"). Each question is coupled with a single answer. The total number of unique answers in the dataset is about 1.9k.
Figure 2.9 shows an example image and a couple of typical questions in this dataset.

Special precautions were taken to reduce unwanted statistical biases that had been previously identified in the VQAv1 dataset (cf. Agrawal et al. (2016); Goyal et al. (2017)). This resulted in the creation of the "balanced" split which achieves a more uniform answer distribution within each question group. The "balanced" split consists of a down-sampled 1.7 million questions.

2.3.3 The CLEVR Dataset

Similar to GQA, the diagnostic CLEVR dataset (Johnson et al., 2016) was proposed for compositional VQA and retrieval-based reasoning. While the high visual complexity of real-world images in GQA and VQA pose a significant challenge in these datasets, CLEVR intentionally avoids this kind of visual complexity to focus on aspects of scene understanding and reasoning.

The CLEVR dataset consists of 100k synthetic images depicting a small number of simple 3D shapes, rendered with the graphics tool Blender (Blender Foundation, 2016). The arrangement of the shapes is based on a randomly sampled scene graph consisting of three to ten objects. The scene graphs are accompanied by annotations of object positions and four attribute types like shape

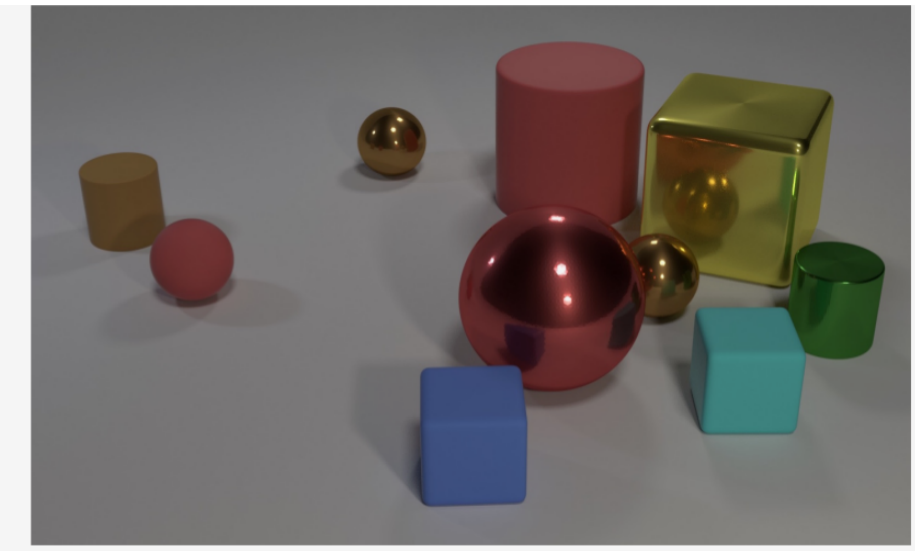

Figure 2.10 – Rendered image and synthesized questions in the CLEVR dataset (image from Johnson et al. (2016)).

and color. Similar to GQA's question generation, which drew inspiration from CLEVR, one million questions were synthetically generated using question templates and the scene graphs as foundation. Each generated question is accompanied by a functional program. The answer set in CLEVR consists of 28 unique entries. Figure 2.10 shows an example image with a few questions in this dataset.

2.4 Shortcut Learning and Generalization

Generally speaking, VQA models solve their task by employing so-called decision rules that they have learned during training. Decision rules essentially describe relationships between a model's input (image and question) and its output (the answer), with the most fundamental decision rule in VQA being the general mapping of image and question to an answer. The choice of a model's architecture (Chapter 2.2), the employed training approach (e.g., Chapter 2.5.3), as well as the used training datasets (e.g., Chapter 2.3 and Chapter 2.5.1), all have a profound impact on the kinds of decision rules a

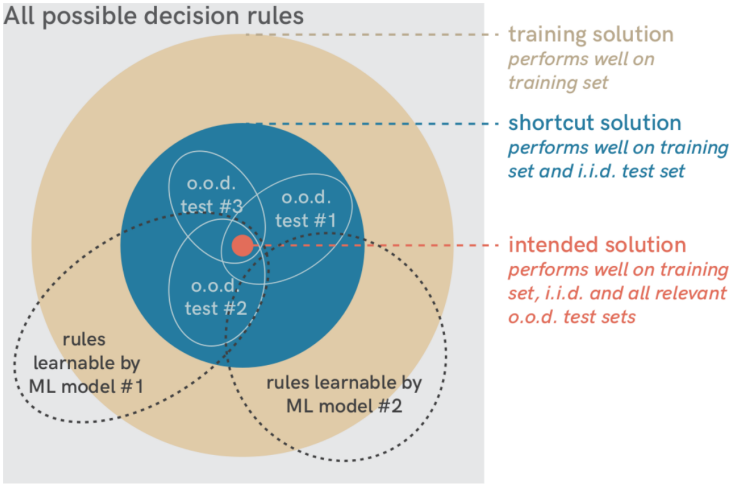

Figure 2.11 – "Taxonomy of decision rules. Among the set of all possible rules, only some solve the training data. Among the solutions that solve the training data, only some generalize to an i.i.d. test set. Among those solutions, shortcuts fail to generalize to different data (o.o.d. test sets), but the intended solution does generalize" (image and caption copied from Geirhos et al. (2020)).

model will acquire in training. One particularly undesirable type of decision rule in VQA (and machine learning in general, cf. Schölkopf et al. (2012); Pfungst and Rahn (1911); Torralba and Efros (2011); Geirhos et al. (2020)) is called the *shortcut*.

Geirhos et al. (2020) defines shortcuts as "decision rules that perform well on standard benchmarks but fail to transfer to more challenging testing conditions, such as real-world scenarios". Following this definition, we describe the process of shortcut *learning* as a model's acquisition of such decision rules in training. We further refer to the successful leveraging of shortcuts in tests as shortcut *exploitation*. Shortcuts are characterized by working particularly well on tests that are independent and identically distributed (i.i.d., also called ID or In-Distribution tests) w.r.t. the training data. This property typically manifests in benchmark datasets that divide all available data samples into train and test set in randomized fashion (e.g., the VQA dataset described in Chapter 2.3).
Shortcut exploitation can significantly impact a model's performance and contribute to high scores on i.i.d. benchmark tests without actually reflecting the model's ability to solve the underlying task, thus giving a false sense of accomplishment. One straightforward way to expose a model's reliance on shortcuts are so-called Out-of-Distribution tests (o.o.d. or OOD tests), which are tests that are constructed systematically different from the original

training & test datasets. Figure 2.11 illustrates the taxonomy of decision rules in the context of solving ID (i.i.d.) and OOD (o.o.d.) test sets.
The field of VQA uses various OOD tests to help uncover shortcut exploitation, some of which we describe in detail in the next section (Chapter 2.5).

Besides *preventing* shortcut learning to manifest in VQA models, *uncovering* shortcut exploitation constitutes an equally important step towards improving VQA generalization. Both aspects have been investigated extensively (e.g., Kervadec (2021); Dancette et al. (2021); Manjunatha et al. (2019); Agarwal et al. (2020); Agrawal et al. (2018)). While, strictly speaking, generalization is not a part of shortcut learning, the two concepts are inherently linked to one another. Shortcut learning is characterized by its interference with a model's adoption of more desirable (human-)intended decision rules that are intuitively understood to solve the underlying task (instead of only the presented dataset). Avoiding shortcut learning and encouraging a model to acquire intended decision rules therefore represents a promising direction in the pursuit of stronger generalization capabilities. By this line of thinking, we can also reasonably assume that an understanding of the extent of shortcut learning in a model can also serve as a meaningful indication of a model's behavior in general real-world scenarios, as well as highlight issues to address to further improve model generalization.

Human-intended decision rules that align with well-understood axiomatic procedures involved in solving the underlying VQA task are likely to benefit a model's performance in generalization scenarios, including OOD settings. One such prominent axiomatic decision rule in VQA manifests as a model's reliance on question-relevant input image regions when inferring an answer to a question, and is sometimes more intuitively described as being "right for the right reasons". This decision rule is called *Visual Grounding*, and we discuss it in detail in the next section.

2.5 Visual Grounding in VQA

Visual Grounding (VG) in VQA measures a VQA system's inherent proclivity to base its inference on question-relevant image regions. Proper[3] VG in

[3]In theory, we can differentiate between *general* VG in VQA, which is characterized by a model's meaningful reliance on *any* visual information during answer inference (as opposed to no involvement of the vision modality), and *proper* VG, which is characterized by a model's meaningful reliance on *question-relevant* visual information. This thesis focuses on investigations of proper VG. Therefore, we only differentiate between "(proper) VG" and "no (general/proper) VG" (i.e., we conflate "no general VG" and "no proper VG" as negative VG cases).

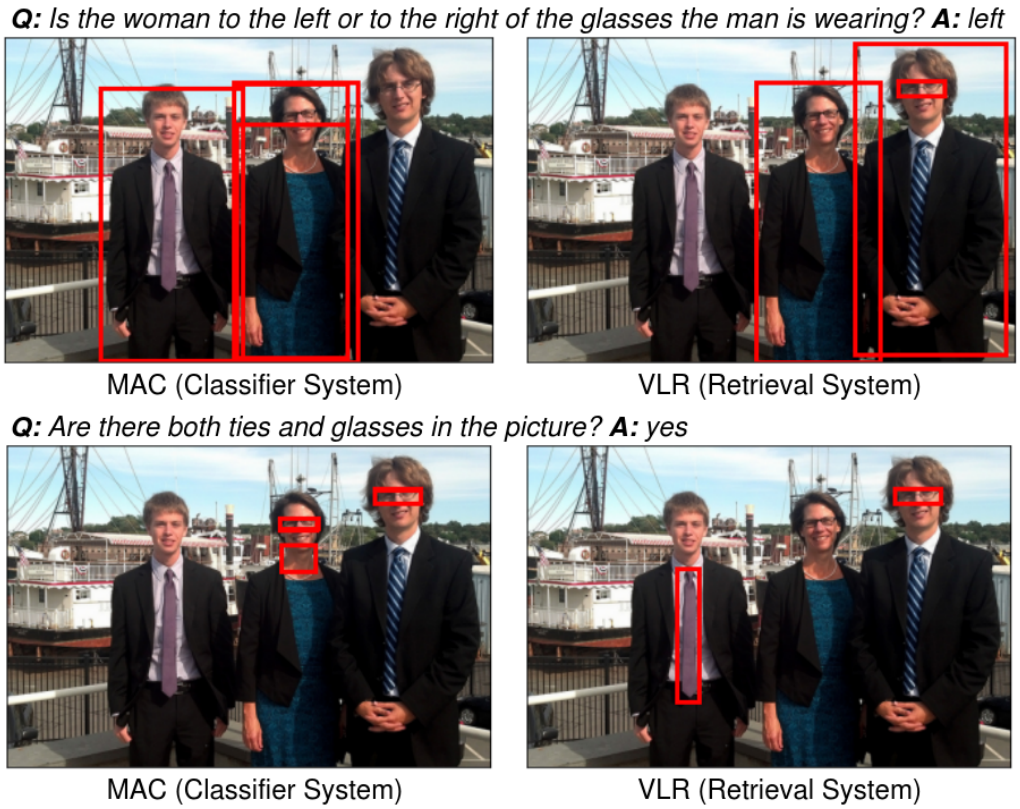

Figure 2.12 – Illustration of Visual Grounding. Model-attended image regions of two VQA systems during inference. Even though both systems return the same correct answer, only VLR's inference (right, introduced in Chapter 3) is correctly grounded on relevant parts of the image, while MAC (left, Hudson and Manning (2018)) focuses on image regions that seem insufficient to inform a correct answer.

a VQA model is defined by a meaningful involvement of question-relevant image regions — or rather the visual features that represent them — in a model's inference process. On a conceptual level, proper VG is an axiomatic requirement of VQA by definition of the VQA task. Figure 2.12 shows an example of how VG quality can manifest differently in two VQA systems. The depicted system on the right (VLR, introduced in Chapter 3) bases its inference on plausibly question-relevant image regions and exhibits "good" VG for the example questions. The system on the left (MAC (Hudson and Manning, 2018)), on the other hand, seemingly fails to do the same, hence exhibiting "bad" VG. Notice how the actual impact of VG quality on VQA performance is not really evident in these given examples, as both systems ultimately output the same correct answers. This phenomenon spawned one

Dataset	#imgs	#questions	Source
VQAv2	205k	1.1m	VQAv1 / Original
GQA-101k (VisFIS)	62k	161k	GQA (balanced)
GQA-large (VisFIS)	82k	1.07m	GQA (balanced)
VQA-HAT (VisFIS)	21k	60k	VQAv1
CLEVR-XAI (VisFIS)	10k	140k	CLEVR-XAI / CLEVR
GQA-OOD	82k	997k	GQA (balanced)

Table 2.2 – Dataset statistics of various ID/OOD dataset splits.

of the overarching questions that VG research in VQA is trying to answer: How does VG impact VQA performance?

Given the axiomatic nature of VG's role in VQA, a disregard of plausibly necessary information to answer a question like those illustrated in Figure 2.12 would be expected to frequently cause a model to arrive at incorrect conclusions. Therefore, correct answers that are consistently produced with low VG quality can reasonably be considered evidence of a model's reliance on various statistical dataset biases and spurious patterns that it learns to exploit during training (cf. Geirhos et al. (2020); Kervadec (2021)). As shortcut exploitation has the potential to severely limit the utility of a model to conditions set by the (training) dataset, there is a natural push in research towards avoidance of their manifestation. A direct evaluation of VG in a VQA model is one way of tracking the extent to which VG-related shortcuts might be exploited in a given dataset. Another more practical and straightforward way to uncover shortcut exploitation is by VQA evaluation with specifically designed test scenarios that aim at reducing shortcut opportunities, thereby revealing this shortcoming in the model by poor performances. We describe both of these types in more detail in the following.

2.5.1 Uncovering shortcut exploitation with OOD evaluation.

Generally speaking, shortcut exploitation works well in test settings where samples originate from the same distribution as the training data (i.e., so-called i.i.d. settings). It tends to fail, however, in generalized settings where samples deviate from these learned biases. Here, the model cannot rely on shortcuts anymore to perform and actual interpretation of relevant input information is required to be consistently successful.

With this issue in mind, various generalized settings have been proposed in an effort to uncover bias and shortcut exploitation in VQA and thereby estimate a model's generalization performance:

- **VQAv2:** Goyal et al. (2017) propose an overhaul of the popular VQAv1 dataset (Antol et al., 2015) that re-balances prior distributions of question-image-answer tuples by including additional images that result in different answers to the same question. This extension of the dataset is intended to reduce language-bias manifestation in particular, as many test questions in VQAv1 could be answered by learning the question-answer priors independent of the image contents. Note, that VQAv2 is not typically employed as an OOD test, but we mention it here for its de-biasing efforts.

- **VQA-CP:** Agrawal et al. (2018) propose an algorithm called "Changing Priors" (CP) to re-distribute samples in the VQAv1/v2 datasets and thereby enforce differences in answer prior distributions in train and test sets for each question type. This results in a new OOD test scenario. VQA-CP's introduction is motivated by reports of problematic VG in VQA models, and OOD testing is proposed with the express intent to encourage VG research in VQA models.

- **VisFIS (various datasets):** Inspired by the CP-based re-distribution approach used in the creation of VQA-CP, Ying et al. (2022) also employs CP to re-distribute samples and create new ID/OOD test splits for the purpose of VG research. New data splits are created for three VQA datasets, namely 1) GQA (balanced) (Hudson and Manning, 2019), 2) VQAv1-based HAT (Das et al., 2016), and 3) a derivative of CLEVR called CLEVR-XAI (Arras et al., 2022).

- **GQA-OOD:** Also based on GQA (balanced), Kervadec et al. (2021) propose an ID/OOD split created with a re-distribution method that categorizes samples based on question type and associated answer frequency in the test set. Here, ID or "head" subsets consist of high frequency answers (given the question type). Low frequency answers are assigned to the OOD or "tail" subset. The training set remains unchanged from the original split with this approach. GQA-OOD is based on the premise that shortcut exploitation should not be able to succeed for questions with rare answers, because rare answers require proper reasoning (which the authors frame as the opposite of shortcut learning, cf. also Kervadec (2021), p. 14). Hence, the "tail" subset represents an OOD generalization scenario.

Statistics of these datasets are listed in Table 2.2.
VQA models have consistently been reported to perform significantly worse in OOD settings compared to the corresponding ID tests, thereby demonstrating the widespread issue of shortcut exploitation in VQA models.

2.5.2 Measuring Visual Grounding in VQA

By OOD testing. OOD tests in general, and specifically the CP-based OOD settings from Chapter 2.5.1, have regularly served as a means to gauge the impact of VG capabilities of VQA models (Selvaraju et al., 2019; Wu and Mooney, 2019; Ying et al., 2022). This evaluation practice is encouraged by (anecdotal) evidence of model behavior in OOD settings which reveals a model's evident insensitivity towards image information (see, e.g., Agrawal et al. (2018); Selvaraju et al. (2019)), thereby reaffirming the interpretation that lack of VG is the main reason behind a model's lower OOD performance and that strengthening VG must lead to improved OOD generalization. In this vein, gains in OOD accuracy have similarly been interpreted as confirmation that improvements to VG are causing these gains (e.g., Selvaraju et al. (2019); Wu and Mooney (2019)), although such conclusions have since been shown to be unreliable (Shrestha et al., 2020).
It is worth noting that even though VG research for VQA models has been the main purpose behind the introduction of several of the listed OOD tests, the methods that are used to create these tests do not actually focus on selecting questions *specifically* based on their VG requirements. It turns out that redistribution methods that only focus on disaligning answer priors in train and OOD test naturally expose some VG-related shortcut learning, but, they far from isolate the impact of VG-related shortcut exploitation on VQA performance. These insights are part of the main contributions of this thesis and we present and discuss them in great detail in Chapter 8.

By dedicated VG metric. Aside from using ID/OOD accuracy evaluations to estimate VG strength in VQA models, other more specialized metrics and measurement techniques have been proposed to quantify VG in VQA systems. Dedicated VG metrics broadly follow two approaches:

(1) Determining and comparing visual input feature importance (FI) scores which measure the relevance of input features for answering a question. Model-based FI-scores are compared with a reference, i.e., annotation-based FI-scores.

(2) Running multiple VQA evaluations with modulated visual feature inputs and comparing model outputs to determine the importance of features to the model.

Examples of VG metrics that belong to approach (1) use FI-scoring methods that are based on a) a model's internal attention mechanism (Bahdanau et al., 2015) (e.g., as part of the VG metric proposed in Hudson and Manning (2019)), or b) gradient-based FI-scores determined by GradCAM (Selvaraju et al., 2017) (e.g., as part of the VG metric proposed in Shrestha et al. (2020)). In feature modulation methods (approach (2)), VG quality is determined by comparing model behavior for selective permutations of input features to estimate their importance to the model's answer inference in an indirect way. Here, the reference (i.e., the annotation-based FI) typically guides the decision on which feature permutations to evaluate (e.g., DeYoung et al. (2020), adopted for use in VQA in Ying et al. (2022)).
In Chapter 4, we discuss VG metrics in greater detail and introduce our own VG metric called FPVG (Faithful & Plausible VG), which is categorized as a feature modulation method.

2.5.3 Improving Visual Grounding in VQA

The concept of VG quality enhancement in VQA models constitutes a straightforward path to fundamental improvements w.r.t. shortcut exploitation. In Chapter 2.2, we have already seen various VQA model designs that evolve the way of how visual information is processed during answer inference. In many cases, these developments can be interpreted as an implicit strife for better VG quality. Some models address such motivations more explicitly by reporting VG measurements (PVR, Li et al. (2019a)) and/or including specialized auxiliary training objectives that focus on VG-related advancements in their models (MMN, Chen et al. (2021)). For the vast majority of models, however, no deeper investigations are presented into how well VG has manifested as a result of the proposed model architecture. Architectural evolution has been primarily driven by overall VQA performance improvements as measured by answer accuracy, without explicit considerations for generalization scenarios (like OOD settings) or VG quality (as is evident by a general lack of VG evaluation in model reports). This research routine has consequently given rise to retrospective adjustments that attempt to improve an already existing model's generalization performance by application of special training paradigms that help strengthen a model's VG quality. In the following, we describe two such influential training approaches, which we will use in later chapters of this thesis.

HINT. Human Importance-aware Network Tuning, or HINT (Selvaraju et al., 2019), attempts to strengthen VG by aligning model-determined feature importance (FI) scores of visual inputs with those determined by human annotations.

Model-based FI-scores are calculated by a variant of GradCAM (Selvaraju et al., 2017), which is an FI-measurement technique originally proposed in the context of image recognition. The importance of a visual input object r for ground-truth answer α^r_{gt} is calculated as:

$$\alpha^r_{gt} = \sum_{i=1}^{|F|} \frac{\partial o_{gt}}{\partial F^r_i}, \tag{2.4}$$

where the sum accumulates the gradients of the ground-truth answer loss o_{gt} w.r.t. visual input object r's features F^r. The gradients are summed in order to obtain a single importance score per visual input object, as opposed to one score per feature dimension.

Human-based FI-scores for each given visual input object are determined by a function involving overlap of the detected bounding boxes of visual input objects with human-annotated question-relevant regions in the image. The higher the calculated overlap of the detected object with the annotated relevant image region, the higher the resulting human importance score for the detected object. Note that the process for determining human-based FI-scores is a central theme of our investigations in Chapter 7.

The two resulting importance scores are then compared by aligning their internal rankings and accumulating score differences where object ranks differ. The resulting **ranking loss** is then finally added to the regular VQA answering loss as a weighted term.

HINT was shown to have significant accuracy impact on the OOD test in VQA-CP, but further analysis in Shrestha et al. (2020) revealed that improvements were not actually the result of VG improvements and were instead attributed to a form of training regularization. We discuss further insights for HINT as a result of our own analyses in later chapters.

VisFIS. Visual Feature Importance Supervision, or VisFIS (Ying et al., 2022), is an ensemble of VG-related training objectives that are calculated alongside the common answer-class-dependent cross-entropy (CE) loss function in VQA model training. VisFIS consists of the following four training objectives:

1. *Sufficiency.* The idea behind this objective function is that visual input consisting of only relevant objects is expected to still be able to

inform the correct answer output. It is calculated as a regular CE-loss for training samples with modified visual input that only represents question-relevant objects.

2. *Uncertainty.* The idea here is that questions accompanied by visual input consisting of only irrelevant objects should not produce a confident answer. It is calculated as the Kullback-Leibler divergence between 1) the uniform answer class output distribution of the VQA model, and 2) the model's output distribution for training samples with visual input that only consists of irrelevant objects.

3. *Invariance.* Question-irrelevant visual input objects should be of low importance to the model's decision when measured with FI-scoring methods. This objective is calculated as an L1 loss that penalizes the model for high weights on irrelevant visual input objects.

4. *Alignment.* Model-based FI-scores should align with annotation-based FI-scores. Accordingly, this objective is calculated as a cosine similarity loss function, comparing importance valuations of visual input objects given by 1) annotations, and 2) an FI method of choice.

Accuracy improvements for this method on three newly proposed OOD tests (described in Chapter 2.5.1) have been reported stronger than for all other evaluated VG methods, including HINT.

2.5.4 VG in VQA vs. other research areas

The definition of Visual Grounding in the context of VQA may differ from its definition in other fields. If adopted to the field of VQA, such definition differences can have a profound impact on how VG is measured and what kind of influence it ultimately has on VQA modeling and model performance. An awareness of the definition of VG in each given body of work is therefore of prime importance. As described in Chapter 2.5, in this thesis, we define (proper) VG as a model's meaningful reliance on (question-relevant) visual features in its process of answer inference.
As a notable example of a different definition, we make reference to research in the field of Referring Expressions (RE) (Kazemzadeh et al., 2014). Unlike in VQA, in RE research, VG is not interpreted as a measurement of *model-intrinsic* quality. Rather, it is defined in the context of explicit localization of task-relevant image regions. This marks a subtle but crucial difference to VG in VQA, where VG describes a model's *intrinsic* characteristic of *relying* on relevant image regions for inference of the answer.

It is worth pointing out that even within the field of VQA VG is sometimes reduced to the aforementioned aspects of RE research, and is thus sometimes being framed as a localization task. Notable examples can be found in the context of VQA applications for the visually impaired (Chen et al., 2022), as well as VQA models that are explicitly trained to localize relevant objects independent of their direct involvement in answer inference, like in MMN (Chen et al., 2021). While improving VQA models by some involvement of a localization task during training may also have a positive impact on VG in VQA (i.e., VG defined as an intrinsic quality of the model), the subtle differences in goals of these tasks still need to be carefully noted in order to determine suitable evaluations and correctly assess resulting model behavior.

2.6 Summary

In this chapter, we established this thesis' background in the field of Visual Question Answering, Visual Grounding and shortcut learning. Many of the introduced principles, methods and datasets will be referenced in later chapters.

Building on these existing foundations, we introduce our own contributions and additions to the body of research in VQA and VG in the remainder of this thesis.

Part II

Methods

Introduction

In Part I, we provided background information on the field of Visual Question Answering, Visual Grounding and related research in the area of shortcut learning, as it pertains to this thesis. In Part II, we describe our development of a number of novel methods and processes that we leverage for performing a thorough analysis of VG's role in VQA generalization and shortcut learning. Part II is organized as follows:

In Chapter 3, we introduce "VQA by Lattice-based Retrieval" (VLR), a VQA system that follows an information retrieval-based design. VLR's implementation closely aligns with human-intended decision rules for VQA and involves a heavy reliance on VG.

In Chapter 4, we fill the need for a dedicated, meaningful and accurate metric to measure faithful and plausible VG in VQA. We introduce our VG metric "Faithful and Plausible Visual Grounding" (FPVG) and verify its properties and advantages over other existing metrics in a series of experiments.

In Chapter 5, we describe the construction of symbolic visual features for VQA, which enable us to gain insights into model behavior related to informational content carried in the visual modality. We further introduce a procedure we call "Information Infusion" to easily manipulate image content through surgical feature modifications, which we rely on in Part III of this thesis.

The following publications share results and content with this part:

- Visually Grounded VQA by Lattice-based Retrieval (Reich et al., 2022)
- Measuring Faithful and Plausible Visual Grounding in VQA (Reich et al., 2023)
- Uncovering the Full Potential of Visual Grounding Methods in VQA (Reich and Schultz, 2024)

Chapter 3

Visually Grounded VQA by Lattice-based Retrieval

In this chapter, we describe VLR ("**V**QA by **L**attice-based **R**etrieval"), a VQA system that breaks with the dominant VQA modeling paradigm of classification and investigates VQA from the standpoint of an information retrieval task. As such, VLR directly ties VG into its core search procedure. VLR operates over a weighted, directed, acyclic graph, a.k.a. "lattice", which is derived from the scene graph of a given image in conjunction with region-referring expressions extracted from the question. VLR's conception is primarily motivated by our investigations into the role of VG in VQA. In taking a step back from high-performing classification-based black box designs, we aim to get a clearer understanding of the interplay of fundamental components of a VQA system through empirical investigations of and with a system that is conceptually employing the kind of visually grounded reasoning that classification-based VQA systems are expected to learn and exhibit. As such, VLR represents a reference system that inherently prevents excessive shortcut exploitation from developing by more closely enforcing human-based decision rules for VQA.

We give a detailed analysis of this approach and discuss its distinctive properties and limitations. Aside from its usefulness in VQA analysis, VLR — as a practical VQA system — also displays the strongest VG characteristics among examined systems and exhibits exceptional generalization capabilities in a number of related scenarios.

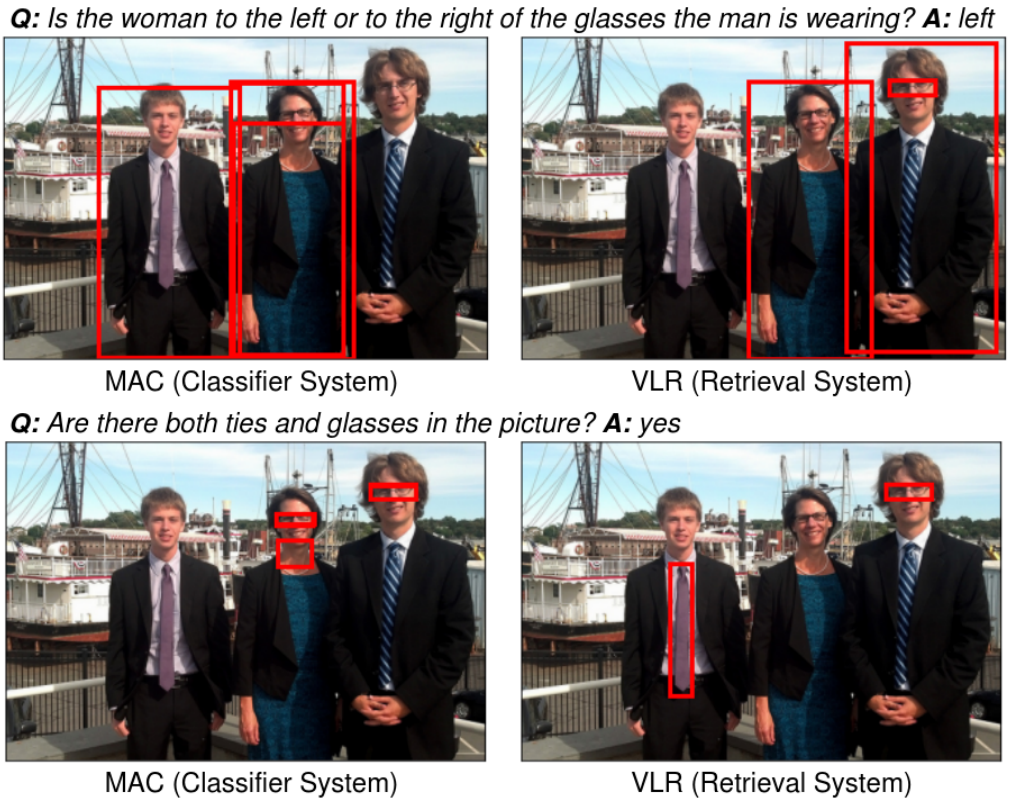

Figure 3.1 – Illustration of Visual Grounding. Attended image regions of two VQA systems during inference. VLR's inference is correctly grounded while MAC focuses on parts that seem insufficient for producing the correct answer (which both systems do).

3.1 Introduction

A VQA system requires proper handling of two types of inputs: 1) language, which encodes the query we seek to answer, and 2) images, which encode the search space for the query and act as knowledge base storing the answer. A third piece that completes this task is an inference engine that models the interaction between the two modalities and enables extraction of an answer. In this context, Visual Grounding (VG) can be described as a measure of how well the inference engine manages to tie region-referring expressions given in the question to relevant regions in the image and, consequently, produces an answer that is in fact based on these very regions. Systems with strong VG are considered intuitively interpretable (see Figure 3.1) and suggest heightened scene understanding. Improving VG in VQA makes models appear more trustworthy (Selvaraju et al., 2019) and can improve accuracy (e.g. Ying et al.

(2022)), while VG that fails to manifest regularly can foreshadow a system's struggles in certain scenarios as we find in Chapter 3.4.4 and later again in Chapter 6.

Some highly influential milestones in VQA system designs (e.g. Anderson et al. (2018); Yang et al. (2016); Hudson and Manning (2019)), have been successful with designing their system's inference procedure closer to that of a human's inference process. While these designs might in theory implicitly enable a system to learn visually grounded inference, explicit efforts to improve VG performance and evaluation thereof have in practice taken a back seat on the road to overall accuracy improvements on popular VQA benchmarks such as Antol et al. (2015) and Hudson and Manning (2019). This is presumably because the involved test scenarios cannot appropriately reward a model's adherence to human-intended decision rules, such as VG, with improved accuracy and a higher benchmark ranking (see also Chapter 2.4 for additional background).
Traditionally, VQA system designs have focused on creating a powerful discriminative classifier, often in the shape of an elaborate deep neural network, which is trained by minimizing an answer performance-related loss function like cross-entropy. Learning to produce correct answers via discriminative classification over a predefined answer set allows these models to identify complex patterns in the training data that help them select the correct answer. However, as image processing methods are still performing imperfectly in complex real-world scenarios and thus only produce sub-par image representations that VQA models have to use as (inaccurate) knowledge base to reason over, VQA systems may have to forego correct visual grounding for (seemingly random or unreasonable) patterns, or shortcuts, that will lead to correct answer selection (e.g. Figure 3.1). This line of reasoning suggests that building models with strong accuracy *and* strong VG, or VQA models that are "right for the right reasons", is harder to accomplish than focusing on strong accuracy alone — especially for classifier-based systems.

Accuracy has long been the primary performance metric driving development of novel VQA systems, with strong VG being a "nice-to-have" property that is often overlooked in the presence of improved accuracy. In this chapter, we take a step back of accuracy-driven VQA system development and put our primary focus on implementing a VQA system structure that prioritizes correctness of human-intended decision rules in general and VG in particular.

To this end, we address the challenge of "right for the right reasons" and design a VQA system that puts VG quality in the inference and answering process center stage. We break with the dominant VQA modeling paradigm

Figure 3.2 – VQA system designs (simplified): The predominant classification design (r) and our lattice-based retrieval approach (l).

of classification and investigate VQA from the standpoint of an information retrieval task (Figure 3.2). Here, we frame the retrieval process of our system as a (scene) graph search problem and look to the field of Automatic Speech Recognition (ASR) for solutions, which has a long history of employing graph search algorithms for speech decoding. We adopt a concept that is integral to the ranking of recognition hypotheses in ASR: the word lattice. Using symbolic features detected in the image (and represented in a scene graph), as well as inference instructions extracted from the question, we construct a weighted, directed, acyclic search graph, a.k.a. lattice. In this VQA-lattice, each path represents an alternative sequence of salient image regions, weighted by their recognition scores for object/attribute/relationship identities. The exact make-up of these scores depends on the region-referring expressions extracted from the question. Inspired by search procedures in ASR once more, we rank the paths through the VQA-lattice according to their likelihood using the Viterbi algorithm, and finally extract an answer via deterministic logic based on the 1-best path.

Approaching the VQA task in the manner described above manages to tie VG principles directly into VLR's core search procedure for the answer, creating a highly visually grounded VQA system as evidenced by evaluations on the compositional VQA focused GQA dataset (Hudson and Manning, 2019). Moreover, following a retrieval-based paradigm with focus on incorporating human-intended decision rules enables VLR to successfully handle anti-shortcut generalization challenges that classifier-based systems heavily struggle with.

Contributions. We summarize the contributions of this chapter as follows:

- We propose a conceptually new VQA approach that is motivated by a strife for strong VG through assimilation of human-based decision rules, named "**V**QA by **L**attice-based **R**etrieval" (VLR).
- We show that VLR achieves significantly stronger intrinsic VG quality than reference systems of various designs.

- We examine VLR's distinctive strengths in various newly developed generalization and Out-of-Distribution scenarios, showing it is particularly well equipped for real-world deployment where such challenges are encountered and exploitation of shortcuts is expected to be less beneficial.
- We share the newly developed generalization tasks with the research community to encourage development in this direction[1].
- VLR sets itself apart from other VQA systems by employing a quasi open-vocabulary answer production that is not restricted to a specially learned, pre-determined answer vocabulary set.

3.2 VQA System Designs

In this section, we discuss VLR's similarities with existing models in VQA, as well as designs that are influenced by an explicit motivation to improve VG.

General designs. VLR follows a straightforward modular system design that shares some resemblance with existing VQA approaches. In particular the isolation of the question interpretation process has been realized in related models by learning to produce a program sequence from the question which is subsequently leveraged to realize an interaction with the input image features (Hu et al., 2017, 2018; Mascharka et al., 2018; Johnson et al., 2017; Li et al., 2019a; Zhao et al., 2021; Chen et al., 2021). Our approach differs from these models in particular in the choice of image representation: we operate on scene graphs and use symbolic features instead of (spatial) sub-symbolic visual features.
Like VLR, other recent approaches increasingly employ scene graphs as image representation due to their effectiveness in compositional VQA tasks (Hudson and Manning, 2019; Hu et al., 2019; Liang et al., 2021; Shi et al., 2019; Kim et al., 2020; Li et al., 2019b). VLR shares similarities with the neuro-symbolic methods NSM (Hudson and Manning, 2019), PVR (Li et al., 2019a) and NS-VQA (Yi et al., 2018) in particular. However, while NSM is trained end-to-end and does not have an explicit mechanism to produce and rank inference paths, VLR is a fully modular system with inference path production and ranking capabilities. And, unlike NS-VQA, which uses a discrete structural scene representation, and PVR, which uses region-based sub-symbolic visual features, VLR operates on a probabilistic, symbolic scene

[1] https://github.com/dreichCSL/GQA_generalization_splits

graph representation as its knowledge base.
With the exception of NS-VQA, all of the above systems have in common an answer production process that determines the output answer by means of a discriminative classifier, defined over a preset answer vocabulary. NS-VQA is different in this regard, as it queries a discrete, structured database for answers in an artificial, constricted scenario (i.e., the CLEVR diagnostic dataset described in Chapter 2.3.3). All mentioned systems are trained using an objective function that serves to directly improve answer performance on the training dataset, which makes them prone to manifest shortcut learning (Geirhos et al., 2020).
Finally, VLR's retrieval-based design resembles DFOL (Amizadeh et al. (2020a), see also Chapter 2.2.6), which proposes a formalism based on predicate logic for (neuro-)symbolic reasoning in VQA. Our work distinguishes itself from DFOL in particular by 1) the introduced graph search concepts that originated in ASR, 2) our more sophisticated Question Parser, and 3) our explicit conceptualization of VLR as a retrieval-based VQA system enforcing human-based decision rules to avoid shortcut learning. Combined, these design differences make VLR more capable of successfully handling a wider array of generalization scenarios than DFOL (see also Chapter 3.4.4 and Chapter 3.4.4).

VG-motivated models. Various systems focus on improving VG performance in the context of the GQA dataset, which we use as reference points in our evaluations: The module-based approach PVR (Li et al., 2019a) improves VG by deriving explicit inference instructions (similar to our work's Question Parser) and including additional supervision for strengthening correct grounding during training of their classifier. Similarly, MMN (Chen et al. (2021), description in Chapter 2.2.7) also uses a Question Parser and employs a scored alignment between predicted object importance and annotated relevant objects in each inference step as additional learning signal. MAC-Caps (Urooj et al., 2021) proposes to use visual capsule modules (Sabour et al., 2017) for question-conditioned selection of detected image properties to improve VG performance.

3.3 System Description

In Chapter 1.1.2, we explored how framing VQA as a retrieval task can help us to intuitively understand the aciomatic necessity of VG to the VQA task for producing correct answers. We draw on this insight to motivate VLR's system design. When framing VQA as a retrieval task, system modularity

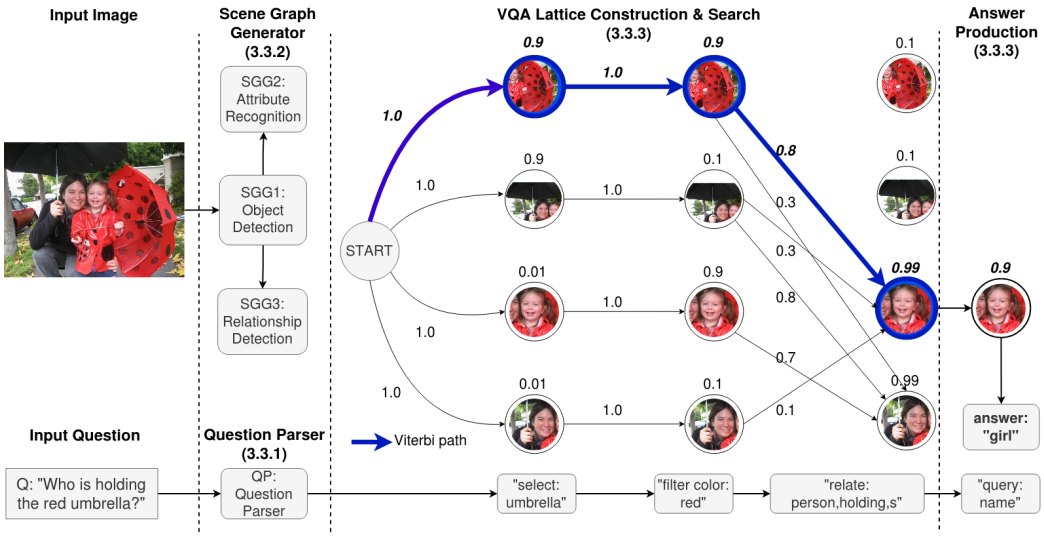

Figure 3.3 – Overview of our system's components. VLR consists of a number of modules and steps (in bold), described in detail in Chapter 3.3. The Scene Graph Generator generates a probabilistic scene graph that acts as VLR's visual knowledge base (Chapter 3.3.2). The Question Parser (pictured at the bottom) parses the input question into queries (Chapter 3.3.1). The VQA lattice is constructed by extraction of the queried probabilities from the scene graph. Once the lattice is constructed, the best path through the lattice is determined using the Viterbi algorithm. Finally, the answer is produced based on the final object of this path (Chapter 3.3.3).

presents itself as an obvious design choice: both the question and image can be processed independently from each other to generate the query and knowledge base, respectively. The interaction between the two, i.e. the actual retrieval process, can then be handled by a third component. Following this outline, we build three modules that together make up VLR:

1. Question Parser: Parses the question into standardized query format.
2. Scene Graph Generator: Generates a scene graph from the input image which acts as the knowledge base.
3. Rank & Answer: Executes the query on the knowledge base and produces the answer.

An overview of the entire system as well as the involvement of each module during processing of an example question is shown in Figure 3.3. In the following sections, we describe each of the three modules and VLR's answering process in detail.

3.3.1 Question Parser

The Question Parser (QP) parses the question into a sequence of operations (also called "program", cf. Hudson and Manning (2019); Yi et al. (2018)) that are subsequently used in the construction of the VQA-lattice. In VLR, an "operation" is essentially a query to the scene graph to acquire specific recognition scores which are used in the construction of the VQA-lattice as node emissions or transition probabilities (see Chapter 3.3.3 for details). We use GQA-provided operation sequences (op-seqs) for training and evaluation of the QP. An "operation" is a tuple given as (operation type, argument). Here, operation types take on values such as "select", "filter" or "relate", while arguments consist primarily of names for objects, attributes and relationships, as well as some functional symbols like logical-or and underscore (where the latter represents a wildcard entry). We determined 136 unique operation types in GQA, and a much higher number of arguments. These can be combined in various ways to form a myriad of unique operation tuples.

Model Description. The QP DNN is based on a pointer-generator architecture with "coverage" mechanism, similar to See et al. (2017). This network is, at its core, a seq2seq encoder-decoder model with attention mechanism and certain extensions that enable it to "copy" a token from the input to the output sequence. The "coverage" mechanism (Tu et al., 2016) tracks to what degree input words have already been involved in the production of previous output elements in the sequence. Integrating "coverage" in training helps to significantly reduce token repetitions in the output sequence, which is a common issue in networks that employ a copy mechanism. The implementation of this architecture is motivated by two concrete goals:

1. Reduced data requirements: We avoid having to train an unnecessarily large output layer for generating all possible op-seq variants. Since an operation tuple can refer to any object, attribute and relationship by name, the output vocabulary of a vanilla seq2seq model without a copy mechanism would have to be much larger to cover all possible entity names. This would present a more challenging learning problem with higher data requirements, as well as stronger dependency on exhaustive program annotations.

2. Open vocabulary: The copy functionality of the QP adds the capability to handle previously unseen entity names (e.g. new object names) reasonably well (see Chapter 3.4.4), whereas a vanilla seq2seq model (e.g., used in Yi et al. (2018); Amizadeh et al. (2020a); Zhao et al.

(2021)) would not be able to output any word outside of its preset vocabulary.

These advantages of the pointer-generator network over a vanilla seq2seq network as basis for VLR's QP directly benefit VLR's generalization capabilities, which coincides with our motivation for minimizing opportunities for shortcut learning.

Model Specifications and Training. We include essential information regarding QP's training process here and defer to Appendix B.1 for additional details, including results of isolated evaluations of this module.

As model input to the QP, we use pre-trained GloVe word embeddings (Pennington et al., 2014) to encode words in the question. The softmax output layer of the model consists of 162 classes, which includes a number of classes (20) representing pointers to input question word positions that are accessed by the copy mechanism. To train the copy mechanism, we leverage a combination of regular expressions and GQA's annotated pointers (from question words to operation arguments). These are used to determine whether or not a token in the output program sequence should be a pointer to a certain question word.

3.3.2 Scene Graph Generator

The Scene Graph Generator (SGG) produces a graph-based, semantic representation of the raw image. This resulting scene graph represents detected objects (alongside their attributes) as nodes and relationships among objects as edges between them (cf. Johnson et al. (2015)). An image's scene graph represents the visual knowledge base that informs VLR's reasoning process. In the following, we provide key information about the structure of VLR's SGG and defer to Appendix A for detailed implementation details, as well as results of isolated evaluations of each of the three underlying tasks (i.e., object, attribute and relationship detection).

The SGG is divided into three sub-modules that each handle object detection, attribute recognition and relationship detection, respectively. A Faster R-CNN model (Ren et al., 2015) handles object detection and (sub-symbolic) visual feature extraction. This model's softmax output distributions (a 1702-dim vector for 1702 object classes per detected object) is used to populate the scene graph for an image. Each object's attributes (617 classes), as well as relationships among detected objects (310 classes), are separately determined by their respective modules. These modules consist of individual models

which produce class distributions that are similarly included in the scene graph to populate attribute and relationship information for each object.

An abstract depiction of the construction of a scene graph by our SGG is shown in Figure 3.3, second vertical section from the left, which illustrates that relationship and attribute recognition is conducted based on outputs of the used object detector.

3.3.3 Rank & Answer

The Rank & Answer module (R&A) is responsible for determining the top ranked visually grounded inference paths and determining an answer to the given question. R&A consists of three parts:

1. VQA-lattice construction
2. Finding/ranking paths through the lattice
3. Producing the answer based on the best path

For each question, VLR first constructs a VQA-lattice which can be described as a question-driven reorganization and condensation of the original scene graph. After ranking paths through this lattice by (visual) likelihood given the query, the most likely path is used to determine the answer in a rule-based fashion. The visual objects traversed in this most likely path represent VLR's Visual Grounding w.r.t. the question, i.e., these are the objects that VLR relied on to arrive at the answer.
We take a detailed look at each of R&A's parts in the following.

Lattice Construction. An Automatic Speech Recognition (ASR) lattice is defined as a "weighted, directed, acyclic graph in which each complete path represents an alternative hypothesis, weighted by its recognition score for a given utterance" (Ljolje et al., 1999). Accordingly, we define a VQA-lattice as a weighted, directed, acyclic graph, where each complete path represents an alternative sequence of regions (objects) ending at the final answer region in an image, weighted by the image-based recognition scores from the scene graph as queried by the question. The source of recognition scores used in the VQA-lattice is determined by queries extracted from the question (i.e., the program generated by the QP). These queries - or operation tuples - which consist of an "operation" and an "argument" (cf. Chapter 3.3.1), specify what the nodes in the scene graph should be queried about. For instance, the QP-extracted operation tuple ("select", "apple") would query each node in the scene graph about its object recognition score for "apple". Similarly, ("filter

color", "green") would query each node about its attribute recognition score for "green". Queries about whole object categories like "animal" or "furniture" are handled by summing the recognition scores for all classes belonging to the category. Negations (e.g. "not white") are handled by subtracting the recognition score of the negated class from 1. Finally, operations involving relationships query the relationship-specific edges between any two nodes in the scene graph.

Rank. Once the lattice is constructed, the search task can be defined as finding the path through the lattice that maximizes the probability of the object/attribute/relationship detection models when applied to regions (objects) in the image according to the operations given by the QP. In contrast to ASR, where the goal is to find the maximum likelihood sequence of words given an audio signal, in VQA we want to find the maximum likelihood sequence of regions in an image, given both the image and a language-based description of the regions of interest (as given by the QP). We accomplish this, in accordance with ASR, by using the Viterbi algorithm, which is given as:

$$V_{0,r} = P(q_0|r) * \pi_0 \tag{3.1}$$

$$V_{t,r} = \max_{x \in I} \left(P(q_t|r) * a_{x,r} * V_{t-1,x} \right) \tag{3.2}$$

where t is the current inference step (=operation tuple), r and x are individual regions in the image, I is the set of regions in a given image, $P(q_t|r)$ is the conditional probability of an image region r matching a language-based region description q from the QP (e.g., object or attribute identity) and a is the transition probability from image region x to r (i.e., a specific relationship between them). Like object or attribute scores, transition probabilities between image regions depend on the QP program and are provided by the relationship model (in SGG). If the current program step describes a relationship between objects, the probability for the specified relationship is extracted from the scene graph. If the current program step does not involve relationships, $a_{x,r}$ is set to 1 if and only if x=r and 0 otherwise. This forces the algorithm to stay with the same image region when processing subsequent attributional or positional descriptions of a queried object (e.g., a "car" that is "red").

The Viterbi path, i.e. the most likely sequence of image regions given the image and the question, is then retrieved from back pointers that remember the identity of the chosen region x in Equation (3.2):

$$x_T = \max_{x \in I} V_{T,x} \tag{3.3}$$

$$x_{T-1} = BkPtr(x_t, t) \tag{3.4}$$

with *BkPtr* being a function that returns region x used in Equation (3.2) for $t > 1$ or r if $t = 1$.

As is typical in information retrieval approaches, VLR can also produce a rank ordered list of the best path candidates after a search over the VQA-lattice. VLR accomplishes this by implementing the parallel List Viterbi Decoding Algorithm (LVA) presented in Seshadri and Sundberg (1994), which determines the n-best Viterbi paths in a given lattice. Note, however, that this feature has no concrete use-case in the presented work.

Answer. VLR's answer production module uses rule-based logic and contains no learnable parameters. To produce the answer to the question, the system depends on the final region(s) of the 1-best Viterbi path(s).
GQA splits QA-pairs into five structural categories: query, choose, compare, logical and verify. Among those categories, query type questions denote "open" vocabulary questions (with a large number of possible answers), whereas all other types constitute "binary" questions (with either "yes"/"no" answers, or the answer given as one of two options presented as part of the question).
VLR's extraction of the answer for "query"-type questions, is a simple query to the scene graph for the class with the largest softmax score[2] in the object's class distribution (e.g., an object or attribute name defined in the scene graph). This query to the scene graph might be restricted to subsets of object (or attribute) classes, depending on QP-determined operations. For instance, the QP-determined op-seq might have restricted the search to the object category of "furniture" before querying the name of the final object in the scene graph, which allows a search space reduction to qualifying object names for the answer.
Some questions require the construction of two separate lattices and paths (e.g., "logical-and" type questions asking about existence of two separate objects). In these cases VLR produces one 1-best Viterbi path per lattice and then applies the final operation considering the ending nodes of both paths to answer the question.
Lastly, "verify"-type binary questions, which query object existence in the image, are answered by comparing the geometric average of all multiplied probabilities in the 1-best Viterbi path with a threshold value, which is determined on a small development set.

[2]Remember that in the scene graph, objects, attributes and relationships are all represented by softmax distributions over possible classes (=names). E.g., in case of object classes, the softmax distribution size is 1702, i.e., one value per object name found in GQA.

System	Accuracy			VG
	Binary	Open	Overall	Grounding
N2NMN (Hu et al., 2017)	n/a	n/a	n/a	55.44
UpDn (Anderson et al., 2018)	74.60	47.30	60.51	94.42
MAC (Hudson and Manning, 2018)	77.90	48.37	62.66	94.90
PVR (Li et al., 2019a)	80.67	49.29	64.47	97.44
MMN (Chen et al., 2021)	**81.89**	**50.92**	**65.91**	98.22
DFOL (Amizadeh et al., 2020a)	67.55	44.74	55.78	114.37
VLR	69.94	46.17	57.67	**128.41**

Table 3.1 – Accuracy per question type and overall VG results on GQA's balanced validation set, sorted by Grounding. Higher is better in all columns. For detailed descriptions of the used metrics, see Chapter 3.4.3 and Chapter 3.3.3. N2NMN and PVR results are taken from Li et al. (2019a) and use different visual features. All other models use visual features produced by VLR's SGG.

In addition to the answer, VLR also returns the (ranked) Viterbi path(s), providing the user with a highly transparent view at the inner workings of the QA-process.

3.4 Experiments

In this section, we evaluate VLR's potential in aspects of answer accuracy, VG quality (measured with multiple VG metrics described below) and generalization performance using specially developed test scenarios.

3.4.1 Ablation-type Study of VLR

In Appendix B.3.1, we conduct a detailed ablation-type study of VLR. We quantify the impact of inaccuracies in each of VLR's modules on VLR's overall VQA performance. Starting from a VLR setting that uses only ground-truth inputs (scene graph and question program), we systematically, module by module, replace these ground-truth inputs with our module-predicted inputs. Among other insights, we find that improvements to the scene graph generator show the biggest potential for boosting VLR's performance. We defer to Appendix B.3.1 for additional results and discussions.

3.4.2 General Evaluation

Preliminaries. We report results for a number of VQA systems on GQA's balanced split in Table 3.1. Accuracy is categorized by QA-type (binary or open). The VG metric used here is GQA's "Grounding" metric (more details in Chapter 3.4.3).
With exception of N2NMN and PVR (for which results are taken from Li et al. (2019a)), we trained all models with the same VLR-produced 1024-dim region-based visual features for a fairer comparison. As mentioned in Chapter 3.2, MMN, PVR and DFOL use modular program generators that separately parse the question into a sequence of operations, similar to VLR's Question Parser.

As VLR and DFOL share certain similarities, comparisons between the two systems are of particular interest. Unfortunately, DFOL's officially shared implementation does not contain code for their program generator and instead uses ground-truth programs from GQA. Therefore, results for DFOL need to be taken with a grain of salt, as they might be somewhat inflated. We can gauge the positive accuracy impact that may have resulted from using ground-truth programs by looking at an equivalent setting in VLR's model ablations in Appendix B.3.1 (Table B.3.1, Systems VLR vs. VLR-2). Here, accuracy shows an absolute improvement of around 2.5%. We surmise that accuracy we report for DFOL might include a similar boost from using ground-truth programs. Results for DFOL reported with other metrics might also be similarly impacted.

Results discussion. As shown in Table 3.1, VLR's accuracy performance falls behind most of the reference systems, but surpasses all of them in VG quality. Given that there is no direct optimization of answer accuracy via minimization of a pertinent loss function involved in VLR, this somewhat lower accuracy is not surprising. However, comparing VLR to the similar DFOL model (which, contrary to VLR, does involve model optimization guided by answer performance), we find that VLR achieves somewhat higher accuracy, even without accounting for DFOL's accuracy advantage from using ground-truth programs.

In contrast to the expected accuracy disadvantage compared to classifier-based VQA models, VLR's architectural focus on implementing human-intended decision rules with intrinsic reliance on VG quality is reflected clearly in its dominating VG performance. We take a deeper look at this development as well as VLR's answer performance in various generalization scenarios and Out-of-Distribution (OOD) settings in more detail below.

3.4.3 Visual Grounding Evaluation

VG Metrics

Our primary VG quality evaluation is performed with two attention-based VG metrics:

1. The official "Grounding" metric from GQA[3]: The "Grounding" score represents the average VG score over all questions. A question's VG score is calculated as the sum of score contributions of each visual input object (here: up to 100 detected objects). An input object's score contribution is calculated as its bounding box overlap percentage with each annotated question-relevant reference object, multiplied with the input object's assigned attention score. Formally:

$$VGscore(q_i) = \sum_{j}^{o \in O_{q_i}} \sum_{k}^{r \in R_{q_i}} (overlap(o_j, r_k) * attention(o_j)) \tag{3.5}$$

$$Grounding = \frac{1}{n} \sum_{i}^{n} VGscore(q_i), \tag{3.6}$$

   where n is the number of questions, q is the question, o is a visual input object in the set of all detected objects O_q for the given image, and r is a reference object in the set of all question-relevant reference objects R_q.

2. A generic IoU-based metric[4]: For each detected input object, we check if it has an IoU > 0.5 with any annotated question-relevant object. If yes, we add that input object's attention score to the VG score for that question (i.e., a question's VG score lies between 0 and 1).

Both metrics aim to measure how much "attention" a model puts on visual input objects that were determined to represent question-relevant objects given in GQA's grounding annotations.

We use a third attention-based VG metric to compare VLR with MAC-Caps (Urooj et al., 2021), which attempts to improve VG in MAC (Hudson and Manning, 2018) using visual capsule modules (Sabour et al., 2017). For the comparison with MAC-Caps, we use the F1-score based metric that was introduced alongside the model (code available at Urooj (2021)). This metric

[3]Original metric description in Hudson and Manning (2019). Official code can be downloaded at `https://cs.stanford.edu/people/dorarad/gqa/evaluate.html`

[4]IoU (Intersection over Union) is a standard evaluation metric used in the field of Object Detection. IoU is also known as the Jaccard index.

measures Precision, Recall and F1-score of IoU@0.5-based matches between ground-truth objects and certain objects determined in the visual input by a form of attention thresholding.

VG measurements are based on attention maps which are interpreted as feature importance weight distributions over visual input objects. These maps are extracted for each evaluated model as follows:

- *UpDn* natively produces a single attention map which we use as-is.
- For *MAC* we select the map produced in the final reasoning step before the answer is generated.
- *DFOL* produces a relevance distribution over objects in each inference step. Here, we use the map produced for the final state (or an average thereof for multiple final states).
- *MMN* uses a transformer-based architecture, employing multiple layers with multi-head attention. We take the average of all attention maps involved in the inference process after the encoding layer, i.e., an average over 7 self-attention layers with 8 attention heads each for one inference step. We use the resulting averaged map representing the final inference step as MMN's attention distribution.
- *VLR* does not employ a native attention mechanism, so we uniformly distribute 100% attention weight among the final object(s) in the 1-best Viterbi path(s), as the answer is fully dependent on them.

We evaluate multiple metrics for the following reasons: First, the Grounding metric (Equation 3.6) is ill-defined and can create scores of $> 100\%$ due to their algorithm allowing attention weights of each input region to be factored in more than once. We still adopt this metric to be able to compare with previously published results, e.g., Li et al. (2019a); Hudson and Manning (2019). Secondly, additional metrics can serve as a way to solidify the results.

Categories for Relevant Objects

GQA annotates question-relevant objects in three categories: Objects referenced in the question (Q), in the short (often one word) answer (A), and the full sentence answer (FA). We evaluate VG for each of these categories, as well as a combined category containing all annotated question-relevant objects (Q+A+FA). In each case, the grounding scores are averaged over all involved questions.

Line	System	QP	SG	Q	A	FA	Q+A+FA	Grounding	Acc
1	Det.Obj.	n/a	VLR	91.45	92.34	91.66	91.49	n/a	n/a
2	UpDn	n/a	VLR	23.72	39.32	28.78	29.90	94.42	60.51
3	MAC	n/a	VLR	25.10	38.56	29.32	30.37	94.90	62.66
4	MMN	MMN	VLR	30.34	29.94	29.14	30.34	98.22	**65.91**
5	DFOL	GQA	VLR	39.04	39.57	42.41	43.56	114.37	55.78
6	VLR	VLR	VLR	**49.05**	**48.47**	**52.99**	**54.28**	**128.41**	57.67
7	VLR-2	GQA	VLR	57.47	31.90	55.11	61.23	132.00	60.16
8	VLR-7	VLR	GQA	62.97	46.16	63.29	70.04	149.82	79.88
9	VLR-Oracle	GQA	GQA	70.93	50.94	70.75	78.46	162.73	91.78

Table 3.2 – VG results, discussed in Chapter 3.4.3. Higher is better in all columns. VLR exhibits strong VG in all categories and metrics (Line 6). Best result in bold (only considering non-Oracle systems, i.e., Lines 2-6). Note, that Line 1 does not show VG measurements but lists average percentages of question-relevant objects in the detected input scene graph (i.e., on average about 8% of relevant objects are not detected).

Results Discussion

Detailed results for the evaluated systems are listed in Table 3.2. IoU-based VG scores for all relevance categories (Q, A, FA and Q+F+FA) show that VLR manifests much stronger VG quality compared to other models in all categories. Most scores improve by $> 20\%$ relative compared to the best performing reference system DFOL. Importantly, VLR is significantly more committed to relying on relevant *final* answer region(s) to produce its answer (Table 3.2, column "A").

Comparisons of VLR and MAC-Caps with MAC-Caps' VG metric are listed in Table 3.3 and suggest a major difference in VG quality. It should be noted that MAC-Caps was designed and evaluated based on grid-based visual features (as opposed to object-based features). Both MAC and MAC-Caps results in this table are based on grid-based features and were taken from Urooj et al. (2021) (and their Appendix). VLR uses object-based features, which evidently presents a substantial advantage over such models in object-centric VG evaluation, as illustrated by the large evaluation differences.

Results for various Oracle-based[5] variants of VLR (Table 3.2, Lines 7-9) show that its VG quality (measured over the entire inference path) improves further alongside accuracy, in particular with better scene graphs, while

[5]Oracle: Systems evaluated with ground-truth annotations to varying degrees instead of using fully model-predicted inputs. E.g., using GQA's scene graph annotations as input instead of the model-generated scene graph.

improvements in QP have less of an impact. For instance, replacing QP-generated programs (Line 8) with Oracle (i.e., ground-truth) programs (Line 9) shows only improvements of around 10% relative in all VG measurements. This is consistent with trends we observed for accuracy measurements (see also VLR's ablation study in Appendix B.3.1) and reaffirms the strong bond between perception modules and VG in VLR.

3.4.4 Generalization & Out-of-Distribution

Compositionality and a retrieval-based design enable VLR to side-step some of the most prominent challenges that current classification-based VQA approaches struggle with, in particular certain content generalization and OOD scenarios. We investigate these scenarios in the following.

Generalization Experiments

We perform experiments with re-partitioned GQA train/test sets to investigate how VQA systems handle four generalization settings that simulate challenges typically encountered in a practical real-world setting. Note that the underlying perception module (our SGG) remains unchanged for all systems in these experiments. We give an overview of the settings here and defer to Appendix B.4 for details on the implementation of the applied re-distribution method. The data splits are accessible on GitHub[6].

- **Generalization to new object names.** We test a model's ability to handle previously unseen object names in questions. Akin to a similar setting used in Hudson and Manning (2019), we remove all QA-samples from training that contain any object name from the food or animal category in the question. The test set then contains only these types of questions.

[6]https://github.com/dreichCSL/GQA_generalization_splits

System	Prec	Rec	F1	Ground	Acc
MAC	1.97	2.28	2.11	41.68	57.09
MAC-Caps (Urooj et al., 2021)	2.53	3.10	2.79	45.54	55.13
VLR	**53.76**	**30.41**	**38.85**	**128.41**	**57.67**

Table 3.3 – VG comparison of MAC, MAC-Caps and VLR, using the VG metric from Urooj et al. (2021), calculated for the final step of inference in the "Q+A+FA" category. MAC and MAC-Caps results are taken from Urooj et al. (2021), VLR is evaluated by us. Higher is better in all columns.

- **Generalization to linguistic variants.** Similar to Hudson and Manning (2019), we test model generalization to variants of questions that are equivalent in terms of inference but are linguistically differently formulated (e.g. "Do you see a car?" vs. "Are there any cars in the image?"). Equivalent questions are re-partitioned such that we only train on questions of one linguistic variant and test on questions of the other variant.

- **Generalization to new answer options.** Unlike classifier-based approaches, VLR does not learn a pre-fixed answer vocabulary. This means that it can — in theory — produce an infinite amount of unique answers without retraining the system itself, while classifier-based systems are restricted to a pre-fixed set of answers. We illustrate this by removing all QA-samples with answers that are food or animal names from training and then test on QA-samples with answers from only those categories.

- **Low-resource training.** To test a model's transferability, we simulate the massive input space of VQA under real-world conditions by causing a shortage of exhaustive (Question,Image,Answer) training tuples and disaligning train/test priors (similar to certain OOD conditions, but in a less controlled manner). Concretely, we limit training samples per answer option to 1000 (test set remains unchanged).

Results discussion. Results of the four generalization experiments are shown in Table 3.4. All models used in this section were trained with the same data. Note that DFOL, which can only be evaluated with ground-truth programs (due to the missing release of its question parser), cannot be reasonably evaluated in these scenarios, as performance of the question parser is a major factor here.

System	Objects	Ling. Variants	Answers	Low-Resource
Train/Test	763k/23k	801k/20k	862k/11k	311k/132k
UpDn	35.75 (65.66; -45.6%)	58.20 (64.33; -9.5%)	0.0 (57.62; -100%)	49.72 (60.51; -17.8%)
MAC	34.99 (64.91; -46.1%)	**58.37** (64.05; -8.9%)	0.0 (57.96; -100%)	49.11 (62.66; -21.6%)
MMN	41.71 (69.31; -39.8%)	57.27 (65.31; -12.3%)	0.0 (61.36; -100%)	51.24 (65.91; -22.3%)
VLR	**56.40** (60.99; **-7.5%**)	52.19 (56.85; **-8.2%**)	**39.88** (51.63; **-22.8%**)	**51.58** (57.67; **-10.6%**)

Table 3.4 – Generalization experiments, discussed in Chapter 3.4.4. Accuracy numbers in parenthesis represent results (and relative difference) when training in a regular setting (i.e., with the unmodified GQA balanced train set). Higher is better for accuracy, lower is better for relative percentage differences.

Table 3.4 shows that VLR achieves the highest accuracy in three of the four experimental settings. Notably, VLR surpasses all evaluated models by large margins when generalizing to new objects and answers, which is a direct consequence of VLR's retrieval-based design and the QP's pointer-generator architecture that supports an open vocabulary (unlike, e.g., MMN's parser). For linguistic variants, most systems exhibit similar relative accuracy degradation (compared to a model trained on all available data). In the Low-Resource setting, VLR suffers a much smaller relative drop than other systems. VLR performs well in this scenario, because it does not need to learn patterns for (Question,Image,Answer) triples; it only needs to learn to accurately parse the question in isolation, which is a much less data-intensive learning task and not directly dependent on (Question,Image,Answer) coverage[7].

Out-of-Distribution Testing

We now investigate performance of VLR in a dedicated Out-of-Distribution (OOD) setting, which was introduced in Ying et al. (2022) as GQA-101k. This data set was created to measure OOD performance in GQA akin to what the VQA-CP split (Agrawal et al., 2018) does for the VQA dataset (Antol et al., 2015). GQA-101k follows VQA-CP's "changing priors" methodology for re-distributing questions in GQA's balanced split (see also Chapter 2.5.1).

Evaluations in Table 3.5 show VLR's exceptional performance on GQA-101k, with VLR's In-Distribution (ID) and OOD accuracy surpassing other evaluated models. Moreover, VLR's achievements in both ID and OOD testing are virtually on par, while all other models exhibit a substantial gap between their ID/OOD results. VLR's performance shows that it can effectively nullify the ID/OOD accuracy gap, thereby successfully meeting expectations for a VQA-system focused on implementing human-intended decision rules for inference.

3.5 Limitations

In this section, we discuss design-related limitations of VLR regarding its application and extension to new use-cases.

[7]It is worth mentioning that the QP's independence from images and answers of a VQA-focused dataset means that it can in principle leverage VQA-unrelated text data sources and benefit from relevant new developments in the Natural Language Processing (NLP) field, such as the recent breakthrough of Large Language Models (LLM).

System	ID	OOD
UpDn (Anderson et al., 2018)	55.21	34.57
MAC (Hudson and Manning, 2018)	54.90	33.08
MMN (Chen et al., 2021)	54.84	37.64
VLR	**55.55**	**56.33**

Table 3.5 – Accuracies in Out-of-Distribution testing on the GQA-101k data split. Higher is better.

(1) *Dependency on annotations:* VLR learns from program annotations that map a question onto a functional notation defined for question types in GQA. Although the performed generalization experiments show VLR to have comparably lower requirements in terms of training data size, it still usually requires additional program annotations when retraining for new scenarios.

(2) *Generalization to new question types:* This limitation is related to (1). VLR can handle compositional questions used in GQA. Although an important subset of question types in the VQA-field fall into this category, VLR will struggle when faced with questions that significantly deviate from GQA in terms of program structure, thus requiring retraining (with additional annotations). Note that this point pertains to the *program structure* of questions, which is different from VLR's ability to generalize well to new *content* and — to a lesser degree — *linguistic variants* in already learned question types.

(3) *Handling sophisticated inference:* As a retrieval-based system, VLR handles questions that can be answered directly based on retrieved paths involving objects/regions and their properties in an image. More involved inference, such as questions that require external world-knowledge and common-sense reasoning to be correctly answered, cannot be handled efficiently by VLR.

(4) *Generalization to new answer types:* Unlike classifier-based VQA systems, VLR can generalize to output answers it has never seen in training — but only if the answer *type* is supported (e.g., returning the name of an object; confirming an object's existence). VLR uses rule-based logic in its answer production. If different types of answers need to be produced (such as counting objects), the answer production has to be manually expanded to add support. Such extension may not necessarily require much effort to implement. E.g., support for "counting"-type

questions could be relatively quickly realized by combining already integrated mechanisms. Concretely, by running an n-best Viterbi search and adding a new rule that counts paths that have a unique final path object and exceed the probability threshold for object existence (note: the same threshold mechanism is used for "verify"-type questions).

3.6 Summary and Conclusion

In this chapter, we have introduced "VQA by Lattice-based Retrieval" (VLR), a modular, transparent system based on concepts from the field of Automatic Speech Recognition and Information Retrieval, that largely embodies an implementation of human-based decision rules for VQA, thereby acting as an approximation of a model that is less prone to shortcut learning.

In breaking with the predominant VQA learning paradigm of classification and designing a system from the standpoint of an Information Retrieval approach, we have shown that VLR manifests significantly better Visual Grounding in the inference process than various reference systems. A number of newly constructed evaluation settings have shown VLR's strong generalization capabilities in terms of handling new object names, linguistic variants, low-resource training and most notably its distinctive ability to produce answers never seen in training. VLR's exceptional performance in a dedicated OOD scenario further adds to its distinguishing strengths and confirms the intended behavior in an anti-shortcut setting.

Despite the discussed limitations of the approach and its shortcomings in ID accuracy compared to state-of-the-art models, VLR — as a retrieval-based system — offers some unique advantages that set it apart from other VQA systems. We can see VLR as a viable alternative to current classifier-based VQA systems in practical use-cases that require a) strong Visual Grounding (e.g., in voice user interfaces and robotics: in order to handle follow-up queries about visual objects; interaction with queried objects anywhere in the inference path; more explainable/predictable interaction), b) robust generalization performance in various scenarios (e.g., in embedded applications such as children toys), and c) an open answer vocabulary, which can be useful for querying large and changing varieties of image contents (e.g., in multimedia search applications). In terms of theoretical considerations, we envision VLR and this study of its properties w.r.t. some of VQA's most prominent issues (VG, generalization, OOD scenarios) to support analytical investigations and general progress in these areas.

Chapter 4

Measuring Faithful and Plausible Visual Grounding in VQA

Metrics for Visual Grounding (VG) in Visual Question Answering (VQA) systems primarily aim to measure a system's reliance on relevant parts of the image when inferring an answer to the given question. Lack of VG has been a common problem among state-of-the-art VQA systems and can manifest in over-reliance on irrelevant image parts or a disregard for the visual modality entirely (Goyal et al., 2017; Agrawal et al., 2018). Although inference capabilities of VQA models are often illustrated by a few qualitative illustrations, most systems are not quantitatively assessed for their VG properties. An easily calculated criterion for meaningfully measuring a system's VG can help remedy this shortcoming, as well as add another valuable dimension to model evaluations and analysis. To this end, we propose a new VG metric that captures if a model a) identifies question-relevant objects in the scene, and b) actually relies on the information contained in the relevant objects when producing its answer, i.e., if its Visual Grounding is both "faithful" and "plausible". The proposed metric is called Faithful & Plausible Visual Grounding (FPVG) and is straightforward to determine for most VQA model designs.

In this chapter, we introduce FPVG and establish its properties through a series of experiments. Code to support the metric calculations on the GQA data set is shared on GitHub[1].

[1] https://github.com/dreichCSL/FPVG

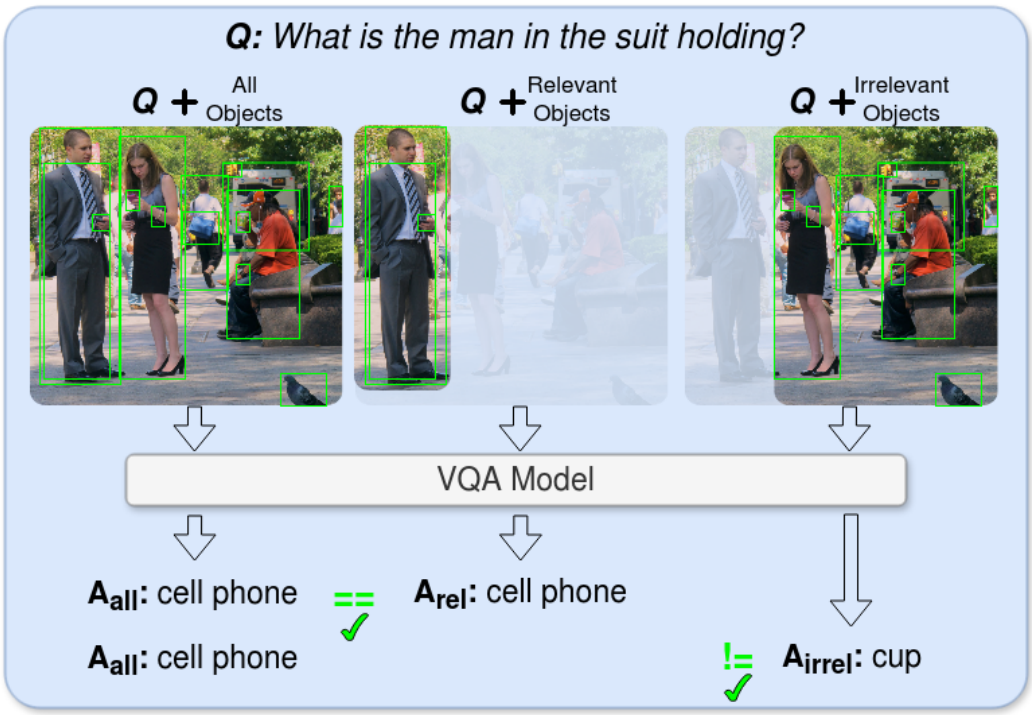

Figure 4.1 – Faithful & Plausible Visual Grounding (FPVG): The VQA model's answer given *all* objects in the image (A_{all}) should equal its answer when given only *relevant* objects w.r.t. the question (A_{rel}), and should differ when given only *irrelevant* objects (A_{irrel}). In the pictured example, the model returns the same answer ("cell phone") when the visual input consists of all or only relevant objects, and returns a different answer ("cup") when given only irrelevant objects. Hence, the question is deemed faithfully and plausibly grounded under FPVG's definition.

4.1 Introduction

Metrics that quantify a VQA model's VG characteristics aim to capture its internal reasoning process based on methods of model explanation. These explanations generally vary in properties of *plausibility* and *faithfulness*. *Plausible* explanations of a model's behavior prioritize human interpretability by making use of illustrations that are intuitively understood by humans, such as providing a clear inference path over relevant objects in an image that lead to the answer decision. However, plausible explanations might not accurately reflect a model's actual decision-making process. *Faithful* explanations, on the other hand, prioritize a more accurate reflection of a model's decision-making process, possibly at the expense of human interpretability. Examples of plausible explanation methods are attention mechanisms (Bahdanau et al.,

2015) over visual input objects and certain other model outputs that may be the result of multi-task objectives that teach the model to produce inference paths without conclusive involvement in the main model's answer decision (Chen et al., 2021). Faithful explanation methods may employ testing schemes with modulated visual inputs followed by comparisons of the model's output behavior across test runs (DeYoung et al., 2020; Gupta et al., 2022). While the latter types of metrics are particularly suited for the use-case of object-based visual input in VQA, they often a) require large compute budgets to evaluate the required number of input permutations (e.g., SwapMix (Gupta et al., 2022), Leave-One-Out (Li et al., 2016)); b) might evaluate in unnecessary depth, like in the case of Softmax-score-based evaluations (DeYoung et al., 2020); and/or c) evaluate individual properties separately and without considering classification contexts, thereby missing the full picture (DeYoung et al. (2020); Ying et al. (2022), see also Chapter 4.3.4).

In this chapter, we propose a VG metric that is both *faithful* and *plausible* in its explanations. Faithful & Plausible Visual Grounding (FPVG) quantifies a model's faithful reliance on plausibly relevant image regions. An example illustrating this concept is shown in Figure 4.1. FPVG is based on a model's answering behavior for modulated sets of image input regions, similar to other faithfulness metrics (in particular DeYoung et al. (2020)), while avoiding their above-mentioned shortcomings, which we discuss in detail in Chapter 4.3.4. A metric to meaningfully quantify VG in VQA models is an essential pre-requisite for accurate VG analysis. The development of FPVG is therefore crucial to our investigations of VG in this thesis.

In the remainder of this chapter, we first provide a broader context for VG metrics in VQA by reviewing existing methods, after which we introduce FPVG and establish its properties and advantages over other VG metrics.

4.2 Measuring Visual Grounding in VQA

4.2.1 Visual Grounding Metrics

Various metrics have been proposed to measure VG in VQA models. We roughly group these into "direct" and "indirect" methods.

Direct Methods

The most widely used methods measuring the importance of image regions to a given question are based on a model's attention mechanism (Bahdanau et al., 2015), or use gradient-based sensitivities (in particular variants of GradCAM

(Selvaraju et al., 2017)). VG is then estimated, e.g., by accumulating importance scores over matching and relevant annotated image regions (Hudson and Manning, 2019), or by some form of rank correlation (Shrestha et al., 2020). Aside from being inapplicable to non-attention-based VQA models (e.g., symbolic methods like Yi et al. (2018); Mao et al.), attention scores have the disadvantage of becoming harder to interpret the more attention layers are employed for various tasks in a model. This gets more problematic in complex Transformer-based models that have a multitude of attention layers over the input image (OSCAR (Li et al., 2020; Zhang et al., 2021), LXMERT (Tan and Bansal, 2019), MCAN (Yu et al., 2019b), MMN (Chen et al., 2021)). Additionally, attention mechanisms have been a topic of debate regarding the faithfulness of their explanation (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019). Gradient-based sensitivity scores can theoretically produce faithful explanations, but require a careful choice of technique and implementation for each model individually to achieve meaningful measurements in practice (Adebayo et al., 2018; Feng et al., 2018). Various works define their own VG metric based on attention measurements (e.g., GQA-Grounding (Hudson and Manning, 2019), VLR[2] (Chapter 3), MAC-Caps (Urooj et al., 2021)) or GradCAM-based feature sensitivities (Shrestha et al., 2020; Wu and Mooney, 2019; Selvaraju et al., 2019; Han et al., 2021), implicitly assuming faithful measurements without thorough analysis or validation of the metric's actual properties.

Indirect Methods

This category includes methods that measure VG based on a model's predictions under particular test (and train) conditions, e.g., with perturbations of image features (Yuan et al., 2021; Gupta et al., 2022; Agarwal et al., 2020; DeYoung et al., 2020; Alvarez-Melis and Jaakkola, 2017), or specially designed Out-of-Distribution (OOD) test sets that are thought to uncover a model's insufficient VG properties (Agrawal et al., 2018; Kervadec et al., 2021; Ying et al., 2022).

4.2.2 Right for Right Reasons

VG can be considered a sub-problem of the VQA desiderata gathered under the term "Right for Right Reasons" (RRR) (Ross et al., 2017; Ying et al., 2022). RRR may additionally include investigations of causal behavior in a model that goes beyond (and may not be strictly dependent on) VG and may

[2]Note that VLR was conceived before FPVG and therefore used other metrics to evaluate VG.

involve probing the model for its robustness and consistency in explanations, e.g., via additional (follow-up) questions (Patro et al., 2020; Selvaraju et al., 2020; Ray et al., 2019; Park et al., 2018).

4.3 Faithful & Plausible Visual Grounding

In this section, we develop a formal definition of FPVG, followed by a motivation of its design and the intuition behind it. We then empirically validate FPVG's property of faithfulness[3] by comparison of its VG quality categorizations with that of various other faithful VG metrics. Finally, we illustrate FPVG's advantages over related metrics.

4.3.1 Metric Formulation

We propose a new metric to determine the degree of Faithful & Plausible Visual Grounding (FPVG) in a VQA model M_{VQA} w.r.t. a given VQA data set S. Here, S consists of tuples s_j of question, image and answer $(q, i, a)_j$. Each such tuple in S is accompanied by annotations indicating relevant regions in image i that are needed to answer the question q. M_{VQA} is characterized by its two modality inputs (i and q) and a discrete answer output (a). Without loss of generality, here, we expect image i to be given as an object-based representation (e.g., bag of objects, scene graph) in line with the de-facto standard for VQA models[4].

FPVG requires evaluation of M_{VQA} under three test conditions. Each condition differs in the set of objects representing image i in each sample s_j of the test. Three tests are run: 1) with all available objects (i_{all}), 2) with only relevant objects (i_{rel}), and 3) with only irrelevant objects (i_{irrel}). Formally, we define one dataset variant for each of these three conditions:

$$s_{j_{all}} = (q, i_{all}, a)_j, \quad s_{j_{all}} \in S_{all} \tag{4.1}$$

$$s_{j_{rel}} = (q, i_{rel}, a)_j, \quad s_{j_{rel}} \in S_{rel} \tag{4.2}$$

$$s_{j_{irrel}} = (q, i_{irrel}, a)_j, \quad s_{j_{irrel}} \in S_{irrel} \tag{4.3}$$

[3]Only faithfulness needs explicit validation. The property of plausibility does not require further validation as it is achieved by FPVG's definition to evaluate a model's reliance on an annotated set of plausibly relevant image regions.

[4]In principle, FPVG can be easily adapted to work with any model (VQA or otherwise) that follows a similar input/output scheme as the standard region-based VQA models, i.e., an input consisting of N entities where a subset can be identified as "relevant" ("irrelevant") for producing a discrete output.

The relevance of an object in i is determined by its degree of overlap with any of the objects referenced in relevance annotations for each individual question (for details on the concrete process, see Appendix C.1). FPVG is then calculated on a data point basis (i.e., for each question) as

$$FPVG_j = Eq(\hat{a}_{j_{all}}, \hat{a}_{j_{rel}}) \wedge \neg Eq(\hat{a}_{j_{all}}, \hat{a}_{j_{irrel}}), \tag{4.4}$$

where $\hat{a}_j$ is the model's predicted answer for sample s_j and $Eq(x, y)$ is a function that returns True for equal answers. FPVG takes a binary value for each data point. A positive FPVG value for sample $s_{j_{all}}$ is only achieved if M_{VQA}'s output answers are equal between test runs with samples $s_{j_{all}}$ and $s_{j_{rel}}$, and unequal for samples $s_{j_{all}}$ and $s_{j_{irrel}}$ (remember that the three involved samples only differ in their visual input). The percentage of "good" (i.e., faithful & plausible) and "bad" FPVG is then given as $FPVG_+$ and $FPVG_-$, respectively:

$$FPVG_+ = \frac{1}{n} \sum_{j}^{n} FPVG_j \tag{4.5}$$

$$FPVG_- = 1 - FPVG_+, \tag{4.6}$$

where n is the total number of samples (i.e., questions).

We further sub-categorize FPVG to quantify correctly ($\top$) and incorrectly ($\bot$) predicted answers $\hat{a}_{j_{all}}$ as $FPVG^{\top}_{\{+,-\}}$ and $FPVG^{\bot}_{\{+,-\}}$, respectively. Hence, samples are assigned one of four categories, following their evaluation behavior. The resulting categories are formally defined as follows:

$$FPVG^{\top}_{+} = \frac{1}{n} \sum_{j}^{n} (FPVG_j * Eq(\hat{a}_{j_{all}}, a_j)) \tag{4.7}$$

$$FPVG^{\bot}_{+} = \frac{1}{n} \sum_{j}^{n} (FPVG_j * (1 - Eq(\hat{a}_{j_{all}}, a_j))) \tag{4.8}$$

$$FPVG^{\top}_{-} = \frac{1}{n} \sum_{j}^{n} ((1 - FPVG_j) * Eq(\hat{a}_{j_{all}}, a_j)) \tag{4.9}$$

$$FPVG^{\bot}_{-} = \frac{1}{n} \sum_{j}^{n} ((1 - FPVG_j) * (1 - Eq(\hat{a}_{j_{all}}, a_j))) \tag{4.10}$$

Equations 4.7–4.10 sum to 1. Figure 4.2 shows an illustration for each of the four categories.

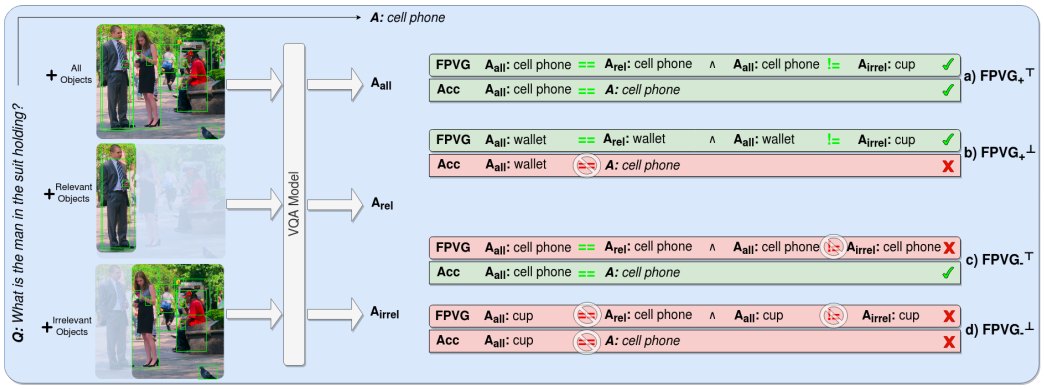

Figure 4.2 – Examples for the four FPVG sub-categories defined in Chapter 4.3.1. Each sub-category encapsulates specific answering behavior for a given question in FPVG's three test cases (A_{all}, A_{rel}, A_{irrel}). Categorization depends on grounding status ("FPVG") and answer correctness ("Acc"). E.g., questions that return a correct answer in A_{all} and A_{rel} and an incorrect answer in A_{irrel} are categorized as (a). The model's behavior in cases (a) and (b) satisfies the criteria for the question to be categorized as faithfully & plausibly visually grounded.

4.3.2 Intuition behind FPVG

The intuition behind tests based on object selections in S_{rel} (relevant objects) and S_{irrel} (irrelevant objects) is as follows:

Testing on relevant objects S_{rel}. In the context of FPVG, the output of a well-grounded system is expected to remain steady for S_{rel}, i.e., the model is expected to retain its original prediction from S_{all}, if it relies primarily on relevant visual evidence. Hence, a change in output indicates that the model has changed its focus to different visual evidence, presumably away from irrelevant features (which are dropped in S_{rel}) onto relevant features — a sign of "bad" grounding.

Testing on irrelevant objects S_{irrel}. In the context of FPVG, the output of a well-grounded system is expected to waver for S_{irrel}, i.e., the model is expected to change its original prediction in S_{all}, as this prediction is primarily based on relevant visual evidence which is unavailable in S_{irrel}.

Summarizing expectations for well-grounded VQA. A VQA model that relies on question-relevant objects to produce an answer (i.e., a well-grounded model that values visual evidence) should:

1. Retain its answer as long as the given visual information contains all relevant objects.

2. Change its answer when the visual information is deprived of all relevant objects and consists of irrelevant objects only.

During (1), answer flips should not happen, if the model relied only on relevant objects within the full representation S_{all}. However, due to tendencies in VQA models to ignore visual evidence, lack of flipping in (1) could also indicate an over-reliance on the language modality (implies indifference to the visual modality). To help rule out those cases, (2) can act as a fail-safe that confirms that a model is not indifferent to visual input.

The underlying mechanism can be described as an indirect measurement of the model's feature valuation of relevant objects in the regular test run S_{all}. The two additional experimental setups with S_{rel} and S_{irrel} help approximate the measurement of relevant feature valuation for S_{all}.

FPVG and accuracy. FPVG classifies samples $s_{j_{all}} \in S_{all}$ as "good" (faithful & plausible) or "bad" grounding by considering whether or not the changed visual input impacts the model's final decision, *independently of answer correctness*. Many VQA questions have multiple valid (non-annotated) answer options (e.g., "man" vs. "boy" vs. "person"), or might be answered incorrectly on account of imperfect visual features. Thus, it is reasonable to expect that questions can be well-grounded, but still produce an incorrect answer, as shown in Figure 4.2, (b). Hence, FPVG categorizes samples into two main grounding categories ($FPVG_+$ and $FPVG_-$). To enable a more fine-grained analysis, answer correctness is considered in two additional sub-categories ($FPVG^{\top}$, $FPVG^{\perp}$) within each grounding category, as defined in Equations 4.7–4.10.

4.3.3 Validating FPVG's Faithfulness

FPVG achieves *plausibility* by its definition to evaluate a model's reliance on an annotated set of plausibly relevant objects and therefore does not require further validation. In this section, we validate that FPVG's sample categorization is also driven by *faithfulness* by verifying that questions categorized as $FPVG_+$ are more faithfully grounded than questions in $FPVG_-$. To measure the degree of faithful grounding for each question, we first determine an importance ranking among the question's input objects. Then we estimate how well this ranking matches with the given relevance annotations.
Three types of approaches are used in VQA to measure object importance by direct or indirect means:

1. Measuring attention (direct): Attention over input objects gives a sense of importance the model assigns to each object (used, e.g., in Li et al. (2019a); Urooj et al. (2021); Hudson and Manning (2019)).

2. Measuring gradients (direct): Gradient-based methods like GradCAM are close to the model's inner workings as they involve estimating a direct link between the importance of the input features and a model's output decision (used, e.g., in Selvaraju et al. (2019); Wu and Mooney (2019); Shrestha et al. (2020)).

3. Feature manipulation (indirect): Manipulations are typically realized by omission of input entities (i.e., vectors representing objects). The manipulated image representation can be zero-padded to maintain the model's size expectations, as is commonly done for variable length inputs in sequence modeling. Other variants used in VQA include replacing omitted objects with certain other values (e.g., constants (Ying et al., 2022), object features from other images (Yuan et al., 2021; Gupta et al., 2022)).

We consider one representative method from each of these three categories: VQA-model UpDn's attention and the feature manipulation method Leave-One-Out (LOO[5]) (Li et al., 2016) were found to deliver the most faithful measurements of feature importance in similar experiments with UpDn on GQA in Ying et al. (2022). We use these two methods and also include GradCAM for completeness.

We measure UpDn's behavior on GQA's balanced validation set. Table 4.1 lists the ranking match degree between object importance rankings (based on S_{all}) and relevance annotations, averaged over questions categorized as $FPVG_+$ and $FPVG_-$, respectively. The "relevant" ("irrelevant") category produces a high score if all relevant (irrelevant) objects are top-ranked by the used method (see Appendix C.2 for details). Hence, faithfully grounded questions are expected to score highly in the "relevant" category, as relevant objects would be more influential to the model's decision.

Results in Table 4.1 show that object importance rankings over the same set of questions and model vary greatly across methods. Nonetheless, we find that data points in both $FPVG_+$ and $FPVG_-$ achieve on average favorable scores across all three metrics with mostly considerable gaps between opposing

[5]LOO evaluates a model N times (N=number of input objects), each time "leaving-out-one object" of the input and observing the original answer's score changes. A large score drop signifies high importance of the omitted object.

	relevant		irrelevant	
Method	$FPVG_+ \uparrow$	$FPVG_- \downarrow$	$FPVG_+ \downarrow$	$FPVG_- \uparrow$
Attention	60.9	26.6	16.7	51.2
GradCAM	10.4	8.5	53.7	67.4
LOO	29.8	16.0	52.0	71.7

Table 4.1 – Ranking match percentage between feature importance rankings and relevant/irrelevant objects for questions in $FPVG_+$ and $FPVG_-$. Model: UpDn.

categories (i.e., $FPVG_+$ and $FPVG_-$). This is in line with expectations and confirms that FPVG's data point categorization is driven by faithfulness.

4.3.4 Comparison with Existing Metrics

Two metrics to measure faithfulness in a model, "sufficiency" and "comprehensiveness", were proposed in DeYoung et al. (2020) and used in the context of VQA in similar form in Ying et al. (2022).
"Sufficiency" and "comprehensiveness" are similar to FPVG and therefore deserve a more detailed comparison. They are calculated as follows.

Definition

Let a model M_θ's answer output layer be represented as softmax-normalized logits. A probability distribution over all possible answers is then given as $p(a|q, i_{all}) = m_\theta(q, i_{all})$. The max element in this distribution is M_θ's predicted answer, i.e., $\hat{a} = \underset{a}{argmax}\ p(a|q, i_{all})$, where the probability for the predicted answer is given by $p_{\hat{a}_{all}} = M_\theta(q, i_{all})_{\hat{a}}$.

Sufficiency is defined as the change of output probability of the predicted class given *all* objects vs. the probability of that same class given only *relevant* objects:

$$suff = p_{\hat{a}_{all}} - p_{\hat{a}_{rel}} \tag{4.11}$$

Comprehensiveness is defined as the change of output probability of the predicted class given *all* objects vs. the probability of that same class given only *irrelevant* objects:

$$comp = p_{\hat{a}_{all}} - p_{\hat{a}_{irrel}} \tag{4.12}$$

A faithfully grounded model is expected to achieve low values in $suff$ and high values in *comp*.

Object relevance and plausibility

The definition of what constitutes relevant or irrelevant objects is crucial to the underlying meaning of these two metrics. FPVG uses annotation-driven object relevance discovery and subsequently determines a model's faithfulness w.r.t. these objects. Meanwhile, Ying et al. (2022) estimates both metrics using *model-based* object relevance rankings (e.g., using LOO), hence, measuring the degree of faithfulness a model has towards model-based valuation of objects as determined by an object importance metric. A separate step is then needed to examine these explanations for "plausibility". In contrast, FPVG already incorporates this step in its formulation, which determines if the model's inference is similar to that of a human by measuring the degree of *faithful* reliance on *plausibly* relevant objects (as defined in annotations).

Advantages of FPVG

FPVG overcomes the following shortcomings of $suff$ and *comp*:

1. $Suff$ and *comp* are calculated as an average over the data set independently of each other and therefore do not evaluate the model for presence of *both* properties in each data point.
2. $Suff$ and *comp* only consider prediction probabilities of the maximum answer class in isolation, which means that even a change in model output as significant as a flip to another class may be declared insignificant by these metrics (for instance, this can happen for $suff$ if the output distribution's max probability $p_{\hat{a}_{all}}$ is similar to $p_{\hat{a}_{rel}}$).

Shortcoming 1. Figure 4.3, left, illustrates why isolating the two properties can cause inaccurate readings (1). The analyzed model assigns "good" $suff$ scores (defined in Ying et al. (2022) as $< 1\%$ absolute probability reduction from $p_{\hat{a}_{all}}$ to $p_{\hat{a}_{rel}}$) to a large number of questions (left two quadrants in Figure 4.3, left). However, many of these questions also show "bad" *comp* ($< 20\%$ absolute drop from $p_{\hat{a}_{all}}$ to $p_{\hat{a}_{irrel}}$) (lower left quadrant in Figure 4.3, left), which reflects model behavior that one might observe when visual input is ignored entirely. Thus, the full picture is only revealed when considering both properties in conjunction, which FPVG does. Further evidence of the drawback stemming from (1) is pictured in Figure 4.3, right, which shows average LOO-based ranking match percentages (cf. Chapter 4.3.3) for data

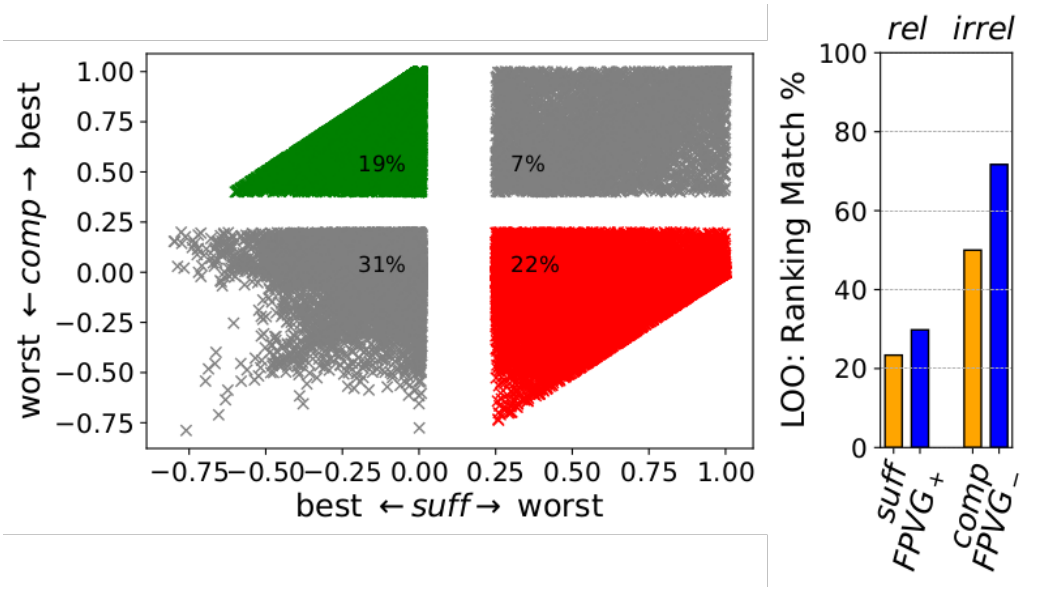

Figure 4.3 – Left: Percentage of samples with best (worst) $suff$ & $comp$ scores (medium scores not pictured). Many samples with the $suff$ property lack $comp$ and vice-versa (gray). Right: LOO-based ranking match percentages for samples in $suff$, $comp$ and FPVG (higher is better). Model: UpDn.

points categorized as "best" $suff$ or $comp$ and FPVG. Data points in FPVG's categories score more favorably than those in $suff$ and $comp$, illustrating a more accurate categorization.

Shortcoming 2. Figure 4.4, left, illustrates problem (2). A large percentage of questions with best (=low) scores in $suff$ flip their answer class (i.e., $suff$'s "best" category fails to reach 0% flipped percentage), even when experiencing only minimal class probability drops ($< 1\%$ absolute). Similarly, some percentage of questions with best (=high) $comp$ scores fail to flip their answer (i.e., $comp$'s "best" category fails to reach 100% flipped percentage), even though the class probabiliy dropped significantly ($>= 40\%$ absolute drop). Both described cases show that failure to consider class probabilities in the context of the full answer class distribution negatively impacts the metric's quantification of a model's VG capabilities w.r.t. actual effects on its answer output behavior. FPVG's categorization avoids this issue by being defined over actual answer changes (Figure 4.4, right: flipped prediction percentages per VG category are always at the expected extremes, i.e., 0% or 100%).

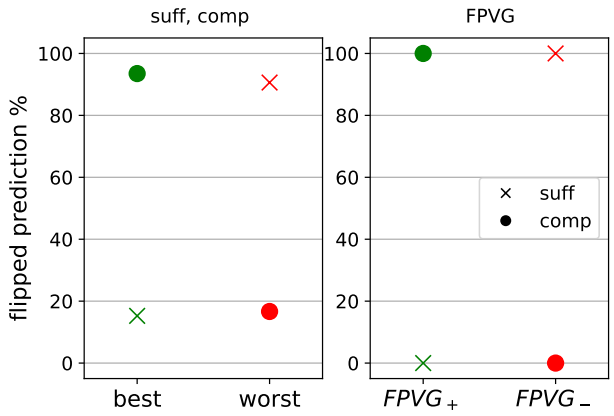

Figure 4.4 – Sample distribution and answer class flip percentages depending on metric categorization. X-axis: VG quality categories based on $suff$ & *comp* (left) and FPVG (right). Y-axis: percentage of flipped answers in each category. Note that in this figure, FPVG's formulation is interpreted in terms of $suff$ (Equation 4.4, right side, left term) and *comp* (right term). Model: UpDn.

Summary. FPVG avoids shortcomings (1) by taking both $suff$ and *comp* into account in its joint formulation at the data point level, and (2) by looking at actual answer output changes (Figure 4.4, right) and thus implicitly considering class probs over all classes and employing meaningful decision boundaries for categorization. Additionally, relying on answer flips instead of an abstract softmax score makes FPVG more intuitively interpretable.

4.3.5 Discussion on other existing metrics

FPVG relies on the method of feature deletions to determine "faithful" reliance on a "plausible" set of inputs. Other VG metrics exist that instead rely on GradCAM (Shrestha et al., 2020) or a model's attention mechanism (Hudson and Manning, 2019) to provide a "faithful" measurement of input feature importance. The two mentioned metrics leverage these measurements to determine if a model relies on "plausibly" relevant objects. For instance, Shrestha et al. (2020) calculates a ranking correlation between the measured GradCAM scores and the rankings based on (plausible) object relevance annotations. The metric in Hudson and Manning (2019) sums all of a model's attention values assigned to visual input objects that have been determined to represent plausible objects.

While "plausibility" is straightforwardly achieved by appropriate selection of plausibly relevant reference objects (which would be the same across these metrics), the property of "faithfulness" is more difficult to obtain and heavily dependent on the employed feature importance technique. Investigations in Ying et al. (2022) cast doubt on the faithfulness of GradCAM measurements, with feature deletion techniques and attention mechanism scoring most favorably in faithfulness in the explored setting. However, as discussed in Chapter 4.2, the faithfulness of attention measurements has not been without scrutiny, and is not straightforward to extract correctly in models that make heavy use of attention mechanisms (such as Transformers). Based on this evidence, we find the method of feature deletions to be the most sensible and versatile choice to achieve faithfulness of measurements in FPVG across a wide range of model architectures in VQA.

4.4 Limitations

Plausibility of explanations in FPVG is assumed to be provided by accurate, unambiguous and complete annotations of relevant objects per evaluated question. Although the GQA data set provides annotations in the shape of relevant object pointers during the inference process for a question, these annotations may be ambiguous or incomplete. For instance, a question about the color of a soccer player's jersey might list pointers to a single player in an image where multiple players are present. Excluding only this one player from the image input based on the annotated pointer would still include other players (with the same jersey) for the S_{irrel} test case. In such cases, FPVG's assumptions would be violated and its result rendered inaccurate. In this context, we also note that FPVG's behavior has not been explicitly explored for cases with ambiguous relevance annotations.

We note that the expectation for having such complete annotations may not be achievable in practice. Each sample's relevance annotation would require verification that no combination of non-relevant object pointers can still lead to the correct answer for the given question (any potentially problematic questions could then be removed from the evaluation). This verification could be done automatically, but only if 1) object-level image annotations provide tags for *all* pictured objects, or alternatively 2) the involved object detector model is accurate enough to recognize *all* pictured objects correctly. Both of these possibilities are, at this point in time, unrealistic expectations in itself, in particular when dealing with rich real-world images, which is the case in GQA.

Secondly, FPVG creates its visual input modulations by matching annotated objects with objects detected by an object detector. Different object detectors can produce bounding boxes of varying accuracy and quantity depending on their settings. When using a new object detector as a source for visual features, it might be necessary to re-adjust parameters used for identifying relevant/irrelevant objects (see Appendix C.1 for settings used in this work). When doing so, the integrity of FPVG can only be retained when making sure that there are no overlapping objects among relevant & irrelevant sets.

Thirdly, comparing VQA models with FPVG across visual features produced by different object detectors might be problematic/inaccurate in itself, as 1) different numbers of objects are selected for relevant & irrelevant sets, and 2) different Q/A samples might be evaluated (e.g., due to missing detections of any relevant objects). If possible, when using a new/different object detector, we recommend including FPVG evaluations for some reference model(s) (e.g., UpDn) as an additional baseline to enable an improved assessment of a model's FPVG measurements that are trained with a different object detector's visual features.

4.5 Summary

In this chapter, we introduced Faithful & Plausible Visual Grounding (FPVG), a metric that facilitates and streamlines the analysis of VG in VQA systems. While its property of plausibility is inherently given by the metric's definition over plausibly relevant objects, we empirically established its faithfulness with a series of experiments and comparisons with other faithful VG metrics.

FPVG embodies a streamlined VG metric with meaningful properties and a crucial tool for accurate and efficient VG analysis of VQA models. As such, FPVG is an essential milestone for our investigations into VG's role in VQA in general and OOD scenarios in particular.

Chapter 5

Information Infusion with Symbolic Features

The input to a VQA model's visual modality is typically represented by high-dimensional sub-symbolic feature vectors that were extracted from a separately trained object detector (OD) like Faster R-CNN Ren et al. (2015). This kind of object-based image representation realizes a reasonable level of symbolization of the raw image space, which facilitates semantic analysis of a model's internal workings, including VG. Nevertheless, the deeper sub-symbolic nature of the extracted feature vectors still obstructs views into a crucial aspect of any input modality: information content. While spatial location information that each vector's features represent in the raw image is usually provided, interpretation of information content represented by the features is hidden by their sub-symbolic nature. As a result, influence of the visual modality on the VQA process cannot be cleanly isolated. Not only does this impede analysis, but it can also cultivate misleading interpretations based on problematic assumptions regarding the input's informational content.

In order to enable easy interpretation of information content carried by the visual modality, we propose to replace sub-symbolic features with symbolic ones that reflect the carried information by using a higher form of symbolism. The ability to control and investigate the information content that the visual modality provides to a VQA model opens up a number of options for deep analysis related to VG that are otherwise closed off when using standard sub-symbolic features. This includes the use of image annotations to repre-

sent input images to isolate the Object Detector's (OD) performance as an influential error source.

In this chapter, we describe the construction of symbolic features that are required for in-depth analysis into VG in Part III. Symbolic features allow a closer inspection of the visual modality's informational payload, thereby enabling us to investigate VG based on *verifiably represented* content instead of content that is only *assumed* to be present in features, based on represented image location alone. Furthermore, symbolic features can be easily manipulated by a procedure we call "Information Infusion", which we leverage in experiments in Part III.

5.1 Symbolic features

In this section, we describe the structure of our symbolic features, the three instantiations we experiment with in this thesis, and lastly the process of "Information Infusion", which can be described as surgical manipulation of image content.

5.1.1 Structure

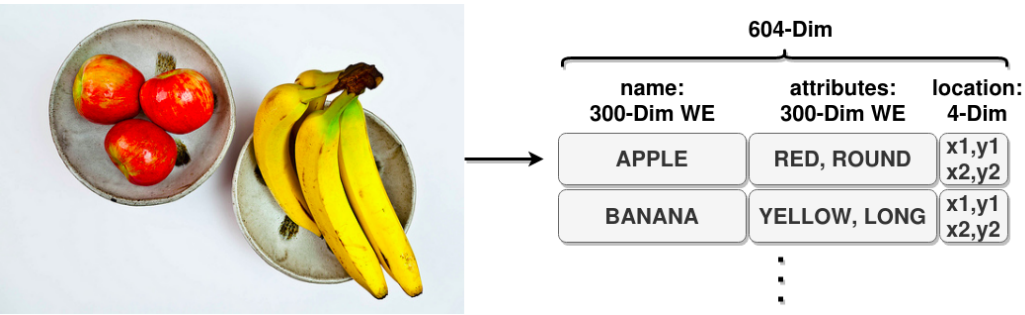

Figure 5.1 – Symbolic features. For a description see text.

In order to establish a firm handle on the informational content of the visual input, we engineer object-based symbolic visual features. Each constructed symbolic feature vector represents an object in the scene (as detected by an OD) and carries information about its name (e.g., apple, banana), attributes (e.g., red, round) and location in the image. This information is encoded by two stacked 300-dimensional GloVe word embeddings (Pennington et al., 2014) and a 4-dimensional vector containing the bounding box location coordinates. Figure 5.1 illustrates an example of the described structure.

5.1.2 Feature extraction

Symbolic features in this thesis are derived from classification outputs of the scene graph generator (SGG) described in detail in Appendix A. We include a few essential details about this SGG for easier understanding of the following sections and defer to the appendix for a full description.

SGG details

Our SGG uses a Faster R-CNN model for 1) object detection of up to 100 objects belonging to 1702 object classes, and 2) bounding box coordinate regression per object. Attribute recognition for each detected object is done for each of the 39 attribute categories present in GQA separately. Each attribute category (e.g., color, shape, material) consists of two or more attribute classes (e.g., in the color category: red, blue, etc.), with an overall total of 617 classes.

5.1.3 Feature Vector Construction

In this work, we experiment with three symbolic visual representations that vary in informational content (structure remains identical):

- **Detection (DET) features** are created using outputs of SGG. We use the maximum class from SGG's object detection as object name (a 300D GloVe embedding) and the (normalized 4D) bounding box coordinates as location information. For determining attribute information (the second 300D GloVe embedding), each of the 39 attribute categories provides its maximum class. To represent the attribute information of all categories in a single word embedding, we take the average of all recognized attribute name embeddings.
- **Oracle (ORA) features** are created accordingly, but without any involvement of SGG. These features are based on GQA's scene graph annotations which contain each object's name, attributes and location.
- **Infusion (INF) features** are created by targeted manipulation of DET features (details below).

5.2 Information Infusion

In addition to interpretability advantages of symbolic features over sub-symbolic representations, they provide one more crucial advantage over standard sub-symbolic features: easy information manipulation. Since we know

where and how information is stored in the vector, altering visual content and thereby controlling the model's *knowledge base* is straightforward. Specifically, we intend to introduce new information and correct wrong detections in given visual representation in our experiments. We call this process "Information Infusion". The resulting features are named **Infusion (INF) features**. The following two manipulation types are used in this work:

1. *Introduction of new objects:* Introducing a new object vector constructed from scratch based on image annotations. VQA models typically impose a strict limit for the number of input objects. New objects are either simply appended to the list of exisiting objects (if said object limit was not exhausted) or replace the object with the least confident object class recognition. Further requirements are that any replaced object cannot be *relevant* to the current question (this affects VG experiments in particular).

2. *Alteration of existing objects:* Modifications of object name information is performed by simple replacement of the respective 300D GloVe embedding. For modifications of attribute information, we first determine the attribute category of the new attribute to be infused. Then, the embedding contributions of the detected attribute of that category is replaced with an equivalently weighted embedding of the new attribute.

5.3 Summary

Symbolic features open up additional options for VQA analysis. In particular the described procedure of Information Infusion is helpful in our VG-centric research endeavors, as visual features play a key role for VG manifestation in a VQA model. Specifically, we apply Information Infusion in Chapter 7 and Chapter 8, where it is instrumental in unveiling the true impact of VG-methods on VQA models as well as for the creation of a VG-centric OOD test.

Part III

Investigations & Insights

Introduction

In Part II, we introduced methods and processes needed for our diagnostic journey into VG in VQA. We now leverage these methods and processes to deepen our understanding of VG. Part III is organized as follows:

In Chapter 6, we start out by establishing an overview of VG capabilities across various VQA architectures using our metric FPVG. We find that modern VQA models, although high performing in ID accuracy, are still lacking in VG quality. We finish the chapter with investigations of the connection between OOD and VG, articulating initial insights into the impact of VG on OOD accuracy.

In Chapter 7, we find that current evaluation practices for VG-boosting methods are problematic due to problematic assumptions w.r.t. presence of relevant information in a VQA model's visual input. We empirically show that VG-methods are much more potent when underlying assumptions are better aligned with actual data conditions encountered in train and test scenarios.

In Chapter 8, we formally show that current OOD tests are unsuitable as proxies for estimating VG quality and its impact on models by VQA accuracy alone. We demonstrate how existing OOD tests offer plenty of opportunities for VG-related shortcut exploitation despite acting as proxies for measuring shortcut-free behavior in VQA models in related works. Based on these findings, we develop a test scenario that properly reflects the significance of VG in the context of shortcut learning in VQA.

The following publications share results and content with this part:

- Measuring Faithful and Plausible Visual Grounding in VQA (Reich et al., 2023)
- Uncovering the Full Potential of Visual Grounding Methods in VQA (Reich and Schultz, 2024)

Chapter 6

Visual Grounding Evaluations

Improving answer accuracy in OOD scenarios has been a primary motivator for VG research in VQA (e.g., Agrawal et al. (2018); Selvaraju et al. (2019); Wu and Mooney (2019); Ying et al. (2022)). Behind such motivation lies an assumption that strengthening VG quality in VQA models will have bias mitigating effects and prevent shortcut learning. In other words, improvements in a model's VG quality are expected to naturally translate to better performances on OOD tests, which are designed to penalize models that rely on shortcut exploitation and reward those that have learned to apply the *intended decision rules* (see also Chapter 2.4). Nevertheless, the identification of a clear correlation between OOD accuracy and metrics of VG quality — which would facilitate a proper definition of the nature of their connection — is still eluding the field, where evidence of incongruency between the two metric types have been reported (Shrestha et al., 2020; Ying et al., 2022).
In this chapter, we perform experiments with VLR and FPVG to establish an understanding about the state-of-the-art of VG in current VQA model architectures. We then conduct experiments to investigate how OOD performance is influenced by VG and report clear tendencies that show impact of VG quality on OOD performance.

6.1 VG Quality in Current VQA Models

In this chapter, we apply FPVG in evaluations of influential model representatives of various VQA architectures to build an overview of the state-of-the-art of VG quality across VQA model designs.

6.1.1 Experiment Setup

Dataset. The GQA dataset (Hudson and Manning (2019), described in Chapter 2.3.2) provides detailed grounding information for a majority of its questions, which benefits a comprehensive analysis of VG. In the following experiments, we use GQA's "balanced" split (943k samples) for training, but include the full train split (14m samples) where required in official training instructions for certain models. Standard answer accuracy evaluation is performed on the full balanced val set (132k samples) for all models. Reported numbers for all FPVG-related results are based on a subset thereof, and size depends on the used visual features (SG-generated features: 114k samples; VinVL features: 110k samples). For more details on how these subsets were determined, see Appendix C.1.

VQA Models. For our VG overview, we choose VQA models from a wide variety of model designs from recent years. Detailed descriptions for most models are provided in Chapter 2.2. Concrete training details can be found in Appendix C.3. The following models are evaluated:
UpDn (Anderson et al., 2018) is an attention-based model that popularized the contemporary standard of object-based image representation. **MAC** (Hudson and Manning, 2018) is a multi-hop attention model for multi-step inference, well-suited for visual reasoning scenarios like GQA. **MCAN** (Yu et al., 2019b), MMN (Chen et al., 2021) and OSCAR+ (Zhang et al., 2021) are all Transformer-based (Vaswani et al., 2017) models. **MMN** employs a modular design that disentangles inference over the image information from the question-based prediction of inference steps as a functional program in a separate process, thereby improving interpretability compared to monolithic systems like MCAN. MMN also makes an effort to learn correct grounding using an auxiliary loss. **OSCAR+** uses large-scale pre-training on multiple Vision+Language (V+L) datasets and is subsequently fine-tuned on GQA's balanced train set. We use the official release of the pre-trained OSCAR+ base model (which is based on proprietary visual features) as starting point for fine-tuning. **DFOL** (Amizadeh et al., 2020c) is a neuro-symbolic method that disentangles vision from language processing via a separate question parser similar to MMN and our VLR model. We introduced **VLR** in Chapter

Model	Acc ↑	Acc_{all}↑	Acc_{rel}↑	Acc_{irrel}↓	$FPVG^{\top}_{+}$↑	$FPVG^{\perp}_{+}$	$FPVG^{\top}_{-}$↓	$FPVG^{\perp}_{-}$	$FPVG_{+}$↑
MAC	60.23	59.20	58.12	44.33	15.40	7.19	43.81	33.60	22.59
UpDn	55.53	57.99	58.51	44.32	15.76	9.68	42.23	32.33	25.44
UpDn+HINT	55.56	57.95	57.88	42.98	16.31	9.72	41.64	32.33	26.03
MCAN	66.18	65.78	67.3	44.62	20.18	6.20	45.60	28.02	26.37
OSCAR+	**70.52**	**69.96**	**71.79**	50.24	20.37	6.00	49.58	24.05	26.37
MMN	68.49	68.23	64.37	43.93	21.93	5.86	46.29	25.92	28.22
DFOL	55.79	57.45	57.36	36.70	20.19	10.03	37.25	32.53	30.22
UpDn+VisFIS	57.09	60.01	63.71	43.25	20.38	12.20	39.63	27.79	32.58
VLR	57.25	57.39	61.29	**35.99**	**24.55**	11.68	**32.83**	30.93	**36.23**
UpDn*	65.22	64.81	68.28	43.00	23.90	9.29	40.92	25.89	33.19

Table 6.1 – FPVG results for various models, sorted by $FPVG_{+}$. All models were trained by us. Accuracy (Acc) is calculated on the GQA balanced val set (132k samples), while all other columns are calculated on an FPVG-dependent subset thereof (white rows: 114k samples; grey rows: 110k samples). Blue arrows indicate desirable behavior for well-grounded VQA in each metric category[1](best results in bold). Gray colored rows use VinVL visual features, others use our own. Bottom line: Results for UpDn* trained with VinVL features are included to allow an easier assessment of OSCAR+ (w/ VinVL) results.

3 as a modular, symbolic method that prioritizes strong VG over accuracy by following a retrieval-based design paradigm instead of the commonly employed classification-based design in VQA.

VG-methods. In addition to these main models, we include two VG-methods that focus on grounding improvements and are both applied to UpDn model training: **HINT** (Selvaraju et al., 2019) aligns GradCAM-based (Selvaraju et al., 2017) feature sensitivities with annotated object relevance scores. **VisFIS** (Ying et al., 2022) adds an ensemble of various RRR/VG-related objective functions (including some data augmentation) to the training process.

Visual Features. Apart from OSCAR+, all models are trained using the same object-based 1024-dim visual features generated by VLR's Scene Graph Generator (see Appendix 9.2 for details). For OSCAR+, we fine-tune the officially released pre-trained base model, which was trained with improved visual features designed specifically for V+L tasks called VinVL (Zhang et al., 2021).

	$Acc_{all}\uparrow$		$FPVG_{+}\uparrow$	
Model	ID	OOD	ID	OOD
UpDn	51.40±0.58	30.83±1.96	17.50±0.87	19.33±0.73
HINT	51.28±0.39	31.34±0.55	18.06±1.23	19.59±0.68
VisFIS	53.28±0.44	33.42±1.03	25.10±0.78	25.18±0.94
MAC	52.10±0.46	31.31±0.50	15.40±0.51	16.72±0.22
MMN	52.28±0.43	36.48±0.56	18.74±0.32	17.88±0.60
VLR	55.64	56.38	37.56	38.51

Table 6.2 – Accuracy (i.e., Acc_{all}) and $FPVG_{+}$ for models evaluated with GQA-101k over five differently seeded training runs.

6.1.2 Results Discussion

Results are listed in Table 6.1, with models sorted by $FPVG_{+}$ (last column). Our first observation is that FPVG and accuracy are not indicative of one another, confirming that our metric for grounding is complementary to accuracy and adds a second angle to VQA model analysis that looks beyond answer correctness. Secondly, we see that (neuro-)symbolic methods like DFOL, and VLR in particular, stand out among (non-VG-boosted) VQA models in terms of FPVG, even while trailing in accuracy considerably. Thirdly, we find that methods that boost grounding characteristics, like VisFIS, show promise for closing the gap to symbolic methods — if not exceeding them. Lastly, we observe that $FPVG_{+}$ is generally low in all evaluated models, indicating that there is still ample room for VG improvements in VQA.

6.2 VG Quality and OOD Performance

We use FPVG to gain insights into the challenge of OOD settings by analyzing VQA models with GQA-101k (Ying et al., 2022), a dataset proposed for OOD testing. GQA-101k consists of a repartitioned train/test set based on balanced GQA and was created following a similar methodology as the OOD split called VQA-CP (Agrawal et al., 2018).

Results in Table 6.2 show median values and maximum deviation thereof over five differently seeded training runs per model type (note that VLR uses

[1]FPVG sub-categories $FPVG_{+}^{\perp}$ and $FPVG_{-}^{\top}$ have no intuitively sensible ranking directive under the FPVG motivation and therefore lack a blue arrow.

	c2i ratio for $FPVG_+$		c2i ratio for $FPVG_-$	
Model	ID	OOD	ID	OOD
UpDn	1.35±.09	0.77±.05	1.11±.03	0.43±.04
HINT	1.36±.07	0.85±.05	1.11±.02	0.43±.01
VisFIS	1.40±.06	0.84±.06	1.23±.02	0.47±.02
MAC	1.44±.06	0.77±.05	1.16±.02	0.45±.01
MMN	1.91±.05	1.21±.12	1.11±.02	0.57±.02
VLR	1.91	2.12	1.09	1.05

Table 6.3 – Correct to incorrect (c2i) answer ratios for questions categorized as $FPVG_{\{+,-\}}$. Note that this table is not intended to serve as a ranking of VQA model performance, as c2i ratios are not suitable for that purpose. We use the table to analyze and discuss the differences in model behavior w.r.t. VG (see text). Data set: GQA-101k.

deterministic inference, so no additional runs were performed for it). Table 6.3 lists correct-to-incorrect (c2i) answer ratios for six model types trained and evaluated on GQA-101k. The c2i ratios are determined for each test set (ID/OOD) and $FPVG_{\{+,-\}}$. They are calculated as number of correct answers divided by number of incorrect answers, hence, a c2i ratio of > 1 reflects that correct answers dominate the considered subset of test questions. In the following analysis, we leverage the listed c2i ratios to investigate and illustrate the connection between VG and (OOD) accuracy.

6.2.1 Understanding the connection between FPVG and accuracy

In Table 6.1 and 6.2 we observe a somewhat unpredictable relationship between $FPVG_+$ and accuracy. We analyze the c2i ratios in Table 6.3 to gain a better understanding of this behavior. Table 6.3 shows that FPVG-curated c2i ratios can vary substantially across model types (e.g., UpDn vs. MMN). These ratios can be interpreted as indicators of how effectively a model can handle and benefit from correct grounding. Large differences between models' c2i profiles explain why the impact of VG on accuracy can vary significantly across models. E.g., MMN has a much stronger c2i profile than UpDn, which explains its higher OOD accuracy even with lower $FPVG_+$.

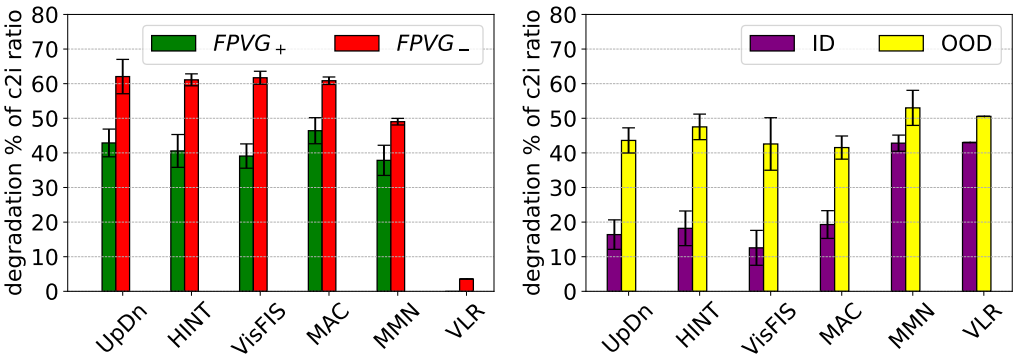

Figure 6.1 – Performance drops when comparing ID to OOD (questions in $FPVG_{\{+,-\}}$, left), and when comparing $FPVG_+$ to $FPVG_-$ (questions in ID/OOD, right). Data set: GQA-101k.

6.2.2 Understanding the connection between FPVG and OOD performance

The inter-dependency of VG and OOD performance plays an important role in VQA generalization. FPVG can help us gain a deeper understanding.

More OOD errors when VG is bad. Figure 6.1, left, depicts relative c2i ratio degradation when comparing ID to OOD settings. All models suffer a much higher c2i drop for questions categorized as $FPVG_-$ than $FPVG_+$. In other words, models make more mistakes in an OOD setting in general, but they tend to do so *in particular when questions are not correctly grounded*. Note, that VLR is affected to a much lower degree due to its quasi-insensitivity to Q/A priors.

VG is more important to OOD than ID. Figure 6.1, right, shows accuracy sensitivity towards changes in grounding quality, i.e., when comparing $FPVG_+$ to $FPVG_-$. We draw two conclusions: 1) All models suffer from c2i degradation, hence, they all tend to make more mistakes for questions categorized as $FPVG_-$ than $FPVG_+$. 2) This tendency is (considerably) more pronounced in OOD which provides evidence that *OOD performance is particularly sensitive to grounding*.

Summary. Our analysis shows that *VQA models have a clear tendency to make mistakes in OOD for questions that are not faithfully grounded*. This tendency is consistently observed across various model types and model instances. Our findings support the idea that weak visual grounding is

detrimental to accuracy in OOD scenarios in particular, where the model is unable to fall back on learned Q/A priors to find the correct answer (as it can do in ID testing). Furthermore, we note that VisFIS, which boasts considerable improvements in FPVG and strong improvements in accuracy over basic UpDn, is unable to overcome these problematic tendencies. This suggests that VG-boosting methods *alone* might not be enough to overcome a model's fixation on language-based priors, which is exacerbating the performance gap between ID/OOD.

6.3 Conclusion

In this chapter, we have shown that FPVG can be a valuable tool in analyzing VQA system behavior, as demonstrated in particular by the presented investigations of the VG-OOD relationship. Here, we found that VG plays an important role in OOD scenarios, where, compared to ID scenarios, bad VG leads to considerably more errors than good VG, thus providing us with a compelling argument for pursuing better-grounded models.

Furthermore, we investigated VQA systems of various architectural designs and found that many models struggle to reach the level of faithful & plausible VG that systems based on symbolic, programmed inference like VLR provides. Of notable interest is also VLR's strong OOD performance which reaches an equilibrium with its ID accuracy, something that other methods are still far away from achieving. This discrepancy in performance between learned and programmed inference can be interpreted as indication of two issues: 1) learned decision rules, as employed by all evaluated classification-based VQA models, are overshadowed by shortcut learning, and 2) while clear tendencies of VG influence on OOD performance were discovered, the extent of VG's contributions towards bias mitigation could not be universally quantified for some currently unknown reasons. We further investigate these matters in greater detail in Chapter 8, where we develop a clear understanding of the source of this mystery and show why commonly used OOD tests are ill-suited for analyzing VG's impact on shortcut behavior in VQA models.

Chapter 7

Uncovering the Full Potential of Visual Grounding Methods

Visual Grounding (VG) methods in VQA attempt to improve VQA performance by strengthening a model's reliance on question-relevant visual information. The presence of such relevant information in the visual input is typically assumed in training and testing. This assumption, however, is inherently problematic when dealing with imperfect image representations common in large-scale VQA, where the information carried by visual features frequently deviates from expected ground-truth contents. As a result, training and testing of VG-methods is performed with largely inaccurate data, which obstructs proper assessment of their potential benefits.

In this study, we demonstrate that current evaluation schemes for VG-methods are impaired due to the problematic assumption of availability of relevant visual information. Our experiments show that these methods can be much more effective when evaluation conditions are corrected. Our findings suggest that the potential value of VG in VQA models has been misrepresented by problematic evaluation methodologies.

Code and data to reproduce experiments and results reported in this chapter has been released on GitHub[1].

[1] https://github.com/dreichCSL/TrueVG

7.1 Introduction

Visual Grounding (VG) in VQA has garnered interest not only as a key aspect to furthering understanding and rationalization of a VQA model's inference procedure, but also as a way to improve Out-of-Distribution (OOD) performance by preventing certain dataset biases to form. Various works have reported evidence of problematic tendencies in VQA models that point to a disregard of relevant image regions during answer inference and the manifestation of Q/A distribution biases in the model (Agrawal et al., 2018; Goyal et al., 2017; Agrawal et al., 2016). We have shown in Chapter 6 that a lack of VG quality in VQA models negatively impacts OOD performance. Similarly, weak VG has been tied to a general unpredictability of answering behavior (Gupta et al., 2022). To alleviate these issues, methods have been developed that seek to strengthen a model's reliance on question-relevant visual features. These *VG-methods*, of which we describe two in Chapter 2.5.3, either modify the training procedure of existing models (e.g., HINT (Selvaraju et al., 2019), SCR (Wu and Mooney, 2019), VisFIS (Ying et al., 2022)), or are integrated directly into specialized model architectures such as MMN (Chen et al. (2021), described in Chapter 2.2.7), PVR (Li et al., 2019a) and VLR (Chapter 3).

On a technical level, the goal of VG-methods in VQA is to align a model's internal valuation of visual input feature importance (FI) with human-based FI, which is given as guidance in training. These human-based FI scores can be inferred from a question's visual relevance annotations, which may be given as highlighted regions in the *raw image* (e.g., spatial heat maps as in VQA-HAT (Das et al., 2016)), or explicit pointers to ground-truth objects (as in GQA (Hudson and Manning, 2019)). Notably, relevance annotations are not given in *input feature space* directly, and therefore a mapping function is required to identify corresponding visual features and determine FI scores. The predominant approaches for such a mapping between image and feature space rely exclusively on *spatial matching*: Visual input features receive their FI scores depending on spatial overlap between the region they represent and question-relevant annotated locations in the raw image (cf. Selvaraju et al. (2019); Wu and Mooney (2019); Shrestha et al. (2020); Ying et al. (2022)). High-scoring features can then be identified as relevant cue objects[2]. In this approach, the actual *visual content* carried by the cue objects is simply assumed to be appropriate without further *semantic verification* and

[2]Visual feature vectors for VQA are commonly generated by object detectors such as Faster R-CNN (Ren et al., 2015) and are therefore assumed to represent objects in an image.

Figure 7.1 – Example of Impaired VG in training: Left: VG-methods in VQA teach the model to rely on specified visual input features without verifying presence of relevant visual information (here: the correct identity of the depicted animal), which leads to incongruity in training. Right: Example of True VG. Ideally, the model should be taught to rely on question-relevant visual features with accurate content.
Example of Impaired VG in testing: Consider the five solid colored squares (left) as a model's visual input and a test question as follows: 1) "Is the truck's back door open?": VG on the truck is required, but the input does not contain a truck (missing object detection is signified by the red dashed square). 2) "What is the cow doing?": VG on the cow is required, but the input does not contain a cow (wrong object recognition is signified by the red colored square). In both cases, impact of proper VG on accuracy cannot be cleanly evaluated.

therefore does not influence their score. In this work, we report evidence that such incomplete verification can result in grossly mismatched cues, thereby leading to inadequate guidance in VG-method training, as illustrated in Figure 7.1, left. Similarly, tests performed under such unchecked conditions fail to accurately evaluate the originally intended use case that VG-methods were designed for, as question-relevant content is often missing in the input and

proper VG impossible. While work such as Shrestha et al. (2020) and Ying et al. (2022) investigate the underlying effects of VG-method application in detail, we are unaware of any study that also considers the impact of these problematic conditions in their analysis. This is insofar an issue as research on VG-methods involves perhaps the most influential and prominent works among any reports that focus on the VG property in VQA models and their impact on OOD generalization (in particular Selvaraju et al. (2019); Wu and Mooney (2019)). As such, reported impact of VG-methods on OOD testing might have a profound effect on the general perception of how important the VG property actually is for OOD generalization. The findings in this chapter suggest that the overall value of VG for OOD scenarios has been severely misrepresented by the use of problematic evaluation methodologies in VG-method research that fail to uncover the full potential of said methods. This is further exacerbated by a general lack of reports being accompanied by thorough VG measurements which would allow a more appropriate assessment of VG efficacy (the lack of thorough VG measurements in related investigations was similarly criticized by Shrestha et al. (2020)). We hope our work can help raise awareness of these issues by its thorough investigation of the mentioned training and testing impairments, and encourage a re-evaluation of the benefits of VG for OOD generalization.

7.1.1 Contributions

In this chapter, we seek to develop a better understanding of the benefits of VG-methods in VQA when training and testing conditions properly support their intended use-case. We identify two impairments and their causes in current evaluation practices for VG-methods and outline our "True VG" methodology to fix them. Our approach establishes a new framework for evaluating VG-methods more thoroughly. Finally, a comprehensive analysis examines the impact of "True VG" settings, providing new insights into the unfulfilled potential of VG-methods.

We summarize the contributions of this chapter as follows:

- We show that the impact of VG-methods on VQA accuracy is misrepresented by commonly used but problematic training and testing procedures that involve severe VG mismatches.
- We propose a new methodology for training and testing VG-methods under corrected conditions with rectified VG misalignments, which results in improved VG and VQA performance (code is provided).
- We provide an in-depth study of VG-method impact in ID/OOD scenarios that is comprehensive in terms of datasets, methods and assessment.

7.2 Background

VG-Methods in VQA. VG-boosting methods used for bias mitigation operate under the assumption that strengthening a model's reliance on relevant visual input will in turn weaken the influence of dataset-inherent biases towards Q/A priors and thereby improve OOD performance in VQA. Hence, evaluations on ID/OOD splits like VQA-CP (Agrawal et al., 2018) and the methodically similarly constructed splits introduced in Ying et al. (2022) are often used to evaluate VG-method effectiveness for VQA. VG-boosting training-schemes may involve data augmentation with modulations of (relevant) visual input in image space (Gokhale et al., 2020) or feature space (Gupta et al., 2022), or training the model with objective functions that encourage an inference alignment with relevant image features for answer production, such as HINT (Selvaraju et al., 2019) and SCR (Wu and Mooney, 2019). VisFIS (Ying et al., 2022) combines both types of approaches in an ensemble of multiple objective functions.

Relevance annotations and Feature Matching. VG-methods typically leverage annotations to point out question-relevant parts in the input image. VQA-HAT Das et al. (2016) gathers such annotations in the form of spatial heat maps. These heat maps are recorded by tracking a user's computer mouse during de-blurring of image regions needed to answer crowd-sourced questions from the VQA dataset (Antol et al., 2015).
Template-based questions found in GQA (Hudson and Manning, 2019) and CLEVR-XAI (Arras et al., 2022) are generated in conjunction with an underlying visual scene graph and provide semantic relevance annotations of involved objects (or image regions) as a natural byproduct.
Computational approaches attempt to determine relevance annotations by employing a mapping between image region annotations and question words (Gokhale et al., 2020), or by leveraging human-sourced textual explanations for the answer to a given question (VQA-X (Park et al., 2018), used in Wu and Mooney (2019)).

In all cases, the image-based relevance annotations are subsequently used to identify relevant cue objects in a model's visual input feature space. The mapping from annotations to input features has traditionally been based entirely on the features' receptive field, i.e., the image *location* that the features represent (Selvaraju et al., 2019; Wu and Mooney, 2019; Shrestha et al., 2020; Ying et al., 2022). The work in this chapter goes one step further and examines VG-methods that consider cue objects that additionally match the *content* of the relevant ground-truth object. To the best of our knowledge,

the effects of such *semantic matching* on VG-method efficacy in VQA has not been explicitly investigated before.

7.3 Impaired Visual Grounding

We posit that evaluations of VG-boosting methods in VQA follow a problematic methodology, which consequently causes an incomplete and potentially misleading understanding of the benefits of VG and VG-boosting methods in VQA.
Specifically, we investigate the following issues:

(I1) **Impaired testing:** Current evaluations of VG-methods hide their full potential by diluting tests with questions that are impossible to correctly ground due to missing relevant visual information. Examples for impaired testing are given in the caption of Figure 7.1 (bottom paragraph).

(I2) **Impaired training:** Impact of VG-boosting methods is muted due to training with a large percentage of unsuitable training samples that are missing relevant visual information necessary for teaching consistently correct inference alignments. This problem is illustrated in Figure 7.1.

We further identify two underlying *causes* for these issues:

(C1) **Noisy features:** Impacts I1 and I2. Object misrecognitions and missing detections of relevant objects in the input image representation occur frequently in large-scale object detection tasks. We find that only 30% of the used training questions (and 27%/26% of ID/OOD tests) in our GQA experiments contain all necessary question-relevant information.[3]

(C2) **Fuzzy spatial matching of cue objects:** Impacts I2. Spatial identification of relevant cue objects may declare irrelevant objects as relevant on account of their close vicinity to the reference location, even if the represented visual content is inadequate and therefore irrelevant (as illustrated in Figure 7.1). We identify a question-average of 2.6 cue objects in GQA training using semantic matching, which is inflated to 5.4 cue objects using spatial matching (counted based on a threshold of $IoU > 0.5$; matching methods defined in Chapter 7.4.2). This means that on average more than half of the objects that were declared question-relevant by spatial matching are in fact irrelevant.

[3]These numbers are based on success rates of *semantic matching* (see 7.4.2 for a description), i.e., only 30% of training questions are accompanied by visual input that contains matches for all question-relevant ground-truth objects.

7.4 Experiment Setup

We empirically show that addressing impairments I1 & I2 outlined in Chapter 7.3 provides new insights into the efficacy of VG-methods in VQA. In this section, we describe our methodology for fixing these impairments for analytical purposes.

7.4.1 Approach

Enhancing the testing process (I1). Developing a complete understanding of the potential of VG-methods requires their evaluation on target cases where proper VG is feasible in principle. A basic requirement for this is that question-relevant information needs to be fully represented in the visual input. Therefore, we determine "True Visual Grounding" (TVG) test subsets, which are verified to only contain questions that are accompanied by *complete* relevant visual features (i.e., features that match all question-relevant reference annotations in both location & content).

Enhancing the training process (I2). VG-methods operate under the assumption that given training targets, i.e., visual features and their FI scores, are viable. Hence, training samples are expected to provide 1) relevant visual features carrying the content that is needed to answer the given question, and 2) FI scores that highlight them correctly in the set of all input features. Object detection-based visual features are noisy (see C1 in Chapter 7.3) and (parts of) the set of cue objects highlighted by spatial matching might be irrelevant and/or incomplete (see C2 in Chapter 7.3), thus resulting in failure to meet this requirement in many cases. We ensure availability of relevant visual content in the input by "infusing" missing information. These infused features can then be paired with perfect FI scores as guidance for VG-methods in training.

7.4.2 From relevance annotations to cue objects

Relevance annotations point out the location (and in GQA also the identity) of ground-truth objects in the raw image that are relevant for answering a given question. In object-based VQA, the raw image and the actual image representation that is fed to the VQA model are not in the same space. Hence, a mapping is needed to identify cue objects in the input representation that match the relevance annotations. We use two types of mapping methods in this chapter:

(M1) **Spatial matching.** Cue objects are identified (scored) by measuring bounding box overlap (IoU) of (detected) visual input objects with relevant ground-truth objects.
(M2) **Semantic matching.** Scoring additionally involves verification that the *content* of the spatially matched cue object matches the ground-truth object's identity (i.e., its name and attributes).

Typically, spatial matching is used in scoring relevant cue objects when relevance annotations point out relevant bounding boxes (such as in GQA). One of the reasons M2 has been neglected so far is the difficulty of interpreting sub-symbolic features w.r.t. their visual content. To circumvent this roadblock in this chapter, we leverage symbolic features that enable controlled and interpretable encoding of feature content (see Chapter 5).

Determining FI scores. Spatial and semantic matching are used to determine FI scores for each visual input object. These FI scores are subsequently used as guidance by VG-methods.
In *spatial matching*, the FI score for each visual input object o_d is set to the highest IoU match with any ground-truth question-relevant object o_{gt}. Concretely:

$$s_{o_d} = max_{r \in GT} IoU(o_d, o^r_{gt}), \tag{7.1}$$

where GT is the set of question-relevant ground-truth objects. In this context, the calculated FI score s_{o_d} can be interpreted as a measure of confidence that the object is relevant to the question, which is a reasonable way of compensating for the lack of insight into the object's actual informational content.
In *semantic matching*, any detected object that sufficiently matches the location ($IoU > 0.5$) *and* fully matches the content of any ground-truth question-relevant object, is identified as question-relevant with full confidence. We therefore assign the maximum FI score to such input objects. Similarly, input objects that do not meet these requirements all receive the minimum FI score.

7.4.3 Symbolic features

To gain a firm grasp on the informational content of the visual input, we engineer object-based *symbolic* visual features instead of using standard *sub-symbolic* features, which are commonly extracted from a late layer of an object detector like Faster R-CNN Ren et al. (2015). The process for creating

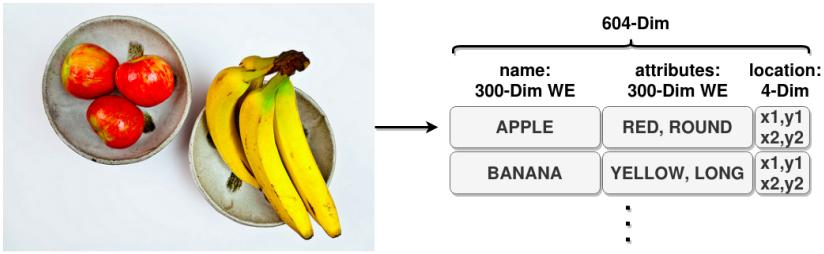

Figure 7.2 – Symbolic features.

symbolic features used in this chapter was described in Chapter 5. We include an overview here for better context.

Each symbolic feature vector represents an object in the image and carries information about its name, attributes and location. We encode each object with two stacked 300-dim GloVe word embeddings Pennington et al. (2014) and 4-dim for coordinates. Figure 7.2 shows an illustration of the makeup of symbolic feature vectors. We construct three symbolic visual representations:

- **Detection (DET) features** are assembled from outputs of a scene graph generator. They represent the standard use case.
- **Infusion (INF) features** are based on DET features, but are minimally "infused" with relevant, question-dependent information to enable what we call "True VG", i.e., training where the required feature content is present in the input and thus intended conditions for VG-methods are met. Note, that Infusion is applied in training only.
  For each training question, we identify which relevant ground-truth objects are a) misrecognized (i.e., a meaningful spatial match ($IoU > 0.5$) exists in DET but semantic match does not), or b) missing entirely (no meaningful spatial match exists). Missing objects are introduced as new objects, assembled from image annotations. Misrecognized objects are adjusted to match the content of the corresponding ground-truth object (e.g., by replacing wrong object names and/or attributes with those given in annotations).

Concrete implementation-related details for these features in the context of this chapter's experiments are described in Appendix D.1. The used scene graph generator is described in detail in Appendix A.

7.4.4 Used Datasets

Primary experiments are performed with the GQA dataset Hudson and Manning (2019), which provides detailed scene graphs and semantic relevance

			Test		True VG Subsets	
Dataset	Train	Dev	ID	OOD	TVG-ID	TVG-OOD
GQA-CP-large	580k	107k	161k	161k	43k	42k
VQA-HAT-CP	32k	6k	4.1k	5.9k	1.1k	1.6k

Table 7.1 – Sample counts for the used ID/OOD splits.

annotations for most of its questions. Moreover, the GQA dataset uses template-generated questions that explicitly refer to information given in its scene graph annotations, thereby creating ideal conditions for our investigations.
In secondary experiments, we use the VQAv1-based (Antol et al., 2015) VQA-HAT dataset (Das et al., 2016), which provides relevance annotations as spatial heat maps over raw images without exact ties to specific ground-truth objects. The crowd-sourced commonsense-type questions in VQAv1 generally exhibit a much weaker connection to the image annotations than questions in GQA.

Specifically, we use the two ID/OOD data splits GQA-CP-large and VQA-HAT-CP from Ying et al. (2022), which were both created with the "Changing Priors" (CP) approach used in the creation of the OOD split VQA-CP in Agrawal et al. (2018). The CP approach redistributes all samples from the original dataset such that the new train and OOD test set have different prior distributions of answers for every question type. Note that we only train with questions that have meaningful relevance cues for all used visual feature types (i.e., we do not use questions for which zero relevant objects were detected at $IoU > 0.5$) to achieve a fair comparison across model variants. Training set numbers listed in Table 7.1 reflect this selection. Similarly, all test sets for VQA-HAT-CP are reduced to the more challenging "other"-type questions (as opposed to questions with yes/no or number answers), as recommended for testing with the VQA dataset in Teney et al. (2020).

7.4.5 Used VQA Models

The classic single-hop attention-based model UpDn (Anderson et al., 2018) traditionally takes center-stage in VQA's VG research on account of its no-frills design which minimizes the risk of results being influenced by an unexpected interplay between VG-methods and additional complex mechanisms. We additionally confirm GQA results with the more powerful Transformer-based model LXMERT (Tan and Bansal, 2019) in a separate section. For training details of both models see Appendix D.2.

7.4.6 Used VG-methods

We evaluate four VG-methods:

(1) **AttAlign** aligns a model's attention weights over visual input objects with the human-based FI scores. The model is trained by adding a cosine similarity-based loss to the standard VQA task loss.
(2) **HINT** (Selvaraju et al., 2019) aligns GradCAM (Selvaraju et al., 2017) determined FI score rankings in the model with those given by human-based FI scores.
(3) **SCR** (Wu and Mooney, 2019) identifies a set of (ir)relevant objects by ranking human-based FI scores. The model is penalized if 1) an irrelevant object receives a higher GradCAM determined FI score (w.r.t. the ground-truth answer) than the top-scoring relevant object, and 2) if the top-scoring relevant object gets an even higher score for wrong answers.
(4) **VisFIS** (Ying et al., 2022) is a high-performing ensemble of "Right for Right Reasons" objectives, which includes cosine similarity-based FI alignment between human-based and model-based FI scores, as well as data augmentation based on (ir)relevant object sets determined by thresholding human-based FI scores.

7.5 Impact on VQA Performance

Detailed numerical results for tests on UpDn models are listed in Table 7.2 for reference. Important accuracy results that we discuss in detail in this section are illustrated in Figure 7.3. Note that all models are tested exclusively with DET features, even when trained with INF features. In the following discussion, we highlight certain results to illustrate the problem with Impaired VG.

7.5.1 Impairment 1: Testing

We first evaluate impact of VG-methods on the True VG (TVG) tests, which only contain questions that are accompanied by complete relevant visual content. Figure 7.3 shows VQA performance impact on UpDn when trained with a VG-method. The left column shows impact on full ID/OOD tests (labeled "Impaired"), while the right column shows impact on TVG tests ("True"). In comparison, Figure 7.3 shows that impact of VG-methods is considerably more pronounced in TVG testing (right column), confirming that Impairment 1 is indeed problematic for evaluating VG-methods, as their

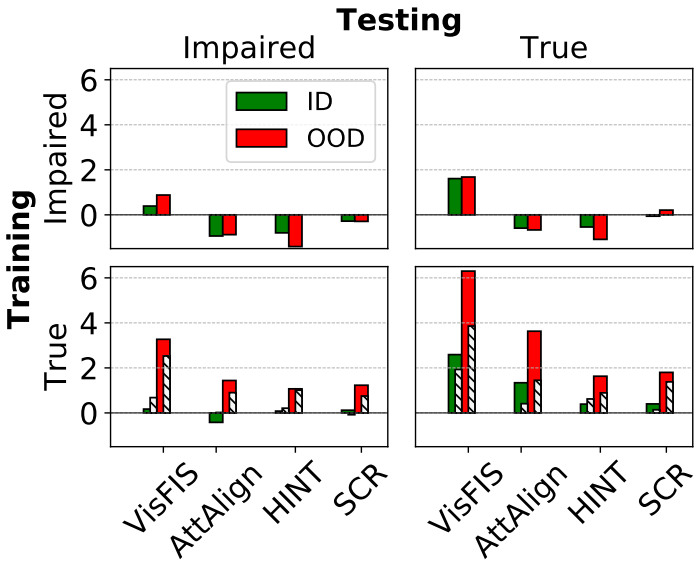

Figure 7.3 – Accuracy improvements (in absolute percent) from VG-methods compared to respective UpDn baselines. Numerical results are listed in Table 7.2 (baselines listed in first two lines). Training (y-axis): DET features with spatial matching ("Impaired", top row), and INF features with semantic matching ("True", bottom row). Striped bars: INF features with spatial matching. Testing (x-axis): Full test ("Impaired") or TVG subset ("True").

impact is significantly muted in testing. Moreover, these results suggest that VG-methods can be considerably more effective than previously suggested in related work where VG-method analysis is conducted based on impaired evaluation practices (Shrestha et al., 2020; Ying et al., 2022).

7.5.2 Impairment 2: Training

Training with INF features and semantically-matched cues. Figure 7.3, y-axis, shows accuracy improvements when applying VG-methods in training under "Impaired VG" and "True VG" settings. "Impaired" training is performed with DET features and spatially matched cues (=standard case), while "True" training is performed with INF features and semantically matched cues (=intended case). Comparing top and bottom rows in Figure 7.3 reveals considerably greater impact of VG-methods in True VG training (bottom row), particularly for TVG tests in OOD. This shows that Impairment 2 is indeed a source of significant result distortion which may lead to misjudgement of a VG-method's efficacy.

Spatially vs. semantically matched cues. Striped bars in the bottom row in Figure 7.3 illustrate the results of "True VG" training using FI scores from spatial matching instead of semantic matching. Differences between the two matching types are particularly noticeable in TVG testing (bottom right) and demonstrate that using semantically matched cues can amplify impact of VG-methods substantially.

7.5.3 Impaired VG vs. True VG

In summary, evaluations in "Impaired VG" settings (Figure 7.3, top left) present rather weak evidence to suggest that VG-methods are particularly beneficial to VQA performance, while "True VG" (bottom right) reveals that they can in fact be *very* effective when relevant visual information is present in the input. Particularly interesting here is the strong positive impact of the AttAlign method, which has been repeatedly declared ineffective in previous work, where conclusions were drawn based on the demonstrated, impaired evaluations (cf. Selvaraju et al. (2019); Ying et al. (2022)).

UpDn Training		Accuracy (All / TVG)		$FPVG_+$ (spatial / semantic)	
VG-method	Features	ID	OOD	TVG-ID	TVG-OOD
n/a	DET	62.12 / 75.22	43.18 / 55.92	26.82 / 12.92	25.32 / 12.92
	INF	61.16 / 78.78	45.40 / 62.54	32.19 / 18.07	31.03 / 17.35
VisFIS	DET	**62.51** / 76.83	44.06 / 57.60	30.68 / 14.81	29.52 / 14.47
	INF-spa	61.84 / 80.71	47.93 / 66.40	34.59 / 18.77	33.55 / 18.14
	INF-sem	61.33 / **81.37**	**48.67** / **68.84**	**37.15** / **22.51**	**36.66** / **22.32**
AttAlign	DET	61.18 / 74.63	42.30 / 55.25	27.98 / 13.43	26.16 / 12.96
	INF-spa	61.18 / 79.19	46.30 / 63.99	33.37 / 18.01	31.86 / 17.35
	INF-sem	60.74 / 80.12	46.84 / 66.17	35.79 / 21.42	35.21 / 21.10
HINT	DET	61.32 / 74.68	41.77 / 54.83	25.67 / 12.51	24.69 / 12.31
	INF-spa	61.37 / 79.40	46.41 / 63.43	32.83 / 18.34	31.94 / 17.53
	INF-sem	61.24 / 79.17	46.47 / 64.17	33.81 / 18.95	33.11 / 18.55
SCR	DET	61.84 / 75.16	42.89 / 56.13	26.34 / 12.98	25.21 / 12.28
	INF-spa	61.08 / 78.92	46.15 / 63.92	33.04 / 18.66	32.11 / 17.91
	INF-sem	61.28 / 79.18	46.63 / 64.34	33.10 / 18.76	32.11 / 18.09

Table 7.2 – For reference: UpDn results on GQA-CP-large. The most relevant results from this table are illustrated in Figure 7.3 and Figure 7.4. For discussions of these results, see the respective figures and surrounding text.

7.6 Impact on Visual Grounding Quality

Following recommendations by Shrestha et al. (2020) to confirm VG improvements with a dedicated metric, we investigate the impact of VG-methods on a model's VG quality using FPVG (introduced in Chapter 4).

7.6.1 Relevance matching in FPVG

FPVG measures a model's VG quality by confirming the model's reliance on question-relevant objects during answer inference. As with VG-method training, identifying question-relevant cue objects is a defining step in FPVG (which it is in any VG metric that aims to measure "plausible" VG, i.e., a model's VG w.r.t. visual objects deemed plausibly relevant to answer a given question). The identification of relevant objects in FPVG follows the same procedure as the determination of FI scores for VG-method training (see Chapter 7.4.2), i.e., it can be performed by spatial or semantic matching. FPVG was originally introduced with spatial matching on sub-symbolic visual features. Given that semantic matching provides the means of more accurately pinpointing the set of question-relevant (and irrelevant) objects in the visual input, we expect it to be able to provide more precise VG measurements than location matching. We report FPVG results for both spatial and semantic matching, but surmise that semantic matching leads to a better-defined VG measurement in principle[4].

As only the TVG subsets provide full semantic matches, we discuss FPVG results on those subsets.

7.6.2 Discussion

Full numerical results for FPVG are listed in Table 7.2. We highlight the most important results in Figure 7.4.

True VG training produces stronger VG than Impaired VG. Figure 7.4, left, shows $FPVG_{+}$ measurements (i.e., the percentage of well-grounded questions in testing) in each evaluated UpDn model. True VG models

[4]We note that FPVG's goal is to quantify a certain desired manifestation of VG in VQA, which is described as a model's faithful reliance on plausibly question-relevant objects during answer inference (cf. Chapter 4). It is vital to identify these objects correctly in order to produce a meaningful measurement in accordance with this goal. Semantic matching leverages more available information from given relevance annotations to identify relevant input objects than spatial matching does. Therefore, we conclude that semantic matching provides more accurate object matches which enables more precise VG measurements with FPVG.

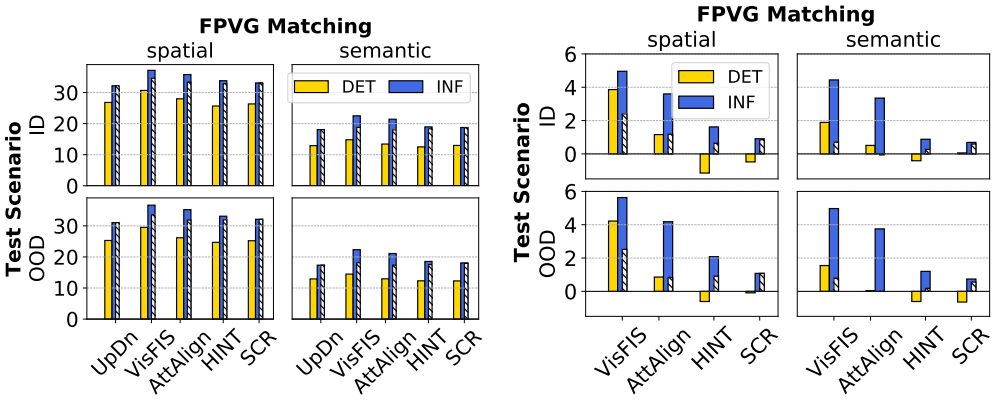

Figure 7.4 – $FPVG_+$ measured on TVG subsets (ID/OOD) for UpDn. Numerical results are listed in Table 7.2. Left: Absolute $FPVG_+$ measurements. Right: $FPVG_+$ improvements (in absolute percent) compared to respective UpDn baselines. Columns categorize the matching method used for FPVG (see Chapter 7.6). Striped bars show results for INF-based models trained with spatial matching.

(blue bars) achieve considerably higher levels of $FPVG_+$ than Impaired VG models (yellow bars) across all examined settings. This includes baseline models trained without VG-methods, which confirms that VG manifestation in models is generally held back when question-relevant information is not consistently provided.

True VG training enables considerably higher $FPVG_+$ improvements than Impaired VG. Figure 7.4, right, shows $FPVG_+$ improvements from training with VG-methods compared to respective UpDn baselines. Improvements are considerably stronger in True VG models (blue) than in Impaired VG models (yellow). This is especially true for FPVG with semantically matched targets (see right column of each block, labeled "semantic").
We interpret these results as additional evidence confirming the influence of Impairment 2 (muted impact of VG-methods due to Impaired VG training). In other words, VG-methods function more effectively in True VG training, where they can consistently align a model's inference with accurate and recurring (i.e., more stable) visual cue objects.

Training with semantic matching has a considerably more positive effect on $FPVG_+$ than spatial matching. Striped bars in Figure 7.4 show $FPVG_+$ results for INF models trained with VG-methods and spatial matching. $FPVG_+$ is substantially more boosted with semantic matching (shown as blue bars). This illustrates that VG-methods can improve a model's VG quality more effectively when training with more accurate guidance.

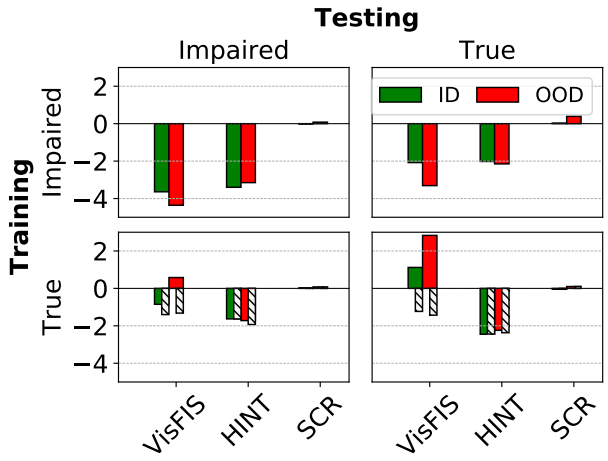

Figure 7.5 – Accuracy improvements (in absolute percent) for LXMERT+VG-methods compared to baseline LXMERT. Numerical results are listed in Table 7.3 (baselines listed in first two lines). Training (y-axis): DET features with location-matched cues ("Impaired") or INF features with content-matched cues ("True"). Striped bars mark results when using location-matched cues instead. Testing (x-axis): Full test ("Impaired") or True Grounding subset ("True").

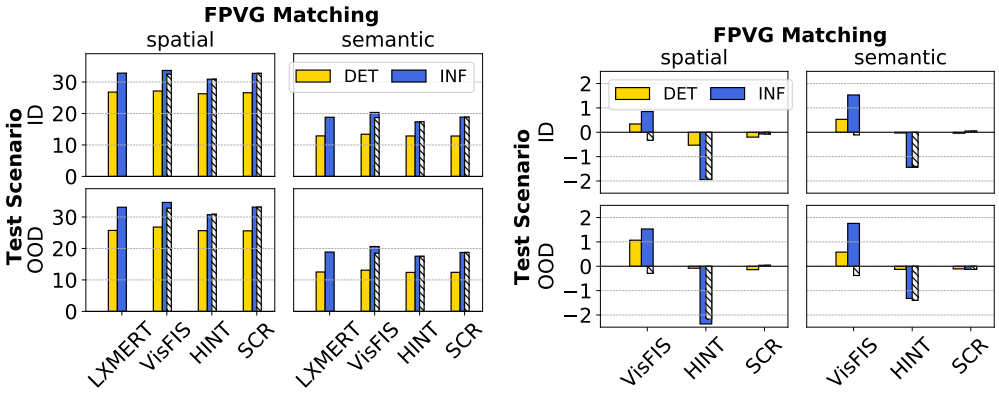

Figure 7.6 – $FPVG_+$ measured on TVG subsets (ID/OOD) for LXMERT. Numerical results are listed in Table 7.3. Left: Absolute $FPVG_+$ measurements. Right: $FPVG_+$ improvements (in absolute percent) compared to respective LXMERT baselines. Columns categorize the matching method used for FPVG (see Chapter 7.6). Striped bars show results for INF-based models trained with spatial matching instead of semantic matching.

7.7 Corroborating Evaluations: LXMERT

We conduct the same tests for LXMERT that we did for UpDn. These evaluations overall corroborate the insights gained for UpDn. All numerical

results for LXMERT on GQA-CP-large are listed in Table 7.3 for reference. Certain important results are illustrated in Figure 7.5 (changes in accuracy) and Figure 7.6 ($FPVG_+$ measurements), respectively. Observations made for LXMERT's evaluations generally match those for UpDn without providing additional insights. We summarize notable model behavior in the following:

1. Overall more favorable model behavior when training with VG-methods in True VG settings (compared to Impaired VG) in both accuracy and $FPVG_+$.
2. Strong negative impact to $FPVG_+$ when applying HINT in True VG settings, as well as a lack thereof in Impaired VG setting (see Figure 7.6, right, yellow vs. blue bars).

W.r.t. (1): We observe no accuracy improvements for VisFIS under Impaired VG settings (see Figure 7.5, top row), while under True VG settings, improvements are made in particular in OOD settings (bottom row). W.r.t. (2): While accuracy degradations for HINT in True VG and Impaired VG settings are comparable on the TVG subsets (see Figure 7.5, right column), we find that $FPVG_+$ measurements are expectedly negatively impacted (to a similar degree as accuracy) *only* in True VG training (see Figure 7.6, right, blue vs. yellow bars). Similarly, observed VisFIS-driven accuracy improvements for True VG are accompanied by expected $FPVG_+$ improvements, while measurements for Impaired VG show contradicting readings. In other words, model accuracy and $FPVG_+$ exhibit stronger signs of a positive correlation with each other in True VG settings.

LXMERT Training		Accuracy (All / TVG)		$FPVG_+$ (spatial / semantic)	
VG-method	Features	ID	OOD	TVG-ID	TVG-OOD
n/a	DET	**67.83** / 80.29	51.79 / 64.18	26.80 / 12.88	25.73 / 12.52
	INF	64.95 / 82.75	52.69 / 69.90	32.83 / 18.80	33.10 / 18.86
VisFIS	DET	64.19 / 78.21	47.43 / 60.87	27.14 / 13.41	26.80 / 13.10
	INF-spa	63.55 / 81.52	51.37 / 68.47	32.50 / 18.69	32.81 / 18.48
	INF-sem	64.10 / **83.87**	**53.27** / **72.73**	**33.68** / **20.33**	**34.63** / **20.62**
HINT	DET	64.43 / 78.27	48.64 / 62.03	26.27 / 12.85	25.65 / 12.39
	INF-spa	63.32 / 80.31	50.76 / 67.53	30.90 / 17.40	30.93 / 17.46
	INF-sem	63.32 / 80.30	50.97 / 67.66	30.89 / 17.36	30.73 / 17.54
SCR	DET	67.82 / 80.32	51.87 / 64.57	26.60 / 12.84	25.59 / 12.41
	INF-spa	64.97 / 82.72	52.77 / 70.00	32.76 / 18.85	33.14 / 18.73
	INF-sem	64.97 / 82.71	52.77 / 70.00	32.76 / 18.85	33.14 / 18.73

Table 7.3 – For reference: LXMERT results on GQA-CP-large. The most relevant results from this table are illustrated in Figure 7.5 and Figure 7.6.

In summary, we observe a more congruent connection between accuracy and FPVG under True VG settings.

7.8 True VG Analysis with VQA-HAT

In this section, we adopt the True VG methodology for the VQAv1-based VQA-HAT-CP datset.

7.8.1 Preliminaries

Model training. The UpDn training setup for VQA-HAT follows the settings used in Ying et al. (2022) (see also Appendix D.2). We report averages and max deviation from the mean for five differently seeded training runs for each evaluated model variants.
Visual features. We create symbolic features based on object detector outputs shared by Anderson et al. (2018), which provide object names and attributes for 36 objects per image.
Evaluation. Accuracy calculations for VQA-HAT follow Antol et al. (2015): A question is 100% correct if the returned answer was given by at least 3 of 10 annotaters per question and otherwise contributes fractional scores (calculated as $min(\frac{\#annotaters\ with\ returned\ answer}{3}, 1)$)to overall accuracy.

7.8.2 Dataset Challenges

There are significant challenges to handle when evaluating VQA-HAT with True VG methodology.

Sparse image annotations

Infusion uses an image's object-level annotations to verify semantic matches between question-relevant objects and detected input objects. Furthermore, annotations act as source for infusing missing information. For VQA-HAT, we use the official MS-COCO image annotations Lin et al. (2014), which is the underlying image database for the VQAv1 dataset. MS-COCO annotates images with 80 different object names (GQA: 1702 classes). No attribute annotations are provided.

Relevance annotations

VQA-HAT's relevance annotations are given as spatial heat maps, as opposed to GQA's unambiguous pointers to annotated objects in the image. As

relevant ground-truth objects are not directly provided in VQA-HAT, but required for Infusion, the first step here is to identify them in the image annotations corresponding to each heat map.

A commonly used metric to determine object importance in VQA-HAT's heat maps was introduced in Selvaraju et al. (2019). We adopt this metric to first calculate importance scores for all ground-truth objects in the image annotations and then apply a threshold to determine which objects are question-relevant. Concretely, the importance score for ground-truth object o is calculated as:

$$score_o = E_{in}(o)/(E_{in}(o) + E_{out}(o)), \tag{7.2}$$

where $E_{\{in,out\}}$ is the averaged pixel-level importance value inside and outside the ground-truth object's given bounding box in the provided heat map. Following Ying et al. (2022), we apply a threshold of 0.55 to these scores to establish the relevance status of each ground-truth object. Finally, FI scores for the visual input are determined with the same spatial or semantic matching procedure used for GQA (as described in Chapter 7.4.2).

7.8.3 Discussion

We highlight the most relevant results in Figure 7.7. Full numerical results for VQA-HAT-CP are listed in Table 7.4 and Table 7.5 for reference. As with GQA, we investigate both accuracy and $FPVG_+$ changes for indicators that show how True VG evaluations improve our understanding of VG-method impact.

Contrary to GQA, impact of True VG settings for VQA-HAT is considerably less pronounced and the observed trends are incongruous. For instance, strong improvements in $FPVG_+$ for VisFIS and AttAlign (Figure 7.7, right) do not consistently translate to better accuracy (Figure 7.7, left), as they did for GQA. Furthermore, differences in accuracy impact between full tests (=Impaired test case) and the TVG subsets (=True/intended test case) are inconsistent across VG-methods, unlike for GQA, where they are strongly amplified for all VG-methods in True VG training (comparing bottom row in VQA-HAT's Figure 7.7, left, with GQA's Figure 7.3).

In summary, we observe no overarching, consistent benefits from adapting the True VG methodology to VQA-HAT, neither from a) introducing annotated object information in training, nor b) in testing cases where this annotated information is already present (TVG subsets). Definitive conclusions that explain these observations cannot be drawn due to the much less favorable dataset conditions compared to GQA, which hinder a proper True VG setup, combined with the different, more challenging nature of VQA-HAT's questions.

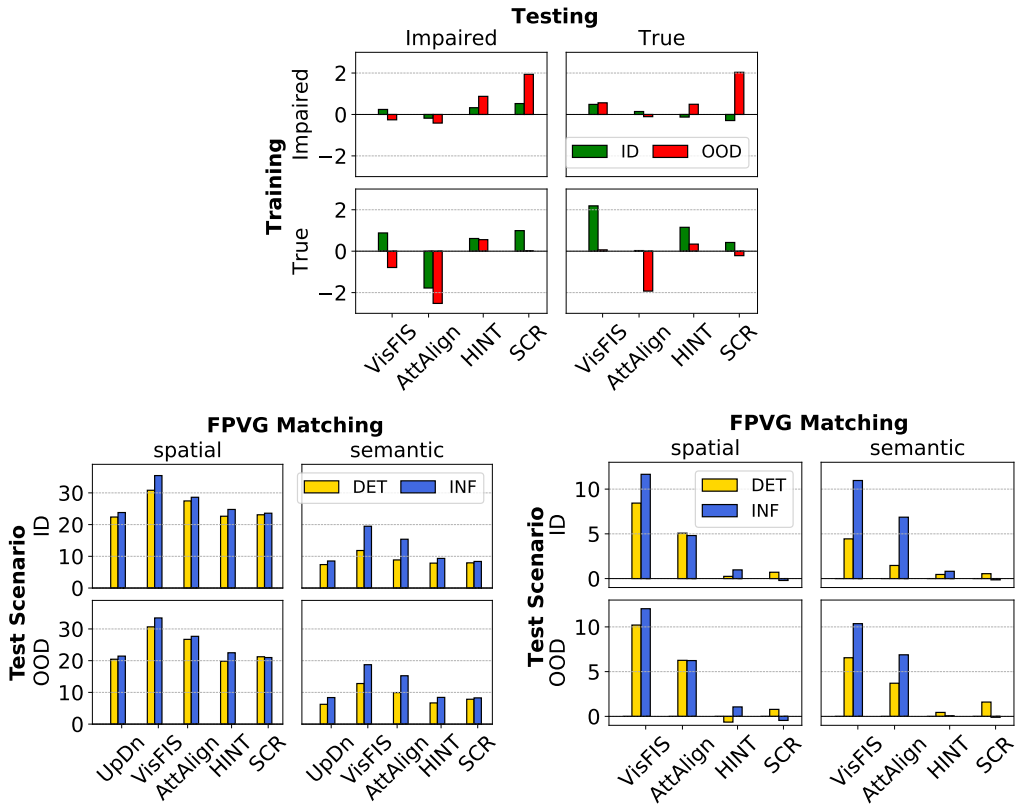

Figure 7.7 – VQA-HAT-CP: Accuracy and $FPVG_+$ measurements (all values based on averages over five differently seeded UpDn models). Numerical results are listed in Table 7.4 (accuracy) and 7.5 ($FPVG_+$). See captions of Figure 7.3 and Figure 7.4 for a general description of these histograms.

VQA-HAT uses commonsense-type questions with less explicit references to relevant objects in the scene than GQA's retrieval-type questions. While the latter type can be expected to work well with symbolic features carrying object-centric information, the requirements for the former type's appropriate informational content is much less clear. This complicates the determination of what an optimal analytical setup for VG in VQA-HAT might entail.

In conclusion, we recommend performing True VG analysis with the GQA dataset, where the dataset conditions are more favorable and the VG task is better defined.

UpDn Training		Accuracy			
VG-method	Features	ID	OOD	TVG-ID	TVG-OOD
n/a	DET	35.28±0.36	24.38±2.21	38.00±1.23	26.84±2.61
	INF	35.24±0.56	25.65±2.06	37.80±1.07	28.56±2.27
VisFIS	DET	35.52±2.89	24.13±2.94	38.49±3.71	27.40±3.00
	INF	36.12±1.09	24.87±1.49	**39.99**±2.12	28.62±1.68
AttAlign	DET	35.10±1.25	23.97±2.14	38.14±0.77	26.74±1.90
	INF	33.46±1.38	23.13±2.33	37.83±2.05	26.63±2.85
HINT	DET	35.61±0.66	25.26±1.96	37.88±0.86	27.33±2.44
	INF	35.85±0.34	26.21±0.98	38.95±0.58	**28.90**±1.82
SCR	DET	35.81±1.26	**26.32**±1.87	37.71±1.43	28.88±1.37
	INF	**36.23**±0.51	25.66±1.80	38.22±0.94	28.34±1.96

Table 7.4 – For reference: Accuracies for UpDn evaluated on VQA-HAT-CP (only "other"-type questions). We report average results and maximum deviation over five differently seeded training runs per model variant. The most relevant results from this table are illustrated in Figure 7.7, top row.

UpDn Training		$FPVG_+$ (spatial)		$FPVG_+$ (semantic)	
VG-method	Features	TVG-ID	TVG-OOD	TVG-ID	TVG-OOD
n/a	DET	22.39±1.92	20.45±3.06	7.39±1.02	6.24±1.01
	INF	23.80±2.40	21.44±4.02	8.52±1.54	8.37±1.50
VisFIS	DET	30.82±11.99	30.66±9.68	11.83±3.25	12.79±2.33
	INF	**35.46**±3.97	**33.47**±3.64	**19.48**±1.74	**18.73**±2.82
AttAlign	DET	27.46±1.45	26.72±1.96	8.86±1.43	9.94±1.97
	INF	28.61±3.22	27.67±2.26	15.38±3.45	15.25±2.46
HINT	DET	22.63±2.74	19.81±3.87	7.85±2.32	6.68±2.02
	INF	24.77±2.40	22.49±3.02	9.34±1.21	8.42±1.03
SCR	DET	23.09±2.22	21.23±2.70	7.94±1.04	7.84±2.11
	INF	23.59±1.66	20.98±2.96	8.38±0.99	8.27±1.46

Table 7.5 – For reference: $FPVG_+$ measurements for UpDn evaluated on VQA-HAT-CP (only "other"-type questions). We report average results and maximum deviation over five differently seeded training runs per model variant. The most relevant results from this table are illustrated in Figure 7.7, bottom row.

7.9 Limitations of True VG

Transferability to other datasets. As demonstrated for VQA-HAT, transferability of the introduced True VG methodology is conditioned on the availability of appropriate annotations and may be better suited to question types where the notion of what constitutes "relevant content" is well-defined and understood (such as in GQA). This limits the transferability of the methodology (and arguably poses a challenge for VG research in VQA as a whole). For future research and in order to properly evaluate True VG on different types of questions and datasets, such as the commonsense-type questions of the VQA dataset, it is at minimum necessary to ensure a high quantity (several 100k) of unambiguous and high-quality relevance annotations (as opposed to vague heat-maps), as well as an appropriate richness of image annotations that cover all annotated question-relevant objects. It should be verified that all annotated questions can in fact be answered correctly by a human who is only presented with the identified relevant objects. The availability of such combined annotations facilitates a much more thorough investigation of a model's capabilities to infer an answer from relevant image regions, regardless of the type of questions. This would provide a solid foundation from which then investigations could be launched to develop a better understanding about what level of visual descriptiveness is required to achieve (additional) VG-induced performance improvements in datasets with various types of questions.

Usability of Infusion and semantic matching beyond analysis. Our experiments are based on non-standard symbolic features, which are not the first choice for image representations in current high-performing VQA models. While the evaluated fixes to Impaired VG in this work are leading to improvements in accuracy and VG and are useful for analysis of VG-methods, they cannot be applied to, e.g., higher performing symbolic features as is. We could imagine future work that investigates how feature content interpretation and Infusion can be applied to standard symbolic features to unshackle these ideas from its bond with symbolic features and make them useful beyond analysis.

7.10 Summary

In this chapter, we have shown that current training and testing practices for VG-methods in VQA are impaired and therefore unable to reflect their full potential benefits for VQA models. We have proposed a methodology

to optimize evaluation conditions to allow for a more thorough analysis of VG-method impact. Our investigations have shown that when conditions are optimized, VG-methods can elicit considerably stronger performance improvements in VQA models in both VG and accuracy, boosting OOD performance in particular.

The findings presented in this chapter show that the potential influence of VG on OOD scenarios has been severely misrepresented by the use of problematic evaluation methodologies in VG-method research. Our analysis may therefore have a profound impact on the general perception of how valuable VG actually is for OOD generalization.

Chapter 8

Visually Grounded Reasoning

In the course of this thesis, we have introduced a number of methods that facilitate investigations of the influence of Visual Grounding (VG) on Visual Question Answering (VQA) models. In this chapter, we use these methods to great effect for defining and empirically validating a novel theoretical model that provides new insights into the role of VG in VQA generalization in the context of shortcut (SC) learning.

In Chapter 8.2, we derive a definition of a concept we call "Visually Grounded Reasoning" (VGR) in VQA: The *VGR Proposition*, which we formulate as a propositional logic statement, formally defines the integral roles of VG and Reasoning in VQA SC learning. Using the VGR Proposition, we are able to show that current Out-of-Distribution (OOD) tests do not provide a reliable basis for conclusions regarding VG-related SC learning in VQA models, despite frequently being employed in such context (Chapter 8.4.2). Our findings also explain why a strong correlation between OOD accuracy and VG has been eluding the field so far (as observed in Shrestha et al. (2020) and Ying et al. (2022) as well as in our own VG investigations in Chapter 6). Based on these insights, we propose an approach to build so-called "shortcut-free" (SC-free) tests in accordance with the VGR Proposition. We show that our SC-free tests are much less likely to be solved by VG-related SC exploitation and are thus better suited for unveiling and analyzing VG-related SC learning in VQA models (Chapter 8.5.1). An evaluation of training recommendations to succeed on such an SC-free test completes our investigations.

8.1 Introduction

Deductive reasoning is a logical process that involves drawing a conclusion from a set of premises. In the context of VQA, we may map this generic description onto a more concrete process: VQA models operate based on a learned set of *decision rules* (cf. Chapter 2.4) that dictate their logical process of deductive reasoning at test time. Models pass to conclusions (i.e., return answers) through the application of these decision rules, while relying on premises represented by 1) the model's "understanding" of the world as learned in training, and 2) current observations represented by image information and the given question[1]. While interpretations such as these may help develop an intuitive understanding of VQA model behavior, an exact *technical* definition that encapsulates the VQA Reasoning process is more challenging to devise. The lack of such a technical definition, in turn, makes it difficult to quantify a model's reasoning capabilities in isolation, which is a prerequisite for accurately measuring a VQA model's progress in this respect.

In an effort to measure VQA Reasoning in their work, Kervadec (2021) circumvents this issue by opting to identify VQA Reasoning by "what it is not", and settles on defining it as "the opposite of exploiting biases and spurious correlation in the training data" and, therefore, the opposite of exploiting SCs[2] (Kervadec (2021), p. 14). Consequently, Kervadec (2021) argues that VQA Reasoning can be quantified as accuracy in Out-of-Distribution (OOD) tests (Kervadec (2021), p. 15), which are widely used to uncover SC exploitation

[1]This description is an analogy between VQA Reasoning and the structure of a logical argument called the syllogism, a common form of deductive reasoning which consists of a major and a minor premise (i.e., a *general truth* about the world (major premise) and an *observation* of an instance belonging to that world (minor premise)), as well as a conclusion w.r.t. the observation. Example of a syllogism (Mill, 1851): "All men are mortal", is a *general truth*. "Sokrates is a man", is an *observation*. "Therefore, Sokrates is mortal.", is the *conclusion*.

[2]Shortcut exploitation in Deep Learning (DL) is framed as undesirable behavior due to its negative impact on generalization and stimulation of unintended model behavior (Geirhos et al., 2020). However, this does not mean that SCs cannot be helpful in certain circumstances in practice. For instance, when crucial information is missing from the input, leveraging SCs in the form of dataset biases may lead to correct answers, while instead following an inference process that heavily depends on the correctness of presented observations (i.e., the model input) may not. Human reasoning can involve making an informed decision about whether resorting to "educated guessing" provides a better chance of success in certain situations. In the context of this thesis, we consider this type of decision making with its added layer of complexity as out of scope, and focus on the distinction of unintended SC exploitation vs. intended observation-driven inference. Hence, in this thesis, we adopt Kervadec (2021)'s interpretation that SC exploitation does not constitute VQA Reasoning.

and estimate VG impact (see Chapter 2.4 and Chapter 2.5.1).
For Kervadec (2021)'s argument to be logically sound, OOD tests are necessarily assumed to consist of questions that cannot be answered correctly when exploiting SCs, including dataset biases, as the mentioned VQA Reasoning definition implies that correct answers in OOD tests can only be the result of correct VQA Reasoning. A similar line of thinking is implied by the common use of OOD testing as a way to infer VG quality in models by their OOD accuracy, i.e., without the use of a dedicated VG metric (Agrawal et al., 2018; Selvaraju et al., 2019; Zhang et al., 2016; Goyal et al., 2017; Agrawal et al., 2016). In the course of this chapter, we show why the assumption that OOD accuracy reflects VG quality is highly problematic and that it may be the main reason why the role of VG in VQA has not been clearly understood so far.

We begin our investigations by building on the principle idea of circumventing the need for an exact definition of VQA Reasoning and its quantification by some dedicated Reasoning metric. We use this idea as a starting point to develop a theoretical model that ultimately explains the role of VG in the VQA inference process. Our theoretical model, called *Visually Grounded Reasoning* (VGR), frames VQA inference as a co-dependency between three parties that describes VQA model behavior in the context of SC learning. These three involved parties are *Answer Accuracy*, *VQA Reasoning* (hereafter "Reasoning") and *Visual Grounding* (VG). Notably, our model formally establishes VG as a crucial component in SC learning in VQA that must be considered separately from Reasoning to properly explain VQA model behavior.

8.2 Visually Grounded Reasoning

In the following, we capture the dependencies between Accuracy, Reasoning and VG in VQA using formal propositional logic statements. We do this in the context of "SC-free" testing. SC-free testing for VQA involves tests that require the use of human-intended decision rules to be solved correctly. In other words, an ideal SC-free test cannot be solved by shortcut exploitation (cf. definitions in Geirhos et al. (2020)). In this work, we identify Reasoning and Visual Grounding as two necessary types of human-intended decision rules to solve an ideal SC-free test. Below, we develop a model that explains the role of each component in that context.

Case	Reasoning	Answer	Validity
1	✗	✗	True
~~2~~	~~✗~~	~~✓~~	~~False~~
3	✓	✗	True
4	✓	✓	True

Table 8.1 – Truth table representing VQA-OOD behavior (i.e., presumably SC-free) under the definition of Reasoning by Kervadec et al. (2021). Note that Case 2 is invalid under this definition, i.e., a True answer cannot result from False Reasoning.

8.2.1 Reasoning

Following Kervadec's proposition to measure Reasoning by accuracy in SC-free testing, we define VQA behavior for SC-free tests with the following logic statement:

Hypothesis 1.

$$Answer \rightarrow Reasoning \tag{8.1}$$

Reformulated as contraposition:

$$\neg Reasoning \rightarrow \neg Answer \tag{8.2}$$

In words: A correct answer implicates correct Reasoning. Incorrect Reasoning results in a wrong answer. Table 8.1 lists the formal truth table for Hypothesis 1.

8.2.2 Visual Grounding

Under Kervadec's Hypothesis 1, SC-free behavior is made out to be defined by only two factors, Answer Accuracy and Reasoning. Notably, Hypothesis 1 does not explicitly mention VG, even though *VG is an axiomatic component of VQA modeling by definition of the VQA task*, which is to answer questions about image contents. Moreover, the process of answering questions while *bypassing correct VG* is by definition of the VQA task a SC. We infer from this argument that VG *must* take a significant role in the explanation of SC-free behavior. As Hypothesis 1 does not explicitly mention VG, VG must necessarily be implicitly included in Reasoning in order for Hypothesis 1 to truly describe SC-free behavior as it originally set out to do — lest it be considered invalid.

Case	Visual Grounding	Answer	Validity
1	✗	✗	True
~~2~~	~~✗~~	~~✓~~	~~False~~
3	✓	✗	True
4	✓	✓	True

Table 8.2 – Truth table representing VQA-OOD behavior (i.e., presumably SC-free) under the definition of VG in Hypothesis 2. Note that Case 2 is invalid under this definition, i.e., a True answer cannot result from False VG.

In order to show that VG's involvement cannot be part of Reasoning in Hypothesis 1 and needs to be established as a separate component, we first establish an argument for VG in its own right and circle back to Hypothesis 1's validity later on.

We first describe VG involvement in SC-free testing with a logic statement of its own. On grounds of the axiomatic nature of VG's involvement in VQA, we posit that a correct Answer necessarily requires correct VG in an SC-free scenario.

Hypothesis 2.

$$Answer \rightarrow VG \tag{8.3}$$

Reformulated as contraposition:

$$\neg VG \rightarrow \neg Answer \tag{8.4}$$

In words: A correct answer implicates correct VG. Incorrect VG results in a wrong answer.
The truth table for Hypothesis 2 is shown in Table 8.2. Note that unlike Hypothesis 1, Hypothesis 2 is not intended to fully encapsulate SC-free testing behavior in VQA on its own, but rather to describe only the VG aspect of it.

8.2.3 The VGR Proposition

Combining the two Hypotheses for Reasoning and VG, we arrive at our proposition for describing VQA behavior in ideal SC-free testing.

VGR Proposition.

$$Answer \rightarrow Reasoning \wedge VG \tag{8.5}$$

Case	Reasoning (RE)	Visual Grounding (VG)	Answer (A)	Hypothesis 1 $A \rightarrow RE$	Hypothesis 2 $A \rightarrow VG$	VGR Proposition $A \rightarrow RE \wedge VG$	FPVG
1.1	✗	✗	✗	True	True	True	BGW
1.2	✗	✓	✗	True	True	True	GGW
~~2.1~~	~~✗~~	~~✗~~	~~✓~~	~~False~~	~~False~~	~~False~~	~~**BGC**~~
~~2.2~~	~~✗~~	~~✓~~	~~✓~~	~~False~~	~~True~~	~~False~~	~~GGC~~
3.1	✓	✗	✗	True	True	True	BGW
3.2	✓	✓	✗	True	True	True	GGW
~~4.1~~	~~✓~~	~~✗~~	~~✓~~	~~True~~	~~False~~	~~False~~	~~**BGC**~~
4.2	✓	✓	✓	True	True	True	GGC

Table 8.3 – All 8 cases of VQA model behavior under the defined logic system for SC-free testing, the VGR Proposition. Evaluation of Answers, given the status of VG and Reasoning, and their corresponding categorization with FPVG. Strikethrough lines represent cases that are invalid under the confines of the VGR Proposition.

Reformulated as contraposition:

$$\neg(Reasoning \wedge VG) \rightarrow \neg Answer \tag{8.6}$$

In words: A correct answer necessitates both, proper Reasoning and VG. Without both, proper Reasoning and VG, the answer cannot be correct. Table 8.3 lists the formal truth table for the VGR Proposition, which we discuss in detail further down.

8.2.4 Hypothesis 1 is flawed as description of SC-free test behavior.

With the VGR Proposition defined, we can now circle back to Kervadec's original Hypothesis 1 that was presented as a complete description of SC-free testing behavior. Table 8.3 lists all eight cases, or permutations, of the three involved aspects, i.e., Answer Accuracy, Reasoning and VG. Here, we find that on the basis of Hypothesis 1 (which conflates VG and Reasoning) Case 4.1 represents valid behavior in SC-free testing. However, Case 4.1 *refutes the axiomatic involvement of VG in VQA* as defined by Hypothesis 2. Concretely, Hypothesis 2 states that Case 4.1 (i.e., a correct Answer given based on incorrect VG) is in fact an SC and thus *invalid* SC-free testing behavior. As a result, we surmise that *VG cannot be conflated with Reasoning* as done in Kervadec's Hypothesis 1 and must be explicitly considered alongside of it as a separate component on equal footing, thereby establishing the VGR Proposition.

GGC	**G**ood **G**rounding, **C**orrect Answer
GGW	**G**ood **G**rounding, **W**rong Answer
BGC	**B**ad **G**rounding, **C**orrect Answer
BGW	**B**ad **G**rounding, **W**rong Answer
$FPVG_+$	Good Grounding (GGC + GGW)
$FPVG_-$	Bad Grounding (BGC + BGW)
Accuracy	GGC + BGC

Table 8.4 – FPVG categories.

8.3 Model Behavior in SC-free Testing

In this section, we bridge the gap from theoretical VQA model behavior in ideal SC-free tests under the VGR Proposition to VQA model behavior that can actually be observed and measured in practice during experimentation with OOD tests.

To interpret VQA model behavior in practice, we use FPVG (Chapter 4), our metric for measuring VG in VQA, which provides a crucial second angle for result categorization on top of Accuracy. In FPVG, every evaluated question is assigned one of four categories based on measurements of a) Answer Accuracy (correct or wrong) and b) VG (good or bad). These four categories are summarized by two overarching VG-based categories, $FPVG_+$ and $FPVG_-$. All FPVG categories and their meaning are listed in Table 8.4 for reference.

Within the confines of the VGR Proposition, we can identify general patterns of VQA model behavior (in terms of measured VG and Accuracy) that a model is bound to exhibit when it is tested with an SC-free test that conforms to VGR. By confirming the presence of these patterns in concrete test results, we can then verify whether a used test set is SC-free according to VGR.

To facilitate the identification of the desired model behavior, we map the first four FPVG categories listed in Table 8.4 onto the eight cases of model behavior described by the VGR Proposition in Table 8.3. For instance, the first FPVG category in Table 8.4, "GGC" represents Case 4.2 of valid model behavior in Table 8.3 (see entry in rightmost column). Based on the mapping in column "FPVG" in Table 8.3, we can unambiguously determine an expected categorization (i.e., general patterns) of results in SC-free testing under VGR in practice. We summarize these patterns as the following Corollaries.

8.3.1 Corollaries of SC-free Testing Behavior

Corollary 1: BGC is zero. When mapping FPVG's four categories onto VGR Proposition's logic system in Table 8.3, we find that one of the categories has no valid matches under the VGR Proposition, namely the category of BGC (i.e., questions that evaluate as correctly answered despite bad VG). Cases 2.1 and 4.1 in Table 8.3 match the conditions for BGC, but are invalid under the VGR Proposition. This lack of a valid mapping suggests that *the category of BGC captures cases that exploit shortcuts*. Formally, this category violates the VGR Proposition for ideal SC-free tests. Hence, the portion of questions aligning with the BGC category should be "zero"[3] in ideal SC-free testing, regardless of the tested VQA model:

$$BGC = 0 \tag{8.7}$$

Corollary 2: GGC equals Accuracy. Since $BGC = 0$ in ideal SC-free tests, we can reformulate FPVG's formula for Accuracy (defined in Table 8.4, bottom line) as follows:

$$\begin{aligned} Acc &= GGC + BGC \\ &= GGC \end{aligned} \tag{8.8}$$

Corollary 3: Accuracy measures true SC-free performance. Only one valid condition matches category GGC in Table 8.3: Case 4.2. Hence, we can unambiguously determine that all questions in GGC have met the conditions described by Case 4.2, as no other valid conditions match this category. The conditions for Case 4.2 are met if a model answers a question correctly while using correct Reasoning and correct VG, which is our definition of SC-free accuracy. By application of Corollary 2, which states that GGC equals Accuracy, we find:

$$\begin{aligned} Acc &= GGC \\ &= Acc_{SC-free} \end{aligned} \tag{8.9}$$

Thus, Accuracy measures true SC-free performance.

Corollary 4: Accuracy cannot surpass FPVG$_+$. This result is derived by applying Corollary 2 to FPVG's formula of VG (i.e., $FPVG_+$, Table 8.4,

[3] In practice, a small percentage of questions is expected to be assigned to BGC due to other factors that are difficult to control, including inaccurate annotations and randomly coinciding answer behavior supporting such assignment.

Line 5):

$$\begin{aligned} FPVG_{+} &= GGC + GGW \\ &= Acc_{SC-free} + GGW \\ &\geq Acc_{SC-free} \end{aligned} \tag{8.10}$$

In other words, per the VGR Proposition, models cannot achieve higher Accuracy than $FPVG_{+}$ in an ideal SC-free test.

8.3.2 Limitation: Theory vs. Practice

The VGR Proposition and its four corollaries describe VQA model behavior in SC-free testing assuming ideal, fully controlled testing conditions. Such ideal conditions are unlikely to be fully enforced in empirical testing and therefore a small degree of transgressions of the corollaries[4] are to be expected and may be unavoidable. Formally, the VGR Proposition does not account for the impact of deviations from ideal testing conditions that are encountered in practice, as these are not straightforward to quantify. Therefore, when using the VGR Proposition for analysis of a test set in practice, we recommend considering how closely the corollaries are approximated, rather than verifying strict and exact observance, when determining their violation.

8.4 Do current OOD Tests reflect SC-free VQA performance?

In this section, we investigate if current OOD tests are appropriate measures of shortcut learning in VQA models under the VGR Proposition. An OOD test is deemed inappropriate for SC-free testing under the VGR Proposition, if there is a VQA model that produces OOD test results that are in gross violation of VGR.

8.4.1 Experiment Preliminaries

Datasets. We evaluate three OOD dataset splits that are based on GQA's balanced split (Hudson and Manning, 2019) and VQA-HAT (Das et al., 2016), which itself is based on VQAv1 (Antol et al., 2015). GQA and the VQA dataset are described in detail in Chapter 2.3. All data splits have been introduced as proxies for quantifying the problem of shortcut learning in VQA

[4]Such as mentioned above for Corollary 1, where, in practice, we do not expect BGC to be *exactly zero*, even though Corollary 1 calls for that.

			Testing	
Dataset	Train	Dev	ID	OOD
VQA-HAT-CP	32k	6k	3.3k	4.7k
GQA-CP-large	580k	107k	139k	137k
GQA-OOD	828k	20k	29k	15k

Table 8.5 – Sample counts for the evaluated data splits.

models. Sample counts are listed in Table 8.5.
GQA-CP-large and VQA-HAT-CP (Ying et al., 2022) are ID/OOD splits which were created by a (re)distribution of questions in VQA-HAT and GQA, respectively, following the "Changing Priors" (CP) approach described in Agrawal et al. (2018). CP redistributes all samples from a dataset such that the new train and OOD test set have different prior distributions of answers for every question type. The third dataset, GQA-OOD (Kervadec et al., 2021), does not modify GQA's train set, but redistributes questions in GQA's val set based on answer frequencies per question type. Rare and frequent answers per question type in GQA's val set are categorized as tail (OOD) and head (ID), respectively.

Visual relevance annotations that point out question-relevant objects in the scene and are required for measuring VG quality are available for all three data splits we use: GQA provides these for most of its questions in the form of detailed VG references (including rich image annotations of the question-relevant objects). VQA-HAT provides relevance annotations in the form of human-provided heat maps of relevant image regions for a small subset of questions in VQAv1, which is reflected in the size of the dataset (see Table 8.5).

VQA Models. We run experiments with two VQA models: **UpDn** (Anderson et al., 2018), a classic, single-hop attention-based model, and **LXMERT** (Tan and Bansal, 2019), a Transformer-based (Vaswani et al., 2017), BERT-like model (Devlin et al., 2019) trained following a pre-train/fine-tune paradigm. Both models are described in Chapter 2.2.
Models are trained with each of the three datasets. LXMERT's pre-training was performed twice (once for each GQA-based split) to ensure the intended sample distributions in each individual split. LXMERT was not (pre-)trained with VQA-HAT-CP on account of its small size.
Since the same training procedures were used here as in Chapter 7 (UpDn, LXMERT), we defer to that chapter for additional training details.

OOD Training		ID						OOD					
Dataset	Model	Acc	$FPVG_+$	GGC	GGW	BGC	BGW	Acc	$FPVG_+$	GGC	GGW	BGC	BGW
GQA-CP-large	UpDn	64.54	22.99	17.50	5.50	47.05	29.96	45.48	21.98	13.79	8.19	31.69	46.33
	LXMERT	69.60	23.71	19.50	4.21	50.10	26.19	53.38	23.40	16.35	7.05	37.04	39.57
GQA-OOD	UpDn	63.48	25.41	19.17	6.24	44.31	30.28	42.91	26.14	15.62	10.52	27.29	46.58
	LXMERT	65.85	25.41	19.98	5.42	45.87	28.72	46.76	24.85	15.83	9.02	30.93	44.22
VQA-HAT-CP	UpDn	52.63	11.42	7.04	4.38	50.45	38.13	35.87	11.99	6.25	5.74	34.59	53.42

Table 8.6 – Accuracy and FPVG results for three current OOD tests, evaluated with UpDn and LXMERT. Analysis of the OOD results reveals that all three tests violate the VGR Proposition (e.g., very high BGC violates Corollary 1) and are therefore unsuitable to measure SC-free performance (see discussion in Chapter 8.4.2).
Sidenote: Reported accuracy numbers for VQA-HAT-CP are lower than GGC and BGC results indicate (GGC+BGC normally equals accuracy). This is because accuracy for VQA-HAT-CP is calculated, as is customary for this dataset, based on fractional correctness scores (see metric definition in Chapter 7.8.1), while FPVG categories do not use such fractional scores.

Visual Features. We use *symbolic visual features* in all evaluated models. Symbolic features are described in Chapter 5. We use the same concrete feature instances that were created and used in Chapter 7.

8.4.2 Result Discussion

Results for ID/OOD splits are listed in Table 8.6. The examined three tests are intended to uncover SC exploitation and act as a proxy measure for a model's Reasoning, VG and generalization capabilities. Therefore, we would expect VQA model behavior to align with the SC-free testing behavior which we derived from the VGR Proposition. Remarkably, however, results in Table 8.6 show that all three examined OOD tests violate the four VGR corollaries of expected model behavior:

- Violation of Corollary 1: BGC represents a *substantial share* of questions, when it should approximate *zero*.
- Violation of Corollary 2 & 3: GGC is considerably *lower* than Accuracy, when it should be in *similar range*.
- Violation of Corollary 4: $FPVG_+$ is far *lower* than Accuracy, when it should be *similar or higher*.

This means that, according to the VGR Proposition, all three OOD tests are unsuitable for quantifying SC-free behavior, and their accuracy-based evaluation does not provide clear and reliable evidence that a model in fact avoids shortcut exploitation.

While in clear violation of the VGR Proposition, results in Table 8.6 do provide some positive indication that these OOD tests at least move in the intended direction of presenting less opportunities to models for exploiting SCs than ID tests: High BGC numbers in all ID tests (44% to 51% of questions) indicate that models benefit a great deal from exploiting SCs. In comparison, OOD tests exhibit considerably lower BGC (27% to 38% of questions). Nevertheless, the BGC category still covers a substantial amount of questions in OOD testing, indicating successful SC exploitation. This suggests that there are factors involved that hinder an analysis of a model's SC behavior on the basis of Accuracy on these tests alone. Furthermore, these results provide important indicators as to why a strong linear correlation between Accuracy and VG has been eluding the field so far (see, e.g., investigations in Shrestha et al. (2020); Ying et al. (2022) and Chapter 2.5.1), namely that correct VG is not a strict requirement for a correct answer in these OOD tests.

Conclusion. We believe this to be a significant finding, as evaluations with existing OOD tests like GQA-OOD, GQA-CP-large and VQA-HAT-CP contradict the VGR Proposition in particular by *exhibiting a large percentage of questions in the BGC category*. This raises strong questions regarding the suitability of these tests for evaluating certain model properties for which they are employed:

1. *Determining the impact of VG:* Various works have explained low(er) accuracy in OOD (vs. ID) testing with a model's disregard of relevant visual information (notably Agrawal et al. (2018); Goyal et al. (2017); Selvaraju et al. (2019)). Consequently, approaches to strengthen a model's VG have been introduced that successfully improve OOD accuracy (Wu and Mooney, 2019; Selvaraju et al., 2019; Ying et al., 2022). As a twist in this narrative, Shrestha et al. (2020) showed that methods in Wu and Mooney (2019); Selvaraju et al. (2019) achieve similar OOD accuracy gains even when intentionally learning *wrong* VG. Similarly, (Ying et al., 2022) reported that VG measurements and OOD accuracy lack a strong correlation. Such reports of inconsistent, unpredictable impact of VG are particularly harmful to the notion of VG as a necessary component in VQA generalization[5], especially because these findings are reported against the backdrop of OOD tests that were created to reflect VG-related shortcomings of VQA models (Agrawal et al., 2018, 2016; Johnson et al., 2016; Zhang et al., 2016; Goyal et al., 2017) and are therefore expected to be directly affected

[5]OOD testing in VQA is also framed as generalization, see Teney et al. (2020)

by changes to VG quality. Our results help understand such reports by showing that, contrary to the mentioned underlying expectations for OOD tests, proper VG can actually still be bypassed by SC exploitation to produce a large number of correct answers. This means, there is no clear dependency relationship between VG and accuracy. Therefore, neither must changes to VG quality be reflected in accuracy, nor can we expect to find a strong correlation between VG and OOD accuracy. As a result, conclusions concerning the impact of VG in VQA, which are based only on accuracy in these OOD tests, should not be considered reliable.

2. *Determining generalization capabilities:* OOD evaluations have been described as a way to evaluate a model's ability to generalize beyond dataset biases and used in that context (e.g., Teney et al. (2020), Hudson and Manning (2019)). Such framing of OOD testing assumes that a model's reliance on such dataset biases (which constitute unintended decision rules and are therefore shortcuts by definition) should not lead to success in those tests. However, our investigations have revealed extensive successful exploitation of dataset biases in OOD tests, which suggests that OOD accuracy may not be particularly indicative of a model's generalization skills in this context.

3. *Determining Reasoning capabilities:* Estimating a model's Reasoning capabilities by OOD accuracy is one of the main motivations behind the introduction of Kervadec et al.'s test GQA-OOD. In the context of their definition of Reasoning as "the opposite of SC learning" (Kervadec (2021) and Chapter 8.1), GQA-OOD would not be expected to allow SC exploitation to significantly influence accuracy. However, as shown above, extensive SC exploitation is still involved in generating the majority of correct answers, suggesting that accuracy results in these tests may not be particularly indicative of a model's Reasoning skills under Kervadec (2021)'s definition.

In summary, *we caution against interpreting model accuracy on these OOD tests as reliable evidence for conclusions in the contexts discussed above.*

8.5 SC-free Testing in VGR

Considering the findings made above, we propose a new test split called GQA-AUG (AUG for augmentation). In addition to a random Q/A prior

distribution in its OOD test split, GQA-AUG is also specifically designed to require the use of proper VG.

8.5.1 Creating an SC-free test for VQA using augmentation

The three investigated OOD tests in Sec. 8.4 focus on controlling Q/A prior distributions in OOD tests. Our investigations have revealed that this approach is unsuitable for SC-free testing in VQA. This is because VQA models run considerable risk of VG-related SC exploitation which these tests do not explicitly account for. In the following, we outline our approach for creating a new test set that only includes questions that are unlikely to resolve to a correct answer without leveraging correct VG. We then show that such a test set is a much more accurate manifestation of SC-free testing under VGR.

GQA-AUG creation. We use the Information Retrieval-based GQA dataset as starting point to develop the new test split GQA-AUG. We focus on GQA's *query*-type questions, which we consider the hardest question type in GQA due to their higher number of plausible answer options in the dataset (as opposed to, say, *verify*-type questions about an object's existence, where the only plausible answers in the training data are "yes" or "no"). If we assume a uniform answer distribution in testing, *query*-type questions such as "What color is the pictured vehicle?" are on average less likely to be resolved correctly without successful retrieval of the required information from relevant visual input. Thus, generally speaking, correct answers for *query*-type questions in particular have a stronger innate dependency on correct VG, which makes these questions an obvious choice for our test. It is worth noting that in VQA practice this stronger requirement for VG in *query*-type questions does not automatically insulate them from being solved by SC exploitation. E.g., SC exploitation can still occur in large numbers if Q/A-prior-based SCs ("educated guesses") also offer a likely (or even more likely) option for success, as we have observed in the GQA-based tests examined above.

With this in mind, we propose **GQA-AUG**, an SC-free test under the VGR Proposition, based on GQA. We create this test by the following steps (illustration shown in Figure 8.1):

1. Identify *query*-type questions in GQA's balanced val set with answers consisting of object names (e.g., dog, car, etc.) which are explicitly tied to image content in given relevance annotations. We call this set of questions **AUG-ID**.

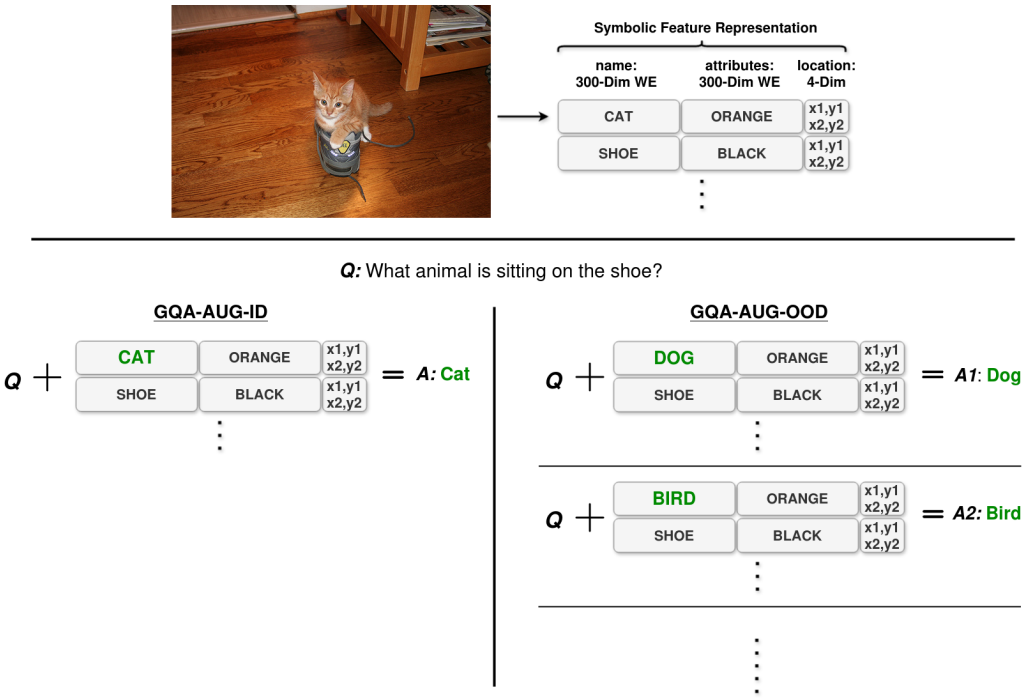

Figure 8.1 – Example of samples in the GQA-AUG dataset (creation process described in Chapter 8.5.1). GQA-AUG-ID (lower left) contains the original GQA test sample with detected symbolic feature representation (shown at the top) and ground-truth answer ("cat"). GQA-AUG-OOD (lower right) contains new samples that differ in both answer ("dog", "bird", ...) and feature content (appropriately modified to support the answer). The question is not changed.

2. Identify the name category (e.g., animal, vehicle, etc.) of the original name/answer.
3. Generate *new Q/A samples* by replacing the answer with up to ten uniquely sampled names from the same object category (e.g., cat → dog, bird, etc.). The question remains the same.
4. Generate *new image representations* for each new Q/A sample to support the new answer. Concretely, we replace relevant parts of the image representation such that it aligns with the new answer (e.g., if the original image supported the original answer "cat", we replace relevant image content to support the new answer "dog", "bird", etc.). We call this set of resulting questions **AUG-OOD**.

This approach of creating new *query*-type questions *alongside visual feature augmentation* is intended to minimize the possibility of correct answers being returned without correct VG. At the same time, step 3 reduces the possibility

			Testing	
Dataset	Train	Dev	AUG-ID	AUG-OOD
GQA-AUG	828k	20k	17k	161k

Table 8.7 – Sample counts for GQA-AUG. We use questions from GQA's balanced train set as train/dev set in our experiments.

that educated guesses based on learned Q/A priors (in GQA's balanced split) can be exploited to return a correct answer.

It is worth mentioning two other image augmentation techniques used in VQA for training (MUTANT (Gokhale et al., 2020)) and training and testing (SwapMix (Gupta et al., 2022)). Our image augmentation process is restricted to testing and distinguishes itself by operating in controlled, symbolic feature space which allows more exact feature content manipulation than both MUTANT and SwapMix. These two approaches either manipulate raw images before input to an object detector (MUTANT), or copy object-based features from other images based on annotations of the image without controlling for actually represented content in feature space (SwapMix), making them less exact than our approach.

GQA-AUG Statistics. Dataset numbers for the GQA-AUG test splits are listed in Table 8.7. AUG-ID consists of 17k unmodified *query*-type questions taken from the GQA balanced val split. Based on AUG-ID, we synthesize 161k new samples using the augmentation process described above, i.e., up to ten generated samples per question in AUG-ID, based on the number of unique object names in the involved object category. On average, AUG-OOD modifies 4.2 query-related objects per question. The average of all question-relevant objects per question in AUG-OOD is 6.6. In 35.7% of questions in AUG-OOD the set of question-relevant objects overlaps fully with the set of modified objects.

8.5.2 Is AUG-OOD an SC-free test?

In this section, we seek to verify if AUG-OOD is an SC-free test according to the VGR Proposition. For experiments, we use the same UpDn and LXMERT model instances (without retraining) as for GQA-OOD in Section 8.4. Table 8.8 shows evaluation results for GQA-AUG for the two models.
We validate each of the four corollaries of the VGR Proposition in the following.

		AUG-ID						AUG-OOD					
Dataset	Model	Acc	$FPVG_+$	GGC	GGW	BGC	BGW	Acc	$FPVG_+$	GGC	GGW	BGC	BGW
GQA-AUG	UpDn	40.60	32.80	20.17	12.63	20.43	46.77	15.73	26.72	12.07	14.65	3.65	69.63
	LXMERT	43.11	30.80	19.47	11.34	23.64	45.55	13.31	18.93	8.39	10.54	4.92	76.15

Table 8.8 – Accuracy and FPVG results for GQA-AUG. AUG-OOD results show a close approximation of the VGR Proposition (e.g., very low BGC approximates Corollary 1), supporting its categorization as an SC-free test. Detailed discussion in Chapter 8.5.2.

Corollary 1: Low BGC All models in Table 8.9 post low numbers in BGC for AUG-OOD. In particular, the values in BGC are substantially lower than those observed in other OOD tests (Sec. 8.4). This is in line with our stipulations for SC-free tests, which states that correct answers should in theory not be returned without being based on question-relevant information.

Corollary 4: Accuracy is not higher than FPVG$_+$ Corollary 4 is met by all models except VLR which exceeds $FPVG_+$ by a small margin. We consider this within acceptable range (see discussion in Sec. 8.3.2). A non-ideal BGC value (exceeding zero) observed for both models is contributing to this result. It is also worth pointing out that AUG-ID results show excessive violation of Corollary 4 in both models, reaffirming that the original set of questions (AUG-ID) is not suitable for SC-free testing.

Corollary 2&3: Accuracy is equal to GGC While Accuracy is not exactly equal to GGC (because some residual BGC still exists), Corollary 2 & 3 are far better approximated in AUG-OOD than in the OOD tests examined in Sec. 8.4.

8.5.3 AUG-OOD Summary

We have shown that results on AUG-OOD approximate the VGR-derived corollaries of SC-free testing to a substantially higher degree than other examined OOD tests. In conclusion, we find AUG-OOD to be a significantly better approximation of SC-free testing than other OOD tests examined in this work.

		Accuracy ($FPVG_+$)		FPVG (AUG-OOD)			
Model	Features	AUG-ID	AUG-OOD	GGC	GGW	BGC	BGW
UpDn	DET	40.60 (32.80)	15.73 (26.72)	12.07	14.65	3.65	69.63
LXMERT	DET	43.11 (30.80)	13.31 (18.93)	8.39	10.54	4.92	76.15
MMN	DET	44.10 (39.78)	20.66 (26.79)	16.59	10.20	4.07	69.15
MAC	DET	41.61 (31.33)	15.45 (21.42)	9.63	11.78	5.81	72.77
VLR	n/a	39.16 (46.57)	79.35 (77.25)	75.45	1.80	3.90	18.85

Table 8.9 – GQA-AUG: Results for five models trained with DET features.

8.6 Improving Performance on GQA-AUG

8.6.1 Baseline VQA Models

In order to get a broader overview of baseline VQA performances on AUG-OOD, we evaluate three additional VQA models of different architectural designs: **MAC** (Hudson and Manning, 2018) is a multi-hop attention-based model developed for GQA-type Visual Reasoning. **MMN** (Chen et al., 2021) is a Transformer-based model which uses question programs generated by an independently trained question parser instead of the otherwise common word embeddings used for raw question input. Our **VLR** system (Chapter 3) uses symbolic, programmed inference with focus on VG and GQA's IR-type questions. **UpDn** and **LXMERT** have already been introduced in Chapter 8.4.1. All five models are trained and tested with the same GQA-AUG data split (sample counts listed in Table 8.7).
Training procedures for UpDn and LXMERT are referenced in Chapter 8.4.1, MAC and MMN follow the general training procedures referenced in Chapter 6. VLR's setup is described in Chapter 3. As mentioned earlier in this chapter, all models are trained with *symbolic* visual features, which we introduced in Chapter 5 and used in Chapter 7.

Result Discussion

Results for all five models trained with detected symbolic visual features (DET) are listed in Table 8.9. Our first observation is that results for the three additional models (MAC, MMN, VLR) exhibit tendencies that are similar to what we found for UpDn and LXMERT, thus re-confirming GQA-AUG as an SC-free test.

We further find relatively high GGW numbers compared to GGC in most models. In the context of VGR, we interpret this as an indication of unused VG potential that may be untapped because of underdeveloped Reasoning

capabilities. We can describe Reasoning in the context of GQA-AUG as a model's ability to correctly answer IR-type questions by involvement of correct VG. VLR is the only model in the table that was designed specifically to implement human-intended decision rules under an IR-based paradigm, which is reflected by its exceptional success on AUG-OOD.

While initial evaluation results for AUG-OOD are satisfactory in terms of approximation of the VGR Proposition, answer accuracy on this test is quite low for all classifier-based models. This is in stark contrast to the high performance of VLR, which is based on programmed rule-based inference instead of *learned* inference. VLR implements human-intended decision rules and thereby avoids the possibility of learning to exploit shortcuts (cf. Geirhos et al. (2020)). We interpret the contrasting results between VLR and the other four models as an indicator that these four models have forgone the adoption of human-intended decision rules of inference in favor of learning to exploit shortcuts — which is what an SC-free test is supposed to evaluate. In other words, AUG-OOD is working as intended.

In the following, we investigate if better performance on AUG-OOD is achievable through adequate training. According to the VGR Proposition, high accuracy on this test signifies the manifestation of both VG and appropriate Reasoning: In theory, this would mean that the model's inference process relies on human-intended decision rules and avoids shortcut exploitation. Hence, we seek to answer the question: Can we train these models to pick up human-intended decision rules?

8.6.2 Learning IR-type Reasoning

GQA can be categorized as an IR-type VQA dataset, as the vast majority of its questions are generated by filling in question-templates with explicit information taken from annotated scene graphs of involved images. Therefore, we see no obvious dataset-related reasons preventing a model to learn IR-type Reasoning capabilities to perform well on AUG-OOD, when simply trained with GQA. Consequently, we look for the problem's source elsewhere. Following the VG analysis in Chapter 7, we form the hypothesis that proper adoption of IR-type Reasoning is prevented by inconsistently presented visual targets in the input from which models learn not only much stronger proper VG (as shown in Chapter 7), but human-intended decision rules for performing Information Retrieval (for the VQA task), as well.

To test this hypothesis, we adopt the True VG methodology introduced in Chapter 7 and train each model with input features that are minimally

		Accuracy ($FPVG_+$)		FPVG (AUG-OOD)			
Model	Features	AUG-ID	AUG-OOD	GGC	GGW	BGC	BGW
UpDn	INF	42.42 (41.95)	63.36 (65.80)	60.28	5.52	3.08	31.12
LXMERT	INF	42.73 (41.59)	58.93 (57.43)	54.81	2.62	4.12	38.45
MMN	INF	44.41 (43.93)	58.62 (57.16)	54.38	2.78	4.24	38.59
MAC	INF	41.60 (43.00)	62.83 (63.74)	60.00	3.74	2.83	33.42
VLR	n/a	39.16 (46.57)	79.35 (77.25)	75.45	1.80	3.90	18.85

Table 8.10 – GQA-AUG: Results for five models trained with Information Infusion. Accuracy and $FPVG_+$ on AUG-OOD is significantly improved compared to standard training, while the VGR Proposition is even better approximated. Discussion in 8.6.2.

modified to contain verified, relevant image content according to the GQA-provided relevance annotations. This method corresponds to the "Information Infusion" approach described in Chapter 5 which was shown to improve training conditions for VG-boosting training methods such as VisFIS (Ying et al., 2022), HINT (Selvaraju et al., 2019) and SCR (Wu and Mooney, 2019). As it turns out, Infusion-based training not only greatly improves general VG quality in models, but also boosts adoption of Reasoning capabilities for IR tasks such as AUG-OOD.

Result Discussion

Numerical results for Infusion-trained (INF) models are listed in Table 8.10. We highlight some of the AUG-OOD results in Figure 8.2. We make the following observations.

Accuracy is greatly improved. The left two plots in Figure 8.2 show that all four re-trained models experience substantial gains in AUG-OOD accuracy (green) and $FPVG_+$ (red), while AUG-ID accuracy (see Table 8.10) remains mostly stable. The latter result is not unexpected, as AUG-ID testing does not control for correct alignment between answer and feature content of question-relevant objects[6], hence any impact of improved IR-type Reasoning that might also be beneficial for AUG-ID is muted. In AUG-OOD, on the other hand, this alignment between answer and visual objects is enforced by design, and Reasoning improvements are reflected clearly.
As a sidenote, it is worth emphasizing that training with Infusion does not

[6]These effects of Infusion have been discussed in detail in Chapter 7. AUG-ID aligns with the "impaired" test case and shows similar result tendencies.

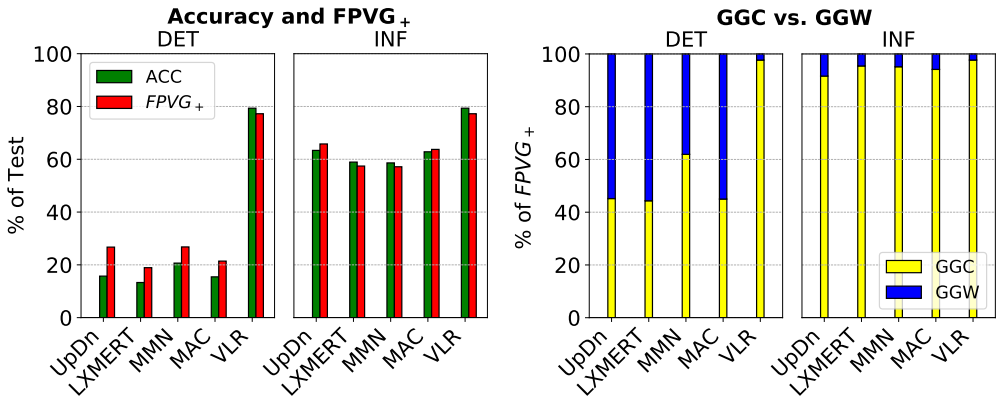

Figure 8.2 – Illustration of AUG-OOD results listed in Table 8.10. Left: Accuracy and $FPVG_+$ on AUG-OOD for DET and INF models. Right: GGC and GGW percentages of $FPVG_+$ in each DET and INF model. Discussion in 8.6.2.

involve any changes to the training set's Q/A prior distribution, meaning, the same Q/A samples are used in both DET and INF. That is to say, improvements on AUG-OOD are not the result of any answer distribution shifts in the training data.

GGC has increased while GGW has shrunk. As illustrated by the two right-hand plots in Figure 8.2, in models trained with Infusion, GGC is considerably higher than GGW and approaching $FPVG_+$. Additionally, $FPVG_+$ (and therefore GGC) aligns much closer with accuracy in INF models than in DET models (notice the differences between red and green bars per model in Figure 8.2, left). This further improves the approximation of the four VGR corollaries. We interpret the more effective involvement of VG (as reflected by increased dominance of GGC over GGW in Figure 8.2, right) as evidence of improved Reasoning: The models have learned to draw on relevant feature content to return correct answers.

VGR Corollaries are better approximated. Finally, these results show that the four VGR Corollaries are even better approximated after Infusion training, thus providing continued validation of GQA-AUG's suitability as SC-free test under the VGR Proposition.

8.7 Analyzing Model Properties with AUG-OOD and VGR

Based on the VGR Proposition and the observed results for AUG-OOD above, we have concluded that INF-trained models have developed significantly stronger Reasoning and VG. In the following, we verify this conclusion once more from another angle. The investigation in this section serves as further demonstration of how VQA model behavior on AUG-OOD can be explained and understood with the help of the VGR Proposition, thereby further confirming its relevance to VQA model analysis.

8.7.1 Reasoning in AUG-OOD

According to the VGR Proposition, both Reasoning and VG are supposed to be necessary to achieve correct answers in AUG-OOD. Since Corollary 2 states that $Acc_{SC-free} = GGC$[7], we also know that GGC reflects correct Reasoning (as well as correct VG). By comparing the unleveraged share of correct VG reflected in GGW to the leveraged share reflected in GGC (remember that $FPVG_+ = GGC + GGW$), we can determine the strength of Reasoning as a measure of how completely VG is leveraged in models. We demonstrate this for concrete model results in the following.

A lack of Reasoning in DET models is reflected in AUG-OOD results (Table 8.11) by a dominance of GGW over GGC (accordingly: low accuracy compared to $FPVG_+$). Improved Reasoning from INF training manifests in far greater dominance of GGC over GGW and a closer alignment of accuracy with $FPVG_+$ (see all INF models in Table 8.11).

According to these interpretations, DET models should see only moderate accuracy impact from pure VG improvements to the models, as their Reasoning evidently cannot fully utilize VG. INF models, on the other hand, should see strong impact to accuracy from VG changes, as their Reasoning knows how to properly leverage and rely on VG. In the following, we investigate if this expected model behavior can indeed be observed in practice.

8.7.2 VG manipulation

We attempt to isolate effects from changing VG quality in a model on AUG-OOD results. Concretely, we manipulate VG quality in INF models using the VG-method VisFIS (Ying et al. (2022), described in detail in Chapter 2.5).

[7] As discussed in Chapter 8.3.2, we accept an approximation of this rule in practice.

			Accuracy ($FPVG_+$)		FPVG (AUG-OOD)			
Model	VG-method	Features	AUG-ID	AUG-OOD	GGC	GGW	BGC	BGW
UpDn	n/a	DET	40.60 (32.80)	15.73 (26.72)	12.07	14.65	3.65	69.63
	n/a	INF	42.42 (41.95)	63.36 (65.80)	60.28	5.52	3.08	31.12
	VisFIS	DET	42.63 (41.39)	16.96 (33.00)	13.98	19.02	2.97	64.02
	VisFIS	INF	42.28 (44.98)	67.56 (68.74)	64.86	3.88	2.71	28.56
	VisFIS-Rm	DET	40.48 (27.43)	9.02 (14.82)	5.45	9.37	3.57	81.61
	VisFIS-Rm	INF	42.62 (36.74)	51.94 (54.30)	47.92	6.38	4.02	41.68

Table 8.11 – GQA-AUG: UpDn trained with VisFIS with regular and randomized guidance ("Rm").

VisFIS has the capacity to *improve* $FPVG_+$ in models, as shown in Chapter 6 and Chapter 7. Similarly, we can *reduce* $FPVG_+$ by training VisFIS with randomized (=inaccurate) feature importance (FI) scores[8].

8.7.3 Model behavior when improving VG

Results for UpDn INF models in Table 8.11 show that training with VisFIS improves both $FPVG_+$ and AUG-OOD accuracy to a similar degree (e.g., UpDn INF improves by about 3% in $FPVG_+$ and about 4% in accuracy, after VisFIS training). DET models, on the other hand, show only weak reception for VG quality improvements, particularly when compared to the magnitude of the observed VG improvements (e.g., UpDn DET improves by about 6% in $FPVG_+$ but only about 1% in accuracy, after VisFIS training). These results confirm our outlined expectations and can be interpreted as follows: 1) INF models have learned the kind of Reasoning that is necessary to solve AUG-OOD. They are now held back primarily by their VG capabilities, and thus improving VG further promises strong additional gains in AUG-OOD accuracy. 2) DET models lack the required Reasoning capabilities to efficiently leverage VG improvements, as discussed above. Improving VG alone thus only has a muted effect on these models and model development should focus on improving Reasoning first.

[8]FI-scores act as guidance that VG-methods like VisFIS use to guide a model's inference. The scores signify the question-relevance of each visual input object. Hence, by randomization of these scores (shuffling them), a model learns to strengthen its reliance on random input objects instead of actually question-relevant ones.

8.7.4 Model behavior when reducing VG

According to our VGR-based expectations, deliberately *reducing* VG in INF models should cause a similarly sized *reduction* of AUG-OOD accuracy. We verify this in additional experiments. Results for INF models trained with "VisFIS-Rm" (randomized VisFIS) in Table 8.11 show that the expected impact on AUG-OOD accuracy alongside a reduction of $FPVG_+$ does indeed manifest (both $FPVG_+$ and accuracy drop by about 11% (UpDn INF → UpDn INF VisFIS-Rm)). This confirms our expectation that a reduction of VG would cause a similarly sized reduction in AUG-OOD accuracy, thus demonstrating the requirement of VG as noted by the VGR Proposition. Meanwhile, Reasoning capabilities remain mostly unaffected by VisFIS training. Changes in Reasoning would be reflected by changes to GGC's share of $FPVG_+$ (remember that $FPVG_+ = GGC + GGW$). Here, GGC's share drops only slightly from about 94% (UpDn INF VisFIS) down to 89% (UpDn INF VisFIS-Rm).
Similarly, DET models trained with "VisFIS-Rm" also show the expected behavior of a model with lower Reasoning, namely that a VG reduction impacts accuracy to a much lesser degree than the size of the VG change might suggest ($FPVG_+$ is reduced by about 12% (UpDn DET → UpDn DET VisFIS-Rm), while accuracy drops only about 7%).

8.7.5 Summary

Our investigations in this section illustrate how the VGR Proposition helps to explain and understand VQA model behavior, which can lead to a better sense of direction for improving performance. The presented results confirm the expected co-dependency of Reasoning and VG, as described by the VGR Proposition. Both properties are required to achieve high AUG-OOD accuracy, and the effects of both model properties can be identified in the results.

8.8 Conclusion

We have introduced the *VGR Proposition*, a propositional logic statement for Visually Grounded Reasoning that formally defines VQA model behavior in shortcut-free testing, which revolves around two axiomatic concepts in VQA: Visual Grounding and Reasoning. The VGR Proposition states that in the absence of shortcut opportunities a VQA model requires both correct Reasoning and correct Visual Grounding to produce a correct answer to a given question.

In utilizing the VGR Proposition, we have shown that current Out-of-Distribution tests are unreliable tools for inferring a VQA model's shortcut exploitation and Visual Grounding from answer accuracy, a purpose for which they are commonly employed. Considering this finding, we have presented an approach for creating VQA tests that better approximate an alignment between answer accuracy and a model's SC-exploitation-free behavior in accordance with VGR.

In summary, in this chapter we have formally established Visual Grounding as a key component of VQA and gained significant insights into its role in VQA generalization and shortcut learning.

Part IV

Conclusion

Chapter 9

Conclusion

9.1 Summary

In this thesis, we have investigated the role of Visual Grounding (VG) in Visual Question Answering (VQA) generalization and shortcut learning. We have developed various methods and methodologies to assist us in our analysis of VG which has led to a number of novel insights and findings. We summarize the thesis' contributions to the field in the following.

In Chapter 3, we have introduced **"VQA by Lattice-based Retrieval" (VLR)**, a VQA system that breaks with the predominant classifier-based modeling paradigm in the VQA field and follows an Information Retrieval-based design. VLR's implementation closely aligns with human-intended decision rules for VQA which explicitly necessitates VG for answer inference. As a result, VLR's propensity for shortcut exploitation is significantly reduced: we have shown that VLR does not suffer from the typical, large performance gap between In-Distribution (ID) and Out-of-Distribution (OOD) test accuracy, unlike other evaluated classifier-based VQA models. Similarly, VLR was also shown to outperform other models on a number of generalization tasks which we have specially developed. We have officially shared these tasks with the research community[1].

With the introduction of our novel VG metric **"Faithful and Plausible Visual Grounding" (FPVG)** in Chapter 4, we have filled the need for a

[1] https://github.com/dreichCSL/GQA_generalization_splits

dedicated, meaningful and accurate metric to measure faithful and plausible VG in VQA. A thorough verification and discussion of its properties as well as advantages over other existing metrics via a series of experiments has attested FPVG with a strong foundation of reliability for subsequent analyses conducted in this thesis. We have shared our implementation of FPVG with the research community[2] for easier adoption of our new metric.

Sub-symbolic visual features extracted from a late network layer of an object detector are widely used as visual input for VQA models. While useful in providing an expressive image representation, their sub-symbolic and opaque nature impede a proper analysis of the visual modality's influence in VQA by obstructing a clear view at the actual content they are carrying. To facilitate deeper investigations into the relevance of visual content for the manifestation of VG, we have described the construction of symbolic visual features for VQA in Chapter 5. Based on these symbolic features, we have developed a method we coined **"Information Infusion"** which enables easy manipulation of represented image content through surgical feature modifications. In Chapters 7 and 8, we have shown how Infusion can be applied in training to improve VQA models in both VG quality and answer accuracy on Out-of-Distribution (OOD) tests in particular. Furthermore, its application as a data augmentation technique in the creation of a new type of OOD tests has been proposed in Chapter 8.

In Chapter 6, we have reported a **large-scale overview of VG quality** in a wide variety of VQA architectures using our metric FPVG. Our results have shown that modern VQA models, although boasting impressive performances in In-Distribution (ID) accuracy, are producing correct answers without the support of proper VG in many cases, which is strong evidence of widespread shortcut exploitation. We have concluded this analysis with investigations of the connection between VG quality and OOD accuracy where, with the help of FPVG, we have uncovered clear tendencies that reflect the importance of a model's VG capabilities for its performance in OOD scenarios: even though each analyzed model architecture was shown to leverage VG quality to a different degree, they all shared a performance sensitivity to VG quality in OOD testing that was much weaker pronounced in ID testing, thus reinforcing the notion that VG plays a significant role in VQA generalization in particular.

In Chapter 7, we have explored a suspicion that common evaluation practices for VG-boosting methods may be problematic and that this may be a contributing factor to the unpredictable nature of VG's impact on VQA performance. In our investigations of current evaluation practices for VG-

[2]https://github.com/dreichCSL/FPVG

boosting training methods, we have shown that training and testing is largely performed with data samples that do not actually allow the manifestation of proper VG on account of missing relevant visual content. We have introduced a methodology, coined as **"True VG"**, to correct this issue and facilitate a more thorough evaluation of VG-methods. The subsequent analysis has led to novel insights regarding the potential of VG-methods for improving VQA performance in OOD accuracy and VG quality in particular. Notably, we have shown that VG-methods are considerably more potent than previous reports suggested, when training and testing conditions are aligned with their intended use-case which involves presence of relevant visual input information. We have shared our implementation with the research community[3] to facilitate reproduction of the involved experiments using the introduced True VG methodology.

Finally, in Chapter 8, we have formally derived the **VGR Proposition**, a novel propositional logic statement describing VQA model behavior on tests that require the application of human-intended decision rules to be solved successfully, which we coined "shortcut-free tests". The VGR Proposition establishes Visual Grounding and Reasoning as two key components of VQA generalization by highlighting the significance of their influences on evaluations of shortcut-free VQA performance. The inception of the VGR Proposition has enabled us to examine a number of prominent OOD tests w.r.t. their value for VG-centric generalization research. By application of the VGR Proposition in our analysis, we have found that the common practice of interpreting answer accuracy on the examined tests as a direct reflection of 1) a model's VG quality, and 2) VG's influence on generalization, follows a flawed concept: we have shown that these tests still offer plenty of opportunities for VG-related shortcut exploitation in particular, which makes VG's impact on accuracy unpredictable. Following these findings, we have proposed an approach for developing new OOD test scenarios that properly reflect the significance of VG in the context of shortcut learning in VQA. Experiments have shown that our test setup prevents VQA models from achieving high accuracy without utilizing proper VG, which is in line with theoretical model behavior described by the VGR Proposition.

In conclusion, the ideas behind VGR represent the culmination of this thesis. Along the way, we have developed a number of novel methods, shared their implementations with the community and conducted extensive experiments and analyses, all of which combined allowed us to gain significant insights into the role of VG in VQA generalization, as summarized above.

[3] https://github.com/dreichCSL/TrueVG

9.2 Closing Remarks

Shortcut learning and the exploitation of dataset biases and spurious correlations presents a significant challenge in the pursuit of strong generalization capabilities in VQA models.

A VQA model's inference process can be described as the application of various acquired decision rules leading to the model's output. In this context, shortcut learning is characterized by a model's acquisition of decision rules based solely on their accuracy-related efficacy on limited training data, instead of their plausibility to humans (cf. Geirhos et al. (2020)). This is problematic for a model's generalization capabilities insofar as human-intended decision rules are expected to generalize far better to new settings than decision rules that were formed on the basis of dataset biases and spurious correlations which are prevalent in current VQA datasets. Visual Grounding in a VQA model, i.e., a model's faithful reliance on plausibly relevant visual information to answer a question, is one such human-intended decision rule and is therefore considered an indicator of a model's generalization capabilities. While the enforcement of VG in VQA models may not always lead to answer accuracy improvements in typical In-Distribution benchmarks, and its impact may be muted even in certain Out-of-Distribution tests, we argue that such meaningful indicators of generalization prowess in models should not be ignored in the absence of exhaustive generalization testing. This is especially true for VQA, where the development of tests to evaluate models in every possible scenario remains out of reach due to the high task complexity.

The role of VG in VQA generalization is that of a meaningful indicator of a model's generalization capabilities by VG's definition as a human-intended decision rule. We hope this thesis is able to contribute to the establishment of VG in VQA as a significant model property that implicates and improves a model's utility and reliability in the vast, real world.

Bibliography

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 9525–9536, Red Hook, NY, USA, 2018. Curran Associates Inc.

Vedika Agarwal, Rakshith Shetty, and Mario Fritz. Towards Causal VQA: Revealing and Reducing Spurious Correlations by Invariant and Covariant Semantic Editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. Analyzing the behavior of visual question answering models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1955–1960, Austin, Texas, November 2016. Association for Computational Linguistics.

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *CVPR*, pages 4971–4980. IEEE Computer Society, 2018.

David Alvarez-Melis and Tommi Jaakkola. A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 412–421, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

Saeed Amizadeh, Hamid Palangi, Alex Polozov, Yichen Huang, and Kazuhito Koishida. Neuro-symbolic visual reasoning: Disentangling. In *International Conference on Machine Learning*, pages 279–290. PMLR, 2020a.

Saeed Amizadeh, Hamid Palangi, Oleksandr Polozov, Yichen Huang, and Kazuhito Koishida. Neuro-symbolic visual reasoning: Disentangling "visual" from "reasoning". In *Proceedings of the 37th International Conference on Machine Learning (ICML-2020)*, pages 10696–10707. 2020b.

Saeed Amizadeh, Hamid Palangi, Oleksandr Polozov, Yichen Huang, and Kazuhito Koishida. Neuro-symbolic visual reasoning: Disentangling "visual"

from "reasoning". In *Proceedings of the 37th International Conference on Machine Learning (ICML-2020)*, pages 10696–10707. 2020c.

Peter Anderson. Repository for Bottom-Up Top-Down VQA. `https://github.com/peteanderson80/bottom-up-attention`, 2018.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, pages 6077–6086. IEEE Computer Society, 2018.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015.

Leila Arras, Ahmed Osman, and Wojciech Samek. CLEVR-XAI: A benchmark dataset for the ground truth evaluation of neural network explanations. *Information Fusion*, 81:14–40, 2022.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018.

Amsterdam Blender Foundation, Blender Institute. Blender - a 3d modelling and rendering package. 2016.

Xiaojun Chang, Pengzhen Ren, Pengfei Xu, Zhihui Li, Xiaojiang Chen, and Alex Hauptmann. Scene graphs: A survey of generations and applications. *arXiv preprint arXiv:2104.01111*, 2021.

Chongyan Chen, Samreen Anjum, and Danna Gurari. Grounding answers for visual questions asked by visually impaired people. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19098–19107, June 2022.

Wenhu Chen, Zhe Gan, Linjie Li, Yu Cheng, William Wang, and Jingjing Liu. Meta module network for compositional visual reasoning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 655–664, 2021.

Kyunghyun Cho, B. V. Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

Corentin Dancette, Remi Cadene, Damien Teney, and Matthieu Cord. Beyond question-based biases: Assessing multimodal shortcut learning in visual question answering. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1554–1563, Los Alamitos, CA, USA, oct 2021. IEEE Computer Society.

Abhishek Das, Harsh Agrawal, C. Lawrence Zitnick, Devi Parikh, and Dhruv Batra. Human attention in visual question answering: Do humans and deep networks look at the same regions? In *EMNLP 2016*, 2016.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online, July 2020. Association for Computational Linguistics.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2:665–673, 11 2020.

Donald Geman, Stuart Geman, Neil Hallonquist, and Laurent Younes. Visual turing test for computer vision systems. *Proceedings of the National Academy of Sciences*, 112(12):3618–3623, 2015.

Tejas Gokhale, Pratyay Banerjee, Chitta Baral, and Yezhou Yang. Mutant: A training paradigm for out-of-distribution generalization in visual question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *EMNLP (1)*, pages 878–892. Association for Computational Linguistics, 2020.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017.

Vipul Gupta, Zhuowan Li, Adam Kortylewski, Chenyu Zhang, Yingwei Li, and Alan Loddon Yuille. Swapmix: Diagnosing and regularizing the over-reliance on visual context in visual question answering. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5068–5078, 2022.

Xinzhe Han, Shuhui Wang, Chi Su, Qingming Huang, and Qi Tian. Greedy gradient ensemble for robust visual question answering. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1564–1573, 2021.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *ICCV*, pages 804–813. IEEE Computer Society, 2017.

Ronghang Hu, Jacob Andreas, Trevor Darrell, and Kate Saenko. Explainable neural computation via stack neural module networks. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *ECCV (7)*, volume 11211 of *Lecture Notes in Computer Science*, pages 55–71. Springer, 2018.

Ronghang Hu, Anna Rohrbach, Trevor Darrell, and Kate Saenko. Language-conditioned graph networks for relational reasoning. In *ICCV*, pages 10293–10302. IEEE, 2019.

Drew A. Hudson and Christopher D. Manning. Compositional Attention Networks for Machine Reasoning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

Drew A. Hudson and Christopher D. Manning. GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6693–6702, 2019.

Drew A. Hudson and Christopher D. Manning. Learning by Abstraction: The Neural State Machine. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché Buc, Emily B. Fox, and Roman Garnett, editors, *NeurIPS*, pages 5901–5914, 2019.

Sarthak Jain and Byron C. Wallace. Attention is not Explanation. 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)*, pages 3543–3556, Minneapolis, Minnesota, June 2019.

Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik G. Learned-Miller, and Xinlei Chen. In defense of grid features for visual question answering. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10264–10273, 2020.

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

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1988–1997, 2016.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. Inferring and executing programs for visual reasoning. In *ICCV*, pages 3008–3017. IEEE Computer Society, 2017.

Kushal Kafle and Christopher Kanan. Visual question answering: Datasets, algorithms, and future challenges. *Computer Vision and Image Understanding*, 163:3–20, 2017.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 787–798, Doha, Qatar, October 2014. Association for Computational Linguistics.

Franklin Kenghagho Kenfack, F. A. Siddiky, Ferenc Bálint-Benczédi, and Michael Beetz. RobotVQA — A Scene-Graph- and Deep-Learning-based Visual Question Answering System for Robot Manipulation. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9667–9674, 2020.

Corentin Kervadec. *Bias and reasoning in Visual Question Answering*. Phd thesis, Université de Lyon, December 2021.

Corentin Kervadec, Grigory Antipov, Moez Baccouche, and Christian Wolf. Roses are Red, Violets are Blue... But Should VQA expect Them To? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2776–2785, 2021.

Eun-Sol Kim, Woo-Young Kang, Kyoung-Woon On, Yu-Jung Heo, and Byoung-Tak Zhang. Hypergraph attention networks for multimodal learning. In *CVPR*, pages 14569–14578. IEEE, 2020.

Jin-Hwa Kim, Kyoung-Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Hadamard product for low-rank bilinear pooling. In *International Conference on Learning Representations*, 2016.

Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *NeurIPS*, pages 1571–1581, 2018.

Wonjae Kim, Bokyung Son, and Ildoo Kim. ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision. In *International Conference on Machine Learning*, 2021.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016.

Guohao Li, Xin Wang, and Wenwu Zhu. Perceptual visual reasoning with knowledge propagation. In Laurent Amsaleg, Benoit Huet, Martha A. Larson, Guillaume Gravier, Hayley Hung, Chong-Wah Ngo, and Wei Tsang Ooi, editors, *ACM Multimedia*, pages 530–538. ACM, 2019a.

Jiwei Li, Will Monroe, and Dan Jurafsky. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*, 2016.

Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. Relation-aware graph attention network for visual question answering. In *ICCV*, pages 10312–10321. IEEE, 2019b.

Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, 2020.

Weixin Liang, Yanhao Jiang, and Zixuan Liu. GraghVQA: Language-guided graph neural networks for graph-based visual question answering. In *Proceedings of the Third Workshop on Multimodal Artificial Intelligence*, pages 79–86, Mexico City, Mexico, June 2021. Association for Computational Linguistics.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

Tsung-Yi Lin, P. Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature Pyramid Networks for Object Detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017.

Zheyuan Liu, Cristian Rodriguez-Opazo, Damien Teney, and Stephen Gould. Image retrieval on real-life images with pre-trained vision-and-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2125–2134, October 2021.

Andrej Ljolje, Fernando Pereira, and Michael Riley. Efficient general lattice generation and rescoring. In *Sixth European Conference on Speech Communication and Technology*, 1999.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *NeurIPS*, pages 13–23, 2019.

Mateusz Malinowski, Marcus Rohrbach, and Mario Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

Varun Manjunatha, Nirat Saini, and Larry S. Davis. Explicit bias discovery in visual question answering models. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9554–9563, Los Alamitos, CA, USA, jun 2019. IEEE Computer Society.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

David Mascharka, Philip Tran, Ryan Soklaski, and Arjun Majumdar. Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In *CVPR*, pages 4942–4950. IEEE Computer Society, 2018.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*, 2013.

John Stuart Mill. *A System of Logic, Ratiocinative and Inductive, Being a Connected View of the Principles of Evidence, and the Methods of Scientific Investigation, 3rd Ed., Vol. 1, Chap. 2*. London: John W. Parker, 1851, 1851.

Timothy Niven and Hung-Yu Kao. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy, July 2019.

Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. Multimodal explanations: Justifying decisions and pointing to the evidence. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8779–8788, 2018.

Badri N. Patro, Shivansh Pate, and Vinay P. Namboodiri. Robust explanations for visual question answering. *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1566–1575, 2020.

Jeffrey Pennington, R. Socher, and Christopher D. Manning. Glove: Global Vectors for Word Representation. In *EMNLP*, 2014.

Oskar. Pfungst and Carl Leo. Rahn. *Clever Hans (the horse of Mr. Von Osten) a contribution to experimental animal and human psychology.* New York, H. Holt and company, 1911, 1911.

Arijit Ray, Karan Sikka, Ajay Divakaran, Stefan Lee, and Giedrius Burachas. Sunny and dark outside?! improving answer consistency in VQA through entailed question generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5860–5865, Hong Kong, China, November 2019.

Daniel Reich and Tanja Schultz. Uncovering the full potential of visual grounding methods in VQA. In *The 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Daniel Reich, Felix Putze, and Tanja Schultz. Visually Grounded VQA by Lattice-based Retrieval. *arXiv preprint arXiv:2211.08086*, 2022.

Daniel Reich, Felix Putze, and Tanja Schultz. Measuring faithful and plausible visual grounding in VQA. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 3129–3144. Association for Computational Linguistics, 2023.

Shaoqing Ren, Kaiming He, Ross B. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015.

Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 2662–2670. AAAI Press, 2017.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. On causal and anticausal learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, page 459–466, 2012.

Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In Regina Barzilay and Min-Yen Kan, editors, *ACL (1)*, pages 1073–1083. Association for Computational Linguistics, 2017.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

Ramprasaath R. Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Dhruv Batra, and Devi Parikh. Taking a hint: Leveraging explanations to make vision and language models more grounded. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2591–2600, 2019.

Ramprasaath R. Selvaraju, Purva Tendulkar, Devi Parikh, Eric Horvitz, Marco Tulio Ribeiro, Besmira Nushi, and Ece Kamar. SQuINTing at VQA Models: Introspecting VQA Models With Sub-Questions. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10000–10008, 2020.

Nambirajan Seshadri and C-EW Sundberg. List Viterbi Decoding Algorithms with Applications. *IEEE transactions on communications*, 42(234):313–323, 1994.

Jiaxin Shi, Hanwang Zhang, and Juanzi Li. Explainable and explicit visual reasoning over scene graphs. In *CVPR*, pages 8376–8384. Computer Vision Foundation / IEEE, 2019.

Robik Shrestha, Kushal Kafle, and Christopher Kanan. A negative case analysis of visual grounding methods for VQA. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8172–8181. Association for Computational Linguistics, July 2020.

Hao Tan and Mohit Bansal. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *EMNLP/IJCNLP (1)*, pages 5099–5110. Association for Computational Linguistics, 2019.

Damien Teney, Kushal Kafle, Robik Shrestha, Ehsan Abbasnejad, Christopher Kanan, and Anton van den Hengel. On the value of out-of-distribution testing: An example of goodhart's law. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *CVPR*, pages 1521–1528. IEEE Computer Society, 2011.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany, August 2016. Association for Computational Linguistics.

Aisha Urooj. Implementation for the paper "Found a Reason for me? Weakly-supervised Grounded Visual Question Answering using Capsules" (Aisha Urooj, et al., CVPR 2021). `https://github.com/aurooj/WeakGroundedVQA_Capsules`, 2021.

Aisha Urooj, Hilde Kuehne, Kevin Duarte, Chuang Gan, Niels Lobo, and Mubarak Shah. Found a reason for me? weakly-supervised grounded visual question answering using capsules. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8465–8474, June 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China, November 2019. Association for Computational Linguistics.

Jialin Wu and Raymond J. Mooney. Self-critical reasoning for robust visual question answering. In *NeurIPS*, 2019.

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 670–685, 2018.

Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alexander J. Smola. Stacked attention networks for image question answering. In *CVPR*, pages 21–29. IEEE Computer Society, 2016.

Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In *NeurIPS*, pages 1039–1050, 2018.

Zhuofan Ying, Peter Hase, and Mohit Bansal. VisFIS: Visual Feature Importance Supervision Right-for-the-Right-Reason Objectives. In *NeurIPS*, 2022.

Zhou Yu, Yuhao Cui, Zhenwei Shao, Pengbing Gao, and Jun Yu. OpenVQA. `https://github.com/MILVLG/openvqa`, 2019a.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6274–6283, 2019b.

Yuanyuan Yuan, Shuai Wang, Mingyue Jiang, and Tsong Yueh Chen. Perception Matters: Detecting Perception Failures of VQA Models Using Metamorphic Testing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16908–16917, June 2021.

Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Yin and yang: Balancing and answering binary visual questions. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5014–5022, 2016.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5579–5588, June 2021.

Zelin Zhao, Karan Samel, Binghong Chen, et al. Proto: Program-guided transformer for program-guided tasks. *Advances in Neural Information Processing Systems*, 34:17021–17036, 2021.

C. Lawrence Zitnick and Devi Parikh. Bringing semantics into focus using visual abstraction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3009–3016, 2013.

Appendix A

Scene Graph Generation

In this Appendix, we report additional details for the Scene Graph Generator (SGG) that was developed as part of VLR (Chapter 3). SGG provides image representations for the vast majority of experiments performed throughout this thesis.

A.1 SGG1: Object detection and visual feature extraction

SGG1 handles object detection and provides region-based visual features that are used as basis for modeling attribute and relationship detection, as well as for training other reference VQA models. We use a Faster R-CNN (Ren et al., 2015) model with ResNet101 (He et al., 2016) backbone and an FPN (Lin et al., 2017) for region proposals. The model is built using Facebook's Detectron2 framework (Wu et al., 2019). The ResNet101 backbone model was pre-trained on ImageNet (Deng et al., 2009).

Training details The model was trained for GQA's 1702 object classes using 75k training images (images in GQA's balanced train partition). Training lasted for 1m iterations with mini-batch of 4 images, using a multi-step learning rate starting at 0.005, reducing it by a factor of 10 at 700k and again at 900k iterations. No other parameters were changed in the official

System	mAP @[.5 : .95]	mAP @0.5	mAP @0.75	mAP small/med/large
SGG1	5.54	9.45	5.76	3.32 / 6.13 / 9.92
UpDn	n/a	10.2	n/a	n/a

Table A.1 – Object detection performances with Faster R-CNN models using MS COCO evaluation metrics.

Detectron2-provided training recipe for this model architecture. Training took about 7 days on an RTX 2080 Ti.

Output We use the softmax output distribution (a 1702-dim vector for 1702 object classes) for each post-NMS[1] detected object to populate a scene graph. Up to 100 objects per image are selected as follows: per-class NMS is applied at 0.7 IoU for objects that have any softmax object class probability of > 0.05.

We also use this model to extract 1024-dim object-based sub-symbolic visual features, which we use in SGG2&3 (attribute recognition, relationship detection models) and for training other VQA reference models which are used in certain comparisons in this thesis. These sub-symbolic visual features are extracted from a layer in the object classification head of the Faster R-CNN model which acts as input to the final fully-connected softmax-activated output layer. This is done for each surviving object (i.e., for the set of objects determined by per-class NMS and top-100 capping).

Results SGG1's object detection evaluation using metrics[2] defined for the MS COCO dataset (Lin et al., 2014) are shown in Table A.1. We also include one more data point in Table A.1 to share a very rough comparison to UpDn's (Anderson et al., 2018) object detection model that was widely used in other VQA-related works to extract object-based sub-symbolic visual features. UpDn's object detector was trained on a heavily cleaned subset of Visual Genome (Krishna et al., 2016) (on which GQA is based) and uses 1600 object classes. Note that evaluation results for this object detector are not published in their paper but instead listed in UpDn's official repository (Anderson, 2018).

[1]Non Maximum Suppression (NMS) is a widely used technique for meaningfully selecting one detected bounding boxes from among an often impractically large set of overlapping bounding boxes, which are deemed duplicate detections of the same object.

[2]https://cocodataset.org/#detection-eval

Note on object categories in VLR Object categories take on a central role in the GQA dataset. Many questions refer to objects by their category instead of their object class identity (e.g. "Is someone standing next to the red piece of *furniture*?"). Category recognition scores are used in VLR's inference procedure in Chapter 3. To determine category recognition scores for detected objects, we simply sum softmax scores of all object classes that belong to the requested category. Like for attribute recognition (see SGG2 below), we determine the categories, as well as the members (object class names) in each category, based on implicit declarations in GQA. Note that there is no explicit definition of the used categories and their members in GQA given as part of the official annotations, but this information can be inferred by appropriate processing of the QA annotation files.

A.2 SGG2: Attribute recognition

We identified a total of 617 individual attribute names and 39 overarching attribute categories in GQA, including a category containing all unassigned attributes ("others" in Table A.2). We train a separate softmax regression model for each of these 39 attribute categories, using the sub-symbolic visual features from SGG1 as input features. Classifier sizes range from a maximum of 427 classes ("other" category) to a minimum of 2 classes (e.g., "weight", "height"), with most classifiers covering 3 or less classes. Categories and category membership were determined based on their declaration in GQA.

Training details Input to each category model is a detected object's 1024-dim sub-symbolic visual feature vector, extracted with SGG1. For training and evaluation, the detected objects are labeled with attributes as follows: We first determine, if a SGG1-detected bounding box matches an (attribute-)annotated bounding box in GQA. If yes, we assign any annotated attribute labels belonging to the ground-truth bounding box to that detected object. A detected bounding box has found a match if it exceeds an IoU of 0.75 with a ground-truth bounding box. The labeled samples are then used to train and evaluate every category model they have labels for.

We train all models using the Adam optimizer with a learning rate of 0.001 and apply L2 regularization to avoid overfitting. As loss function we use a common cross-entropy loss. Models are trained using early stopping with patience of 5 epochs.

Output Attributes for detected objects in the scene graph are created as follows: we feed a detected object's 1024-dim visual feature vector into each of the 39 models and extract the softmax activation output distribution over each category's classes. The resulting 39 distributions then represent the complete attribute information of an object in the image's scene graph.

Results Results for all 39 category models, as well as sample sizes of train, dev and test sets are listed in Table A.2.

Note on attribute categories in VLR Like object categories, attribute categories are used extensively in GQA. In VLR, categories that are mentioned in the question (and/or later in the functional program which is generated by the question parser) can, e.g., help narrow down answer options for a given question. A question such as "What *color* is the chair?" causes VLR's Answer Production sub-module to select an answer from classes (attribute names) belonging to the "color" category. Like for object categories, attribute categories and category membership are determined based on implicit declarations in GQA. Note that a few attributes are part of multiple categories (e.g. the attribute "little" is a class in categories "size" and "age"), which is why the total number of classes in all trained attribute models is slightly higher than the number of unique attribute names.

A.3 SGG3: Visual relationship detection

We identified 310 relationship names in GQA (e.g., "wearing", "holding"), including 17+2 spatial relationships (e.g., "behind"). Due to the overwhelming frequency of the two spatial relationship classes "to the left|right of" in the annotations (see their counts in Table A.4), we place them in a separate category. Hence, all relationship names are split into three categories (spatial types, left-right and others).

Relationships are a directed connection between two objects, i.e., they do not have the commutative property (e.g., "man wearing shirt" is not the same as "shirt wearing man"). We therefore frame relationship detection as a sequence classification problem.
Using sub-symbolic visual features and bounding box coordinates of detected objects from SGG1, we train an LSTM model with a softmax output layer for each of the three categories. We additionally train a binary classifier for each category that learns to predict whether or not a given object pair has *any* relationship in the respective category. Input features for these binary

Attribute Category	Classes	Samples train,dev,test	Chance (=Prior)	Accuracy
activity	15	11205,1244,1787	0.30	0.61
age	3	11610,1290,1754	0.46	0.75
brightness	2	11062,1229,1737	0.85	0.90
cleanliness	4	3887,431,596	0.58	0.70
color	26	456607,50734,71627	0.24	0.53
company	2	522,57,80	0.64	0.57
depth	2	420,46,64	0.62	0.69
event	3	143,15,20	0.60	0.70
face	4	3919,435,608	0.97	0.98
fatness	3	2078,230,327	0.57	0.53
flavor	4	1257,139,237	0.71	0.83
gender	2	887,98,154	0.68	0.76
hardness	2	562,62,90	0.54	0.80
height	2	16250,1805,2559	0.73	0.90
length	2	12920,1435,2041	0.67	0.84
liquid	5	608,67,94	0.53	0.67
location	2	579,64,71	0.82	0.86
material	41	67030,7447,10778	0.30	0.65
opaqness	2	7613,845,1076	0.99	1.00
orientation	2	393,43,68	0.53	0.65
others	427	141885,15765,22644	0.05	0.37
pattern	3	5085,564,893	0.79	0.79
place	5	647,71,101	0.31	0.75
pose	9	31410,3489,4774	0.38	0.64
race	2	656,72,96	0.62	0.62
realism	2	381,42,70	0.61	0.59
room	3	482,53,72	0.43	0.50
shape	5	12031,1336,1749	0.63	0.74
size	6	50809,5645,7779	0.54	0.72
sport	4	6705,745,955	0.61	0.97
sportActivity	8	8325,925,1233	0.28	0.78
state	6	3393,377,540	0.46	0.64
texture	2	19,2,9	0.78	0.56
thickness	2	2843,315,432	0.56	0.68
tone	2	10885,1209,1733	0.85	0.86
type	3	6442,715,917	0.64	0.97
weather	9	11477,1275,1691	0.63	0.76
weight	2	1980,219,307	0.85	0.84
width	2	1156,128,219	0.74	0.76
Avg (std)	-	-	0.58 (0.14)	0.73 (0.14)
Weighted avg (std)	-	-	0.31 (0.21)	0.59 (0.20)

Table A.2 – Attribute recognition with softmax regression models, results sorted by category name. One model per category. See text (A.2) for details.

models consist of GloVe word embedding vectors representing the detected objects' class names.

Training details For training and evaluation, detected object pairs from SGG1 are labeled with relationships as follows: We first determine, if an SGG1-detected bounding box matches a (relationship-)annotated bounding box in GQA. If yes, we assign any annotated relationship labels belonging to the ground-truth bounding box to that detected object, but *only* if the target object in that relationship has also been matched by another detected object. A detected bounding box matches with a ground-truth bounding box if it exceeds an IoU of 0.75 with it. A labeled relationship (i.e. an ordered pair of objects) is used to train and evaluate the category model that contains the labeled relationship class.
For *recognition* of relationships in the three categories, we train an LSTM, each with 512 hidden units followed by a dropout layer (drop rate of 0.3) and a softmax output layer. Input to the LSTMs is a sequence consisting of two object's 1028-dim vectors (1024-dim visual features, 4-dim bounding box coordinates, taken from SGG1). The vectors are ordered according to the directed relationship of the involved objects (i.e., relationship subject→object).
For *detection* of presence of a relationship given an ordered pair of objects (subject, object), we train LSTM models, which are architecturally similar to the recognition models, but use a sigmoid activation in the output layer instead of a softmax activation. We train one model per relationship category. In contrast to the recognition models, the input features consist of 100-dim GloVe word embedding vectors representing the SGG1-detected 1-best object name (i.e., the maximum softmax entry in the 1702-dim object class distribution). The vectors are appended by the respective 4-dim bounding box coordinates. The intuition behind using word embeddings instead of visual features is that language-based semantics of relationships between objects can provide a richer basis to model prior probabilities of relationships between objects than the observed visual features.

Note that to train/evaluate the binary classifiers, we also need samples of the negative class, i.e., examples of object pairs that are not in the relationship category of the positive class. We select these negative samples from among object pairs that only have relationships in other categories but not in the category in question.

Output Similar to SGG1&2, we use a model's softmax output to define a probability distribution over relationship classes between two objects. Note that a probability distribution is only generated with *recognition* models if

Relationship Category	Class	Samples test	Prec	Rec	F1
Spatial	pos	10.5k	0.40	0.85	0.55
	neg	139k	0.99	0.90	0.94
Spatial (left|right)	pos	134k	0.99	0.97	0.98
	neg	15k	0.81	0.94	0.87
Other	pos	9k	0.44	0.90	0.59
	neg	141k	0.99	0.93	0.96

Table A.3 – Results of relationship *detection* models per category. These binary models determine whether or not an ordered pair of objects has any relationship in that category. We list Precision, Recall and F1-score for a positively/negatively classified relationship detection in a category.

Relationship Category	Classes	Samples train,dev,test	Chance (=Prior)	Acc
Spatial	17	69k, 8k, 11k	0.27	0.55
Spatial (left|right)	2	861k,96k,134k	0.5	0.997
Other	297	55k, 6k, 9k	0.40	0.77

Table A.4 – Results of relationship *recognition* models per category. These models assume there is a relationship between two objects (given as an ordered pair) in the respective category and determine which one it is.

the respective *detection* model outputs the positive class for a given ordered object pair, thus predicting the existence of a relationship between the two objects. Otherwise, we set all values for that relationship category to zero in the scene graph representation.

Results Results for recognition models are shown in Table A.4, results for detection models are shown in Table A.3.

A.4 Scene Graph Representation in VLR

We represent the scene graph internally as a collection of matrices to improve computational efficiency in VLR (Chapter 3) as follows:

Nodes All objects in the scene graph are represented in a matrix of dimensions (#obj_in_img, #obj_classes). Object attributes are represented in a matrix of dimensions (#obj_in_img, #attr_categories, #max_attr_class_in_cat). Each object class (1702 total) and attribute class (617 total across 39 categories) receives a softmax score from their respective classifier, which is then stored in these matrices.

Edges All relationships are represented in a matrix of dimensions (#obj_in_img, #obj_in_img, #rel_classes). Each object in the image can be in a relationship (310 total across 3 categories) with another object, but not itself. Each relationship class receives a softmax score from the classifiers described in A.3.

Appendix B

VLR Implementation and Experiment Details

This Appendix contains complementary information to Chapter 3.

B.1 Question Parser

In this section, we give some additional details of the model setup and training process for VLR's Question Parser, which is described in Chapter 3.3.1.

Pre- and Post-processing

For training the QP, we *pre*-process the operation sequences in a separate step, which in particular involves splitting each operation tuple into multiple tokens. For instance, the operation tuple "(relate; cat,next to,s)" would be represented as a sequence of four tokens. A *post*-processing step of the QP output is then required to revert the QP's output sequence into the original tuple format. Subsequent modules (namely VLR's R&A in Chapter 3.3.3) process the operation sequence (i.e., the program) in the original GQA tuple-based format. Note that there are instances where the pre-/post-processing steps introduce issues to the final QP output, which can cause problems during construction of the VQA-lattice. To quantify this, we isolate and report the impact of this step in our ablation study (labeled "VLR*" in Table B.2).

Model specifications and Training

We use 50-dim, pre-trained GloVe word embedding vectors to encode words in the question. As recurrent layers in the encoder/decoder, we use a GRU (Cho et al., 2014) with 128 units. Question length is limited to 20 tokens which matches 99.5% of our training data. The total question counts before removal are 849k (train), 94k (dev) and 132k (test). Output sequences in the test set are 7.4 tokens long on average. The softmax output layer of the model consists of 162 classes. Here, 20 classes are reserved for pointers to input question words, 1 class to signal empty/no operation, 93 classes are used for operation terms that occurred > 100 times in training (without thresholding, there were 136 terms). The remainder of classes is used for explicit modeling of the most frequently occurring (multi-) words in operation arguments, as well as non-word functional tokens like underscore, which never appear in the input question and thus cannot be copied from it. We use a) regular expressions and b) GQA's annotated pointers (from question words to operation arguments) to determine whether or not a token in the target sequence should be a pointer to a certain question word in the input sequence at train time.

Results

The QP reaches 97.02% element accuracy and 79.60% full program accuracy on the test set (GQA balanced val set: 132k questions with average output sequence length of 7.4 tokens per question program).

B.2 Rank & Answer

In this section, we give some additional details about the inner workings of VLR's lattice construction process described in Chapter 3.3.3.

Lattice Construction

GQA uses 12 unique, fundamental operation types (e.g., "filter", "select"; note: not to be confused with complex operation types like "filter color", etc), of which 9 (such as "query", "and") are used in VLR for determining the final operations in the answer production module that produces the answer to the question (see also Chapter 3.3.3, "Answer" paragraph).

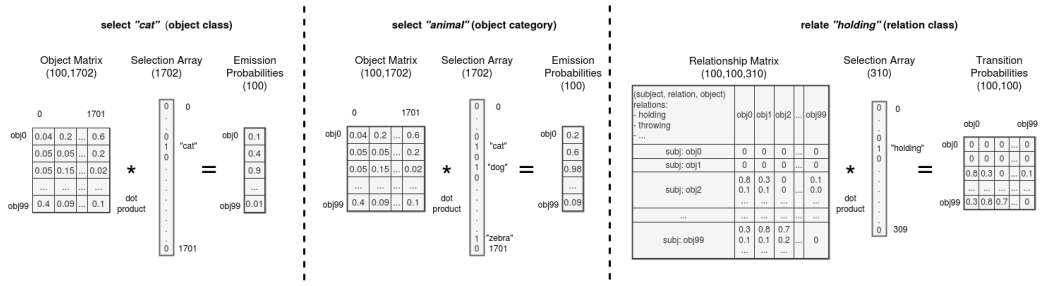

Figure B.1 – Illustration of elementary operations used in the construction of the VQA-lattice. Matrix operations are used to extract node emission and transition probabilities, which are subsequently used in the VQA-lattice. For additional description see B.2. Depicted numbers were randomly chosen.

Elementary Operations

VLR internally maps all operation types to (combinations of) two elementary query operations: "select" and "relate". These elementary operations are essentially queries to the scene graph representation matrices (described in Appendix A.4) and extract 1) node emission and 2) node transition probabilities. These are then used in the VQA-lattice for a given question and image. An illustration of how these elementary operations work is shown in Figure B.1 and described below.

Select. An operation sequence (representing the parsed question) consists of at least two operation tuples. Each tuple consists of an operation type and an argument value (also described in Chapter 3.3.1 and 3.3.3, "Lattice Construction" paragraph). To execute an elementary operation, we first use the argument value to create a "Selection Array". In Figure B.1, left, the argument value is "cat" which results in the creation of a one-hot "Selection Array" that is 1 at the object class position for "cat" and 0 everywhere else. Taking the dot product of this one-hot vector and the object matrix then results in the node emission probabilities for all 100 detected objects in the image. The middle image in Figure B.1 illustrates the same process for arguments that query object categories (which encompass multiple object classes). Queries about attributes are done in a similar fashion and also result in node emission probabilities for each detected object.

Relate. The second elementary operation, "relate" (Figure B.1, right) extracts transition probabilities from the scene graph to populate edges between nodes in the VQA-lattice as a hollow matrix (square matrix with

diagonal entries all equal to zero), representing the fact that each node in the lattice can theoretically transition to any other node except itself.

Note that if node emissions are extracted by queries to the Attribute Matrix, we always insert the identity matrix to represent the transition probabilities of this step (Viterbi requires transition probabilities in each step), as no actual transition to other nodes is happening when attributes are queried.

Limitation

VLR uses word-based symbolic representations for classes and categories. This means, in particular, that class names queried by operations need to match class names used in the SGG module, in order to get a correct match (or even a match at all). This strict requirement can be relaxed by introducing a normalization step before the matching procedure to align names in the query with the names used in the scene graph. We apply some normalization (lemmatization of plural/singular forms of object names), which treats most mismatches encountered with VLR in GQA. Although other mismatches are rare in GQA ($< 1\%$ of questions), it might be helpful to include a normalization step in other scenarios.

B.3 Experiments

B.3.1 VLR Ablation Study

The modular nature of VLR allows for an in-depth ablation-type evaluation. We list a number of module combinations in Table B.2, which we discuss below.

Preliminary Note on Scene Graph Variants

Rows in the ablation Table B.2 can be interpreted as representing the varying degrees of involvement of GQA ground-truth annotations in the ablation (roughly: the higher the variant number, the larger the reliance on annotations). For VLR's ablation study (Table B.2) we create four scene graph variants, populated to varying degrees by entries from automatic detections vs. ground-truth annotations. For easier reference, we list these four scene graph variants in Table B.1.

Scene Graph Variant	VLR-detected	GQA-annotated
VLR, VLR-1, VLR-2	obj, attr, rel	-
VLR-3, VLR-4	obj, attr	rel
VLR-5, VLR-6	obj	attr, rel
VLR-7, VLR-8, VLR-Oracle	-	obj, attr, rel

Table B.1 – Additional information on the scene graph variants used in the VLR ablation experiments in Table B.2, see B.3.1 for details.

System	QP	Scene Graph	Binary	Open	Grounding	Acc
VLR	VLR	VLR (obj,attr,rel)	69.94	46.17	128.41	57.67
VLR-1	VLR*	VLR (obj,attr,rel)	72.35	47.52	130.45	59.53
VLR-2	GQA	VLR (obj,attr,rel)	73.58	47.59	132.00	60.16
VLR-3	VLR	VLR (obj,attr); GQA (rel)	70.11	50.88	137.67	60.18
VLR-4	GQA	VLR (obj,attr); GQA (rel)	75.18	53.40	141.19	63.93
VLR-5	VLR	VLR (obj); GQA (attr,rel)	76.86	56.52	138.84	66.36
VLR-6	GQA	VLR (obj); GQA (attr,rel)	82.99	59.65	142.23	70.95
VLR-7	VLR	GQA (obj,attr,rel)	84.06	75.97	149.82	79.88
VLR-8	VLR*	GQA (obj,attr,rel)	89.46	79.88	152.89	84.52
VLR-Or	GQA	GQA (obj,attr,rel)	96.47	87.38	162.73	91.78

Table B.2 – Ablation study for VLR. Shows VLR's performance for various combinations of using annotated (=Oracle) and predicted scene graphs and operation sequences. "GQA" entries stand for Oracle inputs from GQA annotations. "VLR*" is defined in B.1. Here, we skip the learned QP but still go through pre- and post-processing (see B.1) which introduces some errors. "VLR-Or" represents VLR using full Oracle input (=GQA annotations), which acts as the upper bound of VLR.

Ablation: Rank & Answer (R&A)

The upper-bound performance of VLR when running in full Oracle mode (GQA-annotated scene graph and programs) is 91.78% accuracy (TB.2, VLR-Or). Errors occur, e.g., due to 1) issues in annotations, or 2) when processing uncommon operation tuples that are not explicitly handled in R&A's lattice construction procedures or answer production.

Ablation: Question Parser (QP)

Using the GQA scene graph and QP-generated programs results in a steep drop in accuracy to 79.88% or 13% relative reduction (VLR-Or →VLR-7). Closer analysis reveals in particular problems w.r.t. inaccurate program

pointers input question words that cannot be correctly resolved, as well as general classification errors of the QP. Comparing VLR programs with GQA ground-truth programs, executed on a fully VLR-detected scene graph (VLR-2 →VLR), the mentioned accuracy drop shrinks significantly to 4% relative, suggesting a certain overlap between challenging visual scenes and questions that are more difficult to process.

Ablation: Object Detection (SGG1)

Object detection is by far the most influential component in terms of accuracy impact. If an *undetected* object is queried somewhere in the question, VLR's ability to arrive at the correct answer is heavily impaired.

VLR's SGG1 detects on average 91.49% of all objects in GQA's annotated inference chain, and 92.34% of objects needed to answer a question (detection determined for $IoU > 0.5$). This means that a significant number of questions cannot be reasonably answered correctly on account of critical objects missing in the scene graph. All numbers are listed in Chapter 3.4.3, Table 3.2, Line 1.

Aside from object *detection* issues, correct object classification (or *recognition*) for a large variety of objects such as that found in GQA (1702 object classes) is clearly a challenge (cf. A.1 for object recognition results).
To evaluate the impact of object recognition quality on VLR's overall accuracy, we create a scene graph variant that consists of SGG's detected object locations and recognized object classes, but uses all annotated attributes and relationships from matching ground-truth objects ("matching": bounding box with highest IoU > 0.5). As expected, this heavily impacts VLR's overall accuracy which falls from 91.78% to 70.95% (-23% relative) for Oracle programs (VLR-Or →VLR-6), and from 79.88% to 66.36% (-17% relative) for VLR-generated programs (VLR-7 →VLR-5).

Ablation: Attribute recognition (SGG2)

Similar to object detection, attribute recognition is important for identifying objects referenced in an operation sequence. We replace the ground-truth attributes in the partial Oracle scene graph from VLR-6&5 with detected outputs from the SGG2 module (i.e., the attribute models). Although the impact of this change on VLR's accuracy is not as strong as when introducing detected objects, we still observe a large drop from 70.95% to 63.93% (VLR-6 →VLR-4), and from 66.36% to 60.18% (VLR-5 →VLR-3).

Ablation: Relationship detection (SGG3)

Finally, we integrate outputs from SGG3 (relationship models) into the scene graph to arrive at a fully VLR-generated scene graph without annotation involvement. There is a smaller sized accuracy reduction compared to the introduction of attribute recognitions: We observe performance drops from 63.93% to 60.16% (VLR-4 →VLR-2) for Oracle programs, and 60.18% to 57.67% (VLR-3 →VLR) to reach VLR's final overall accuracy without any annotation involvement in either input modality.

B.3.2 Model Training

In this section we include details for training procedures of models used in evaluations and comparisons in Chapter 3. In general, all models use GQA's balanced train set for training and the balanced val set for testing. A small dev set (either some part of the train set or separately provided in case of experiments on GQA-101k (Ying et al., 2022)) is used for model selection. Note that with exception of GQA-101k's test sets (which mix questions from balanced train and val sets), all images that are used in testing are unseen during training of our visual perception module (=SGG). All trained models use our same 1024-dim object-based visual features (for 100 objects/image max) as input.

MMN

MMN (Chen et al., 2021) consists of two main modules that are trained separately: A program parser and the actual inference model, which takes the predicted program from the parser as input. We mostly follow the settings in the official code-base but detail some aspects of our customization here.

For the program parser, we run training for 20 epochs (official setting: 10 epochs) and choose the best model (lowest loss on dev set). For the inference model, we run up to 15 epochs of bootstrapping (using the balanced train set) with Oracle programs and another up to 12 epochs of fine-tuning with parser-generated programs. We use early stopping of 1 epoch and select the model by best accuracy on the dev set (using Oracle programs in bootstrapping mode and predicted programs in fine-tuning mode).

Training on generalization/OOD splits is done accordingly. Notably, the program parser was always retrained on each new split (same as for VLR).

DFOL

DFOL (Amizadeh et al., 2020b) uses a vanilla seq2seq program parser, but neither code nor generated output for this is provided in the official code base. Thus, evaluations are run with ground-truth programs from GQA. DFOL is trained on a loss based on answer performance to learn weights in its visual encoding layers that produce an image representation similar to the one used by VLR, given high-dimensional visual features as input.

Training is done based on the official instructions for a complex 5-step curriculum training procedure. We train the first 4 curriculum steps with the entire ∼14 million questions in the "all" training data partition, as specified in the instructions. As this is extremely resource intensive, we train for one epoch in each curriculum step. Finally, we run the 5th step with the "balanced" train data only (∼1m questions) for several epochs until training finishes by early stopping of 1 epoch.

Note that due to missing code for the program parser, DFOL cannot be reasonably evaluated in our generalization/OOD experiments, as the program parser's performance is crucial to a realistic evaluation for this model (like in VLR).

MAC

MAC (Hudson and Manning, 2018) is a monolithic VQA model based on a recurrent DNN architecture. The model takes a constant number of inference steps per question for its interaction with the visual knowledge base. We follow the official training guidelines given in the official code base and use 4-step inference. We train the model on GQA's balanced train set and use early stopping of 1 epoch based on accuracy on a dev set to select the best model. Training on generalization/OOD splits is done accordingly.

UpDn

UpDn (Anderson et al., 2018) is a classic attention-based model with a single attention step guiding the merge of vision and language modalities. We use the implementation shared by (Shrestha et al., 2020). Following the instructions there, we train UpDn for 40 epochs and select the best model based on accuracy on a dev set. Training on generalization/OOD splits is done accordingly.

B.4 Dataset Creation for Generalization Experiments

Three of four generalization splits used in Chapter 3.4.4 are straightforward to create based on the given description. However, the creation of the fourth split, "linguistic variants", is significantly more involved. For reproducibility, we describe in detail how we created the data partitions for this experiment. We tried to replicate the "structural" generalization experiment from Hudson and Manning (2019), but found this to be infeasible with the short description given in that paper. Hence, we developed a process for re-partitioning the train/test data for our experiment.

B.4.1 Step 1: Determining question sets for each linguistic variant

As starting point, we use the four explicitly mentioned linguistic variant pairs mentioned for the "structural" data split in Hudson and Manning (2019). Concrete examples are listed in Table B.3. We first determine all questions that belong to each linguistic variant for each of the four example pairs (training vs. test) listed in Table B.3 as follows:

1. Pair 1: Based on a question's (GQA-annotated) functional program, we determine all questions containing relationship names in passive form ("cover-ed") vs. present participle form ("cover-ing").
2. Pair 2: We determine all questions starting with "Do/Does" vs. "Is/Are".
3. Pair 3: We determine all questions containing an attribute category name (like "material", "shape", etc.) vs. no such term.
4. Pair 4: We determine all questions containing the word "called" vs. "name of".

At this point, we have two sets of questions for each of the four variant pairs from the table. Note, this procedure is performed for the GQA balanced train set and val set separately.

B.4.2 Step 2: Using program templates to determine equivalent inference

The GQA-annotated functional programs can be generalized as templates by using placeholders in the program's arguments (e.g., for objects, attributes,

training	test
What is the OBJ *covered* by?	What is *covering* the OBJ?
Is there a OBJ in the image?	*Do* you see any OBJ in the photo?
What is the OBJ made of?	What *material* makes up the OBJ?
What's the *name of* the OBJ that is ATTR?	What is the ATTR OBJ *called*?

Table B.3 – Linguistic variant pairs used for re-partitioning the train/test set. For each pair (=row) we list examples of each of the two linguistic variants (=column). Questions belonging to a linguistic variant will be either in the new train or test partition.

relationships). For instance, "(select: car), (verify color: red)" can be generalized to "(select: OBJ), (verify color: ATTR)". We use these templates for generalized matching of inference chains between questions. To identify questions with equivalent inference programs, we now determine those templates that occur for questions in both variants of a variant pair from step 1. Note that we only select templates that occurred at least 100 times for each variant of a given variant pair. We found this threshold to be necessary because of incorrect program annotations in GQA that did not match the question. Without the threshold, this issue was compromising the experimental setup by causing the selection of large amounts of questions that did not actually have an equivalent inference match in the other variant group.

B.4.3 Step 3: Selecting questions for the new partitions

After having determined the set of qualifying inference templates for each of the variant pairs, we now create the new train and test partitions. Note that no questions are moved between GQA's balanced train set and val set. Instead, the new partitions are created only by *removing* questions from a set. This retains GQA's integrity of keeping only unseen *images* in the test partition. In particular, the images in test are unseen in object detector training of our SGG).

We first select questions for each of the four linguistic variant pairs and then combine all selections to create the final train/test partitions.

Train set re-partitioning For re-partitioning GQA's balanced train set, we select all questions with matching inference templates from the "test"-labeled variant subsets in Table B.3 and remove them from the balanced train set.

Test set re-partitioning For re-partitioning GQA's balanced val set, we only select questions with matching inference templates from the "test"-labeled variant subsets in Table B.3.

Appendix C

FPVG Implementation and Experiment Details

This Appendix contains complementary information to Chapter 4 and Chapter 6.

C.1 Determining Relevant Objects

FPVG can only be meaningfully evaluated with questions for which the used object detector found both relevant and irrelevant objects. If, e.g., no question-relevant objects were detected, the question is excluded from FPVG evaluation. Hence, different subsets of the test (here: GQA's balanced val set) are evaluated depending on the used object detector. Table C.1 lists some statistics related to this for each of the object detectors used in our FPVG evaluations throughout this thesis. The set of relevant objects is determined by $IoU > 0.5$ between detected & annotated bounding box. The set of irrelevant objects excludes all detected bounding boxes that cover $> 25\%$ of any annotated relevant object to avoid any significant inclusion of relevant image content.

C.2 Feature importance ranking scores

Scores in Table 4.1 (Chapter 4) were calculated as follows:

Object Detector	#questions	#obj max	#obj (average) all | rel | irrel
Detectron2 (Wu et al., 2019)	114k	100	91 | 5 | 62
VinVL (Zhang et al., 2021)	110k	100	45 | 2 | 31

Table C.1 – Object detector bbox statistics for FPVG evaluation.

A question's "relevant" score measures how many of N annotated relevant objects in set $relN$ are among the $topN$ relevant objects (as determined and ranked by the used metric). It is calculated as $\frac{topN \cap relN}{relN}$, where a higher value is desirable for $FPVG_+$). A question's "irrelevant" score measures how many of M annotation-determined irrelevant objects in set $irrelM$ are among the $topM$ metric-determined relevant objects. It is calculated as $\frac{topM \cap irrelM}{irrelM}$, with a lower value being desirable for $FPVG_+$.

C.3 Model Training

In this section we include details for training procedures of models used in this work's evaluations. Generally, we use GQA's balanced train set to train all models and the balanced val set for evaluations. A small dev set (either a small, randomly excluded partition of the train set (20k questions), or separately provided in case of experiments on GQA-101k (Ying et al., 2022)) is used for model selection.

C.3.1 Visual Features

Detectron2-based visual features are generated by our object detector described in detail in Appendix A (and Appendix A.1, in particular).

Note that with exception of GQA-101k's repartitioned test sets (which mix questions from balanced train and val sets), no images used in testing were used in training (the latter generally applies to evaluations on GQA's original balanced set).

In chapters that refer to this appendix, most models are trained with the mentioned Detectron2-based visual features (1024-dim object-based visual features for a maximum of 100 detected objects/image) as input. For OSCAR+, we use the officially released pre-trained base model which uses VinVL visual features (Zhang et al., 2021).

C.3.2 MMN

MMN (Chen et al., 2021) consists of two main modules that are trained separately: A program parser and the actual inference model, which takes the predicted program from the parser as input. We mostly follow the settings in the official code-base but detail some aspects of our customization here.

For the inference model, we run up to 5 epochs of bootstrapping (using GQA's "all" train set (14m questions)) with Oracle programs and another up to 12 epochs of fine-tuning with parser-generated programs (from the official release), using GQA's balanced train set (1m questions). We use early stopping of 1 epoch and select the model by best accuracy on the dev set (using Oracle programs in bootstrapping mode and predicted programs in fine-tuning mode). The program parser was not retrained.

C.3.3 DFOL

DFOL (Amizadeh et al., 2020c) uses a vanilla seq2seq program parser, but neither code nor generated output for this is provided in the official code base. Thus, evaluations are run with ground-truth programs from GQA. DFOL is trained on a loss based on answer performance to learn weights in its visual encoding layers that produce an image representation similar to the one used by VLR (Reich et al., 2022), given high-dimensional visual features as input.

Training is done based on the official instructions for a complex 5-step curriculum training procedure. We train the first 4 curriculum steps with the entire 14 million questions in GQA's "all" training data partition, as specified in the instructions. As this is extremely resource intensive, we train for one epoch in each step. Finally, we run the 5th step with the "balanced" train data only (1m questions) until training finishes by early stopping of 1 epoch.

C.3.4 MAC

MAC (Hudson and Manning, 2018) is a monolithic VQA model based on a recurrent NN architecture which allows specification of the number of inference steps to take over the knowledge base. We follow the official training procedure guidelines given in the released code base and use 4-step inference. We train the model on GQA's balanced train set and use early stopping of 1 epoch based on accuracy on a dev set to select the best model.

C.3.5 UpDn, HINT, VisFIS

UpDn (Anderson et al., 2018) is a classic, straightforward attention-based model with a single attention step before merging vision and language modalities. We use the implementation shared by Ying et al. (2022). Following the scripts there, we train UpDn for 50 epochs and select the best model based on accuracy on a dev set.

HINT (Selvaraju et al., 2019) and VisFIS (Ying et al., 2022) are two VG-improvement methods. VisFIS is trained according to the released scripts. HINT is trained according to Shrestha et al. (2020) (using the VisFIS codebase), i.e. we continue training the baseline UpDn model with HINT (using GQA annotations to determine importance scores) for 12 more epochs and select the best resulting model (accuracy on dev set).

C.3.6 VLR

VLR (Chapter 3) is a modular, symbolic method that requires a full scene graph as visual representation. Similar to DFOL and MMN, it makes use of a (trained) question parser to produce instructional inference programs. The actual inference module does not require training. Training of the question parser and creation of the scene graph is described in Chapter 3. The used scene graph is described in Appendix A.

C.3.7 MCAN

MCAN (Yu et al., 2019b) is a Transformer-based model that uses co-attention layers and a form of multi-hop reasoning to hone in on attended vision and language information. We use the model implementation by Yu et al. (2019a) to train the "small" model (6 layers).

C.3.8 OSCAR+

OSCAR (Li et al., 2020) is a SOTA Transformer-based model that leverages pre-training on various V+L tasks and data sets. The subsequent release of new and elaborately trained visual features, known as VinVL (Zhang et al., 2021), further elevated its performance. We use this stronger version of OSCAR, called OSCAR+, in our evaluations. For training, we leverage the officially released pre-trained model and the VinVL features. Fine-tuning is done on GQA's balanced val set according to instructions accompanying the official release.

Note that we included results of UpDn (named "UpDn*", last line in Chapter 6, Table 6.1) trained with these stronger VinVL features, in accordance with our recommendation in the Limitation section (Chapter 4.4) for new/different visual feature sources.

Appendix D

Implementation Details for Chapter 7

D.1 Scene graph detection and symbolic feature creation

D.1.1 Visual feature generation

Symbolic features used in **GQA** evaluations were created based on classification outputs of the scene graph (SG) generator described in detail in Appendix A. The used SG-detector uses a Faster R-CNN (Ren et al., 2015) model for object detection for 1702 object classes. Attribute recognition for each detected object is done for each of the 39 attribute categories separately. Each category consists of two or more attribute classes (617 classes in total). Features for up to 100 objects are extracted per image.

For **VQA-HAT** evaluations and symbolic feature creation, we use the shared feature set generated with the object detector (OD) described in Anderson et al. (2018). The visual features from this OD provide the detected object class (one out of 1600 total) and up to one attribute class (out of 400 total) for 36 objects per image.

D.1.2 Symbolic feature creation

The different types of symbolic features are created as follows:

DET features use the maximum class from object detection as object name (300D GloVe embedding) and the normalized 4D bounding box coordinates as location information. For determining attribute information (300D GloVe embedding), each category's maximum class is involved. To represent the attribute information of all categories in a single word embedding, we take the average of all recognized attribute names.
Note that in VQA-HAT evaluations we do not use bounding box information (following Ying et al. (2022)). Furthermore, whenever no attribute information was provided, a word embedding for the "UNKNOWN" token (the average over all word embeddings) was used.

INF features use DET features as foundation. Modifications to object name information is realized by simple replacement. For attribute information modification in GQA evaluations, we first determine the attribute category of the annotated attribute that needs to be infused into an existing feature vector. Then, the embedding contributions of the (wrongly) recognized attribute of that category is replaced with an equivalently weighted embedding of the (correct) annotated attribute.

D.2 Model Training Details

For UpDn and LXMERT model training and evaluations, our experiments make use of implementations shared by Ying et al. (2022) (UpDn, LXMERT, VG-methods) and Tan and Bansal (2019) (LXMERT).

D.2.1 UpDn

UpDn models are trained for 50 epochs with 256 batch size for GQA-CP-large and 64 for VQA-HAT-CP. Model selection after 50 epochs is based on performance on the held-out dev set. We train either with or without one of the four examined VG-methods (VisFIS, AttAlign, HINT, SCR). Other hyperparameters, including training parameters specific to each VG-method were adopted from Ying et al. (2022).

D.2.2 LXMERT

Pre-training. We pre-train LXMERT (Tan and Bansal, 2019) from scratch for 30 epochs with 256 batch size, using all three types of symbolic features (DET, ORA, INF) to create three individual models. Model selection after 30 epochs is based on performance on the held-out dev set. We use the original

implementation and training scheme described in the original paper, with a few changes:

1. We exclude the attribute-related loss (unsuitable for more than a single attribute per object).
2. We restrict pre-training to the GQA-CP-large train set (see Table 7.1) to retain ID/OOD distributional integrity in our experiments.
3. Instead of the original setting of 36 objects, we use 100 visual objects (as provided by the used SG-detector).
4. We use a smaller version of the model to speed up training and adapt to the reduced amount of training data: Hidden layer dimensions are reduced from 768 to 128. Intermediate layer size is reduced from 3072 to 512. Number of attention heads per self-attention layer is reduced from 12 to 4.

Pre-training is done individually for each feature type (DET, ORA, INF). VG-methods are not applied in pre-training.

Fine-tuning. For fine-tuning, we train each model for 35 epochs with 64 batch size and use LXMERT's proposed two-layer classifier with softmax-based answer output. Model selection after 35 epochs is based on performance on the held-out dev set. Fine-tuning affects all LXMERT weights, not just the added VQA classifier.

Dissertation Revision

The following changes were made to the original first draft of this dissertation. Note that the reviewers' assessment of the dissertation's scientific merit was performed based on the original draft, i.e., before these changes.

Changes

- Added a section to draw attention to the ablation-type study performed for VLR in Chapter 3.4.1.
- Added a clarification about how dataset percentage numbers in Chapter 7.3, paragraph "Noisy features", were determined.
- Changed the used terminology throughout Chapter 7. In particular, "Flawed VG" was replaced with "Impaired VG". Other parts of the dissertation, where this terminology was used, were adjusted accordingly.
- Updated the bibliography to include our publication "Uncovering the Full Potential of Visual Grounding Methods in VQA" (Reich and Schultz, 2024), which was recently accepted at ACL 2024.