Visual Reasoning and Image Understanding: A Question Answering Approach

Md Moshiur Rahman Farazi

A thesis submitted for the degree of
Doctor of Philosophy
The Australian National University

December 2020

I declare that the research presented in this Thesis represents original work that I carried out during my candidature at the Australian National University. To the best of my knowledge, it does not contain material previously published by another person, except where due reference is made.

Md Moshiur Rahman Farazi
1 December 2020

To my mother, Niger Sultana, and my wife, Zannat Tahera Lamia.
Two of the greatest human beings in my life,
who shaped me into the person I am today.

Acknowledgments

I am thankful to the Almighty, the most Gracious, who in His infinite mercy has guided me to complete this dissertation.

Foremost, I would like to convey my gratitude to my primary supervisor Dr. Salman Khan, who has always been generous with his time, and provided guidance, critique and advice throughout the tenure my PhD degree. Your patience, mentorship and motivation has been the cornerstone of this dissertation. I would like to extend my gratitude to the chair of my panel, Professor Nick Barnes, who has always been supportive of my ideas, and provided support and freedom to explore new things. I would also like to thank my associate supervisor, Dr. Miaomiao Liu, for her insightful comments and discussions.

I would like to acknowledge the support provided by the Australian National University (ANU) and Commonwealth Scientific and Industrial Research Organisation (CSIRO) in the form of scholarships, travel bursary, training and overall infrastructure. Specially, I would like to thank CSIRO Scientific Computing for providing access to their super-computing infrastructure and GPU clusters that made development and deployment of very deep convolutional models possible.

I would like to thank my colleagues at CSIRO, specially Shafin Rahman, Arif Chowdhury, Khandaker Asif Ahmed, Sameera Ramasinghe, Lin Li, Ali Cheraghian, Saeed Anwar, Soumava Kumar Roy and Lars Petersson, for all the stimulating discussions that helped generate new ideas. I would also like to thank my peers at ANU, specially Zakir Hossain, Razon Kuman Mondal, Mehedi Hasan, Nasir Uddin, Rifat Ahmmed Aoni, Sheikh Mohammad Atiq and Imran Hasan, for their encouragement and friendship.

I would like to thank my friends in Canberra, specially Mohammad Rahim Rajin, Nishat Falgunee, Mohiuddin Ahmed, Fatema Raiyan, Ahmed Tanmay Tahsin Ratul, Atiya Rahman, Wasif E Elahi, Mahinoor Sultana, Mohammad Anwar-Us-Saadat, Rahnum Tasnuva Nazmul and Asaduzzaman Khan, who made us feel at home in Canberra, and provided support both in personal and professional capacity. I will cherish our friendship forever.

I wish to express my indebtedness to my father, Md Mukhlesur Rahman, and my mother, Niger Sultana, for providing me with a loving environment that helped me learn, grow and adapt. I am grateful for your sacrifices, encouragement and the tenacity with which you two have pushed me to be my very best self. I would like to thank my sister Onaiza Ferdaus and my baby brother Mushfiqur Rahman Farazi for their love and support. I am so blessed to have such a wonderful family.

Everyone in my extended family always has been tremendously supportive of my endeavours. Specially, I would like to thank my uncle Mohammed Mizanur Rahman and his family, who helped us settle down when we first moved to Australia, and has supported us throughout my PhD journey and beyond. I would like to take this opportunity to convey my heart-felt thanks to Maksudur Rahman, Mahbubur Rahman, Mufakharul Islam, Mahmodur Rahman, and all my extended family members for their encouragement and support. I would also like to thank my parents-in-law, Momammad Delwar Hossain and Shahnaz Manju, for their unwavering support and trust in me.

Finally, I would like to express my sincere gratitude to my best friend and my wife, Zannat Tahera Lamia. You have been an amazingly supportive and loving partner. You had to move to Canberra for me, leaving your family and a stellar career. Your support, patience and tremendous sacrifice were instrumental for the completion of this dissertation. Without you as my co-pilot in this bumpy ride, I would have been completely lost.

Abstract

Humans have amazing visual perception and cognition abilities which allow them to comprehend what the eyes see. At the core of human visual perception, lies the ability to translate and link visual information with linguistic cues, and navigate these two domains seamlessly. Superior visual reasoning and image understanding is required to enable an Artificial Intelligent (AI) agent to achieve human-level visual cognition. The premise of Visual Question Answering (VQA) is to evaluate an AI agent's ability in the three major components of visual reasoning by asking natural language questions about an image. *First*, the ability to combine multi-modal information from the visual and language domains, *second,* to attend to salient image regions and question parts relevant for answering the question, and *third,* to model the relationships between objects in the image. Drawing inspiration from human visual perception, in this dissertation, we develop visual question answering models that can comprehend holistic understanding of the scene for achieving superior visual reasoning and image understanding.

Based on the observation that humans tend to ask questions about everyday objects and their attributes in the context of a given image, we develop a *Reciprocal Attention Fusion (RAF)* model that generates image- and object-level attention maps to identify important visual cues with respect to the question. Further, we hypothesized that for achieving even better reasoning, a VQA model needs to attend to all object instances, as well as paying particular attention to the ones deemed important by the question-driven attention mechanism. Thus, we develop a *Question Agnostic Attention (QAA)* model that forces any VQA model to consider all objects in the image along with their learned attention representations, which in turn results in a better generalisation across different high-level reasoning tasks (e.g., counting, relative position). Furthermore, humans learn to identify relationships between objects and describe them with semantic labels (e.g., in front of, seating, helping) to get a holistic understanding of the image. We develop a *Semantic Relationship Parser (SRP)* that parses an image into subject-relationship-predicate triplets, and extracts visually grounded semantic features from the triplets. This enables VQA models to convert visual relationships to linguistic features, much like humans, and use them to generate an answer which requires high-level reasoning than only identifying objects.

In an open-world scenario, an AI agent tasked with VQA will be subjected to visual and linguistic concepts not found in the training set. Humans tend to infer about the unknown by comparing with the closest known concept. Inspired by this observation, we develop an *Exemplar based transfer learning* model imitating human reasoning, where the model learns to identify the closest visual-linguistic example from the training set and transfer that knowledge to reason about the unknown concept. To facilitate future research in this direction, we release a new VQA dataset,

dubbed *Open-World VQA* dataset, and demonstrate that exemplar based learning transfer helps to achieve superior reasoning and a better VQA accuracy across all standard datasets (including our proposed one).

One serious bottleneck in developing visual-linguistic AI agents is the computational burden. VQA models tend to become increasingly complex with the development of deeper Convolutional Neural Network (CNN) architectures, but often with a trivial improvement in accuracy. We conducted an extensive *Accuracy vs. Complexity Trade-Off* study to help the community navigate the maximum efficiency curve for developing visual-linguistic AI agents. We recommend design choices for two setups with unique design goals – one where a light-weight model is warranted with a reasonable accuracy (e.g., mobile platform) and another where higher accuracy is of main concern (i.e., an offline setting).

In summary, in this thesis, we endeavour to improve the visual perception of visual-linguistic AI agents by imitating human reasoning and image understanding process. This dissertation investigates how AI agents can incorporate different levels of visual attention, learn to use high-level linguistic cues as relationship labels, make use of transfer learning to reason about the unknown and also provides design recommendations for building such systems in practice. We hope our effort can help the community build better multi-modal AI agents the can skilfully comprehend what the camera sees.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

There have been significant advancements in Artificial Intelligence (AI) research empowering many aspects of modern society. Different Computer Vision (CV) and Natural Language Processing (NLP) techniques are being extensively investigated in order to develop systems that can reason and understand visual concepts like humans. Superior visual reasoning and image understanding is the key to developing AI agents that can interact with humans in the real word. When an AI agent is asked a natural language question about an image, much detailed understanding of the image is needed in order to provide an intelligent answer. Quantifying AI agents ability to answer natural language questions for a given image, allows us to evaluate its ability to analyze and translate visual information, perform visual reasoning required to answer the question, and translate the system response into natural language answer. Simultaneous progress in the Computer Vision (CV), Natural Language Processing (NLP), and Knowledge Representation & Reasoning (KR) domains are underway to solve this compelling AI task. Still, it remains an open ended problem as it requires a diverse set of AI capabilities including fine-grained recognition, object detection, activity recognition, knowledge based reasoning and common sense reasoning.

1.1 Motivation

1.1.1 Visual Turing Test

The holy grail of computer vision and natural language research is to develop an AI system that can seamlessly interact with humans through visual perception and intelligent conversation. Humans can seamlessly interact with a visual scene and answer complex open ended questions about it. They do it by identifying the objects and their attributes in a scene, recognising any relationship between the objects, reasoning about the question and relating the visual domain with the semantic domain to generate a natural language response. For an AI model to achieve that, it needs to perform each of these tasks separately and combine the individual results to generate a response. Significant progress has been made in the fields of machine perception [Krizhevsky et al., 2012; Szegedy et al., 2015; Girshick, 2015; Ren et al., 2015b; Simonyan and Zisserman, 2014; He et al., 2016, 2017] and natural language understanding [Mikolov et al., 2013b; Pennington et al., 2014; Mikolov et al., 2013a;

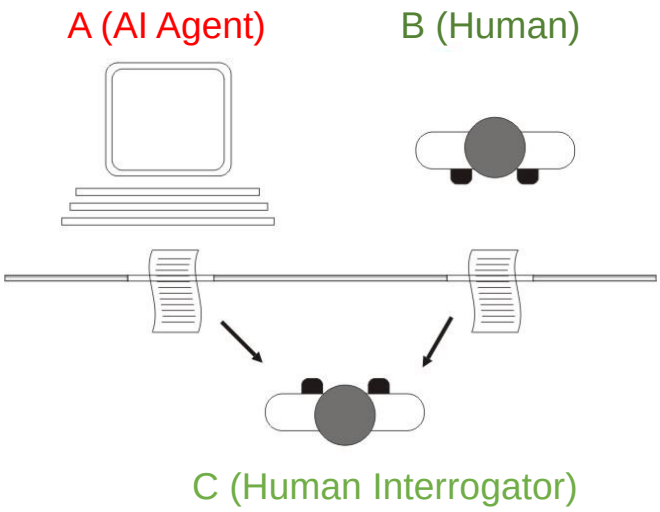
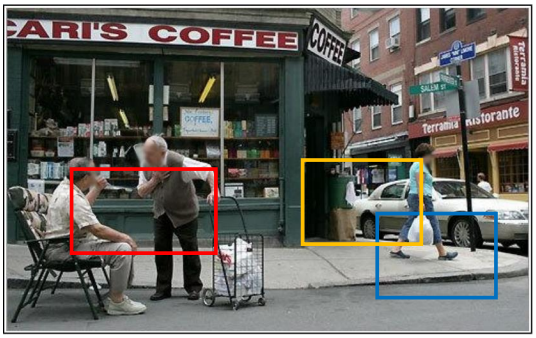


Figure 1.1: *Illustration of Turing's classic Imitation Game.* The classic version of the imitation game is played between three players, where player 'C' is the interrogator, tasked to determine which of the other player – A or B – is not human. 'C' is limited to using transcribed responses from the other players. Image adopted from WikiMedia Commons under ⓒ licence.

Karpathy and Fei-Fei, 2015; Devlin et al., 2018] to perform visual reasoning related tasks. When combined together, these sub-tasks create a holistic approximation of the complex human reasoning process. Interestingly, computer vision and language understanding models achieve near human accuracy in some of these sub-tasks (i.e., object recognition, language modelling), however, when combined together they preform poorly, resulting in low performance in cases where an integrated understanding is required e.g., answering simple questions about an image.

There have been several attempts to model human reasoning through creating benchmark evaluation tests to aid the development of intelligent machines that can *reason*. The first notable attempt to perceive machine intelligence was in the early 50's by Alan Turing. Turing first proposed the task of question answering as an *Imitation Game* (illustrated in 1.1) to determine if an intelligent agent has achieved indistinguishable reasoning skills (i.e., the ability to think) from a human [Turing, 1950]. At the beginning of the *Imitation Game*, the human interrogator, player 'C', is made aware that one of the other players, 'A' or 'B', is not a human rather an intelligent agent, and the interrogator is asked to distinguish them by asking both participants a series of questions. The conversation is limited to transcribed response to avoid a bias from audio or speech processing. By the end of the game, if the interrogator cannot confidently distinguish the intelligent agent from the human, the intelligent agent passes the test. Turing's choice of question answering as *the* task to measure machine intelligence, bolsters the ability of question answering task to act as a strong indicator of human-level reasoning.



1. Q: Is there a person in the blue region?	A: yes
2. Q: Is there a unique person in the blue region? (Label this person 1)	A: yes
3. Q: Is person 1 carrying something?	A: yes
4. Q: Is person 1 female?	A: yes
5. Q: Is person 1 walking on a sidewalk?	A: yes
6. Q: Is person 1 interacting with any other object?	A: no
⋮	
9. Q: Is there a unique vehicle in the yellow region? (Label this vehicle 1)	A: yes
10. Q: Is vehicle 1 light-colored?	A: yes
11. Q: Is vehicle 1 moving?	A: no
12. Q: Is vehicle 1 parked and a car?	A: yes
⋮	
14. Q: Does vehicle 1 have exactly one visible tire?	A: no
15. Q: Is vehicle 1 interacting with any other object?	A: no
17. Q: Is there a unique person in the red region?	A: no
18. Q: Is there a unique person that is female in the red region?	A: no
19. Q: Is there a person that is standing still in the red region?	A: yes
20. Q: Is there a unique person standing still in the red region? (Label this person 2)	A: yes
⋮	
23. Q: Is person 2 interacting with any other object?	A: yes
24. Q: Is person 1 taller than person 2?	A: amb.
25. Q: Is person 1 closer (to the camera) than person 2?	A: no
26. Q: Is there a person in the red region?	A: yes
27. Q: Is there a unique person in the red region? (Label this person 3)	A: yes
⋮	
36. Q: Is there an interaction between person 2 and person 3?	A: yes
37. Q: Are person 2 and person 3 talking?	A: yes

Figure 1.2: *Sequence of questions about an image during a visual Turing test.* The three bounding boxes are random and are used for each of the four instantiations, i.e., Blue region for Person 1, Yellow region for Vehicle 1 and Person 2, and Red region for Person 3. Each question is associated with a designated bounding box. The figure is adapted from [Geman et al., 2015].

Turing's Imitation Game is limited to one mode of input (i.e., natural language), whereas humans can combine visual and semantic inputs to achieve an even higher level of reasoning. Geman et al. [2015] proposed a query-based test for computer vision system called the Visual Turing Test, to quantitatively measure a computer vision system's ability to interpret ordinary images in a natural scene. In this test, the computer vision system is subjected to a series of binary question about the image, generated by an automatic query generator. The questions are unambiguous and curated by a human to make sure that these questions can be answered by using commonsense knowledge about the scene content and would not require any external knowledge. One example of such a system is illustrated in Fig. 1.2. As shown in the figure, the questions require basic reasoning skills the can be sourced from the image itself, not requiring any background information. Further, the questions follow a natural storyline, somewhat similar to what humans do while looking at a new image. Its worth mentioning that the Visual Turing Test was proposed only as an evaluation setup, not a protocol for developing intelligent computer vision systems. Through Visual Turing Test, question answering was further established as the preferred task to evaluate visual reasoning and image understating ability of an AI agent.

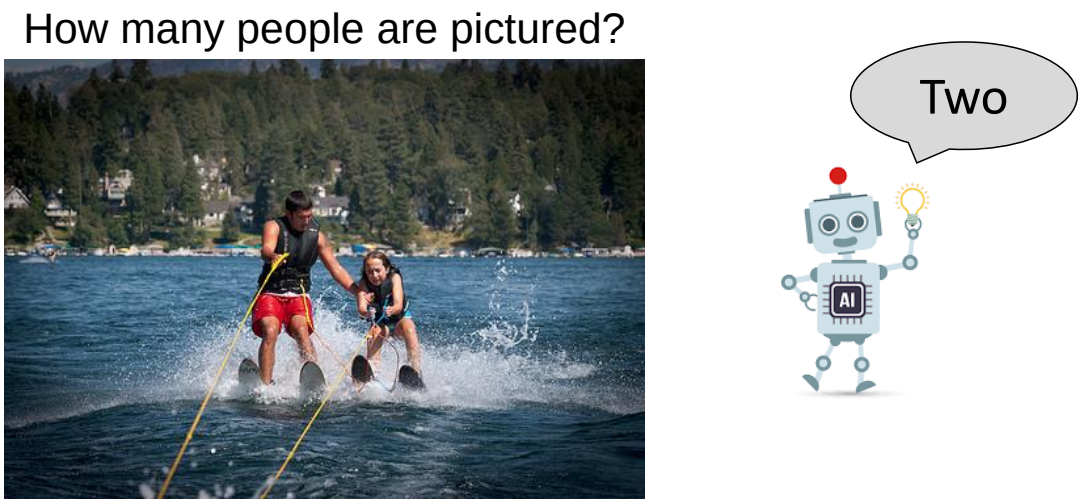


Figure 1.3: *Illustration of a Visual Question Answering Task.* Given a natural language question about an image, a visual question answering model is asked to provide a natural language answer. Image from Visual Genome [Krishna et al., 2016] dataset.

1.2 Visual Question Answering

Even though questions are asked about a visual input in the visual Turing test, the questions are limited to a binary nature. In reality, an AI agent tasked with question answering will encounter free-form open-ended questions about the scene and its contents, which require detailed understanding of the image and related concepts. Thus for achieving human-level reasoning, an AI agent needs to perform several sub-tasks related to visual perception and language understanding, and must get a holistic understanding of the scene from a visual and semantic perspective. As the progress in a field is typically facilitated and measured by creating a set of benchmark evaluation tests, there is a need to establish such benchmarks based on a task that can sufficiently imitate the complexity of human reasoning process.

Visual Question Answering (VQA) [Antol et al., 2015] is an *AI Complete* task where an AI agent is presented with an image and a natural language question about the image, and is asked to provide an intelligent answer (Fig. 1.3). VQA has recently emerged as a viable alternative to the visual Turing test as it can imitate the complexity of human reasoning through complex question answering. The questions asked can be arbitrary in nature and require simple to very complex reasoning ability. For example, questions related to simple reasoning task can be, 'What is the man carrying?' (object recognition), 'Is the man carrying an umbrella?' (object detection), 'What color is umbrella?' (attribute classification), 'Is the day sunny?' (scene classification). On the other hand, there might be questions that require advanced reasoning skills like 'What is the color of the smaller umbrella on the left of the blue one?' (positional reasoning), 'How many umbrellas are there?' (counting), 'Is it likely to rain today?'(commonsense knowledge). Therefore, in a VQA setting, an AI agent

can be subjected to a wide variety of questions which would require a diverse set of AI capabilities in computer vision, natural language processing and commonsense reasoning to predict the correct answer. If an AI agent can perform on-par with humans in a VQA setting, it would mean that the model has been able to emulate human reasoning in certain aspects and can perform the sub-tasks required for question answering. Therefore, in this dissertation, we develop better Visual Question Answering models taking inspiration from human visual perception and reasoning abilities.

1.3 Definitions

In this section, we define some terminology that has been used throughout this dissertation. The definitions are arranged in alphabetical order.

- **AI Agent:** An AI agent is an entity that perceives information from the environment through its sensors and carries out a task based on the perceived information. In the scope of this dissertation, we focus on AI agents that perceive visual information from the environment as an image, and are tasked with answering a question about the input image.

- **AI-Complete Task:** The term 'AI-complete' was formalized by Stuart C. Shapiro in the book *Encyclopedia of Artificial Intelligence* [Shapiro, 1992]. The name draws a parallel with NP-Complete or NP-Hard task from computational complexity theory. A task is AI complete when it is too hard to be solved by a single algorithm. It would require an algorithm to perform equally well in unexpected scenarios and when subjected to ambiguity. Finding a general solution to an AI-Complete task would mean that the AI has some sense of general consciousness. The classical examples of two AI complete tasks are, (1) designing a computer vision system that can *see*, and (2) a natural language processing model that can *converse* as well as a human.

- **Attention Map:** The Attention Map is a learned matrix that represents the relative importance of each spatial image location with respect to the question. The size of the attention map is set equal to the resolution of spatial gird if image level feature is used, and to the number of top object proposals if object level feature is used.

- **Bidirectional Encoder Representations from Transformers (BERT):** BERT is a bidirectional language encoder model [Devlin et al., 2018] that uses an attention mechanism that learns contextual relations between words in a sentence and generates sentence level encoding. As opposed to directional models such as Skip-Thought which *reads* the text input sequentially (e.g., left-to-right), BERT encoder *reads* the entire sentence at once enabling the models to learn the context of a word based on all of its neighbourhood, irrespective of the reading sequence.

- **External Knowledge Base:** The External knowledge base is a structured representation of semantic and/or visual data, generated from unstructured large corpus of text (e.g., Wikipedia) and/or image repository (e.g., Google image search). The VQA model employs an *inference engine* that identifies relevant information from the knowledge base to help the overall reasoning of VQA model.

- **Feature Extraction:** Feature extraction is a process of generating meaningful feature representations from the input raw image pixels through a series of computational steps. The generated feature representation is highly discriminative and non-redundant than the input data, and can be used more efficiently for further downstream tasks compared to using the raw image pixels.

- **Image Level Feature:** The image level visual features are extracted from the last convolutional layer of deep Convolutional Neural Network (CNN) architectures (e.g., VGG16 [Simonyan and Zisserman, 2014], ResNet [He et al., 2016]). The resolution of the image level feature representation is fixed and is set equal to the spatial grid locations of the CNN architecture [Khan et al., 2018].

- **Joint Embedding Space:** Joint embedding space is an intermediate feature representation space, where the input feature representation are combined to be used for further downstream visual-linguistic tasks.

- **Multi-modal Fusion:** When feature representations from two or more modalities are combined together through simple operation like summation or concatenation, or complex bilinear models, it is called multi-modal fusion.

- **Object Level Feature:** The object level feature representation consists of a set of visual features of the image regions that contain objects. The object features are extracted using object detectors (e.g., Faster-RCNN [Ren et al., 2015b]).

- **Question Agnostic Attention:** The process when the attention map is generated in a bottom-up fashion without considering the question, is called Question Agnostic Attention.

- **Relationship Triplet:** The semantic relationship triplet contains the class labels of two objects in an image and a relationship label depicting the semantic relationship between them.

- **Semantic Feature Representation:** Semantic features are the conceptual representations of a single word, a part of a word, or a phrase, that can be used to parameterize the linguistic meaning of a sentence. In the scope of this thesis, the semantic features are extracted from questions via a pretrained language model to obtain vector representations of the question words.

- **Semantic Relationship Feature:** The semantic feature representation of the relationship triplets of an image is called semantic relationship feature.

- **Skip-Thought Model:** Unlike word2vec which operates at word level, Skip-Thought model [Kiros et al., 2015] encodes a sentence into a fixed length vector representation called skip-thought vector. In the context of VQA, a skip-thought model trained on larger text corpora is used to generate vector representation of the question as semantic features.

- **Stochastic Gradient Descent (SGD):** Gradient descent is an iterative optimization algorithm that is used to minimize an objective function by moving towards the steepest descent as calculated from the negative of the gradient. However, the gradient descent algorithm becomes computationally infeasible as the visual and language models are considerably more *data heavy*. Stochastic approximations of the gradient descent is used in such cases, where the actual gradient of the dataset is replaced by an estimate from a subset (often called 'mini-batch'). Such stochastic approximations of gradient descent algorithm is called Stochastic Gradient Descent.

- **Visual Feature:** Representation of high and/or low level visual cues in a discriminative feature space is called visual features. The visual features of an image are usually of lower dimension than the original raw pixel based representation, and can be used more efficiently for downstream tasks involving neural networks or other machine learning models.

- **Word2vec:** Word2vec represents a family of shallow neural network architectures that effectively converts natural language words into vectorized word embedding called *word vector*. One of the most popular word2vec model is Skip-gram [Mikolov et al., 2013b], which is a neural network with a single hidden layer trained to predict the immediate neighbors of a given word. The learned weights of the hidden layers is considered as the vector representation of the input word.

1.4 Research Objective and Contributions

The task of Visual Question Answering (VQA) requires human level capability of analysing image and question, and combining these visual and language responses to generate an adequate answer. Developing a system to answer natural language questions for a given image provides the opportunity to explore the system's ability to analyze and translate visual information, and perform visual and language related tasks, which is a passive measure of visual reasoning capability. However, this problem is far from being solved. To aid the research for improving visual reasoning through better visual question answering models, in this thesis, we identify four major knowledge gaps in vision and language systems, and make contributions towards bridging these gaps. We enumerate these in the following sections.

1.4.1 Object central attention

Human attention in an image is object centric [Judd et al., 2009]. During visual linguistic tasks, such as VQA, it is only natural that humans mostly ask questions about objects in the context of the given image. This requires humans to identify the objects and its attributes, and the functionality of that object in the context of the given image.

Knowledge Gap: Current VQA models are limited to a single level visual feature representation, either grid level image features, or object level features. However, for achieving superior visual reasoning, a model needs to attend to both levels of visual features through efficient attention mechanism. Further, the visual linguistic attention mechanisms are general propose and are limited to learning attention distribution based on the questions in the dataset. This limits the model's ability to answer question that require reasoning over all objects in the image (e.g., counting).

Contributions: To the best of our knowledge, we propose the first VQA model that combines different level of visual and semantic features through an efficient attention mechanism. This allows the VQA model not only to look at the whole image or only at a specific object, rather to reason about the objects in local and global context. We list our contributions toward object central attention as follows:

- **Reciprocal Attention:** In Chapter 3, we propose a reciprocal attention model that captures the complex interplay between image grid and objects level features. This provides complementary understanding of the rich scene semantics from both image and object level.

- **Question Agnostic Attention:** The attention map over the image level or object level features is learned with respect to the question. In Chapter 4, we propose an attention mechanism that is question agnostic, and is learned on the objects present in the image. This forces the VQA model to look at all object instances as spatially grounded *object maps*.

- **Spatial and Channel Attention in joint embedding space:** In Chapter 5, we propose an attention mechanism that selectively attends to visual scene details by applying spatial and channel attention on the joint feature embedding learned from visual and semantic inputs. We show that such selective attention mechanism is useful when the VQA model is answering questions about an unknown concept by transferring knowledge from the known concepts.

- **Mutual and Self Attention:** In Chapter 6, we propose to combine multi-modal attention learned by fusing visual and semantic features, and mono-modal self attention learned by applying multi-head attention on visual and semantic features separately. This helps to achieve holistic understanding of the image based on semantic relationship features.

1.4.2 Transfer learning to reason about unknown concepts

An open-world setting for VQA would require a vision system to acquire knowledge over time and later use it to intelligently answer complex questions about unknown concepts for which no visual and linguistic examples were available during training. Humans reason about an unknown concept by finding a similar known concept and inferring from the known about the unknown; an AI agent tasked with VQA should be able do the same.

Knowledge Gap: The major limitation for developing VQA models in an open-world setting is the lack of visually and semantically separated concepts for training and testing. Further, there is a need for an efficient way to store and access the knowledge base built from the known examples.

Contribution: We develop a VQA model for the open-world, when required to reason about an unknown concept, it identifies similar visual and semantic concepts in the joint embedding space and transfers knowledge to facilitate the reasoning process. This approach does not require access to external knowledge base and/or expensive pretraining, rather employs an efficient search and retrieval on its training states, which makes it generalizable across datasets. We discuss this in detail in Chapter 5 which includes the following key contributions:

- We reformulate the VQA problem in a transfer learning setup where closely related known instances from an exemplar set are used to reason about unknown concepts.
- We propose a new Open-World VQA dataset, dubbed OW-VQA, to enable impartial valuation of VQA algorithms in a real-world scenario and report impressive improvements over recent approaches with our proposed model.
- We present a novel network architecture and training schedule that maintains a knowledge base of exemplars in a rich joint embedding space that aggregates visual and semantic information.
- We propose a hierarchical search and retrieval scheme to enable efficient exemplar matching on a high dimensional joint embedding space.

1.4.3 Semantic relationships in visual reasoning

For understanding how objects interact in a scene image, AI agents not only require to consider the visual features of the objects and its parts, but also need to leverage the natural language grounding (i.e., relationship label). Enriched semantic relationship modeling captures high level verbal and non-verbal cues which are required for achieving a better reasoning ability.

Knowledge Gap: Semantic relationship modelling is an important missing piece in the existing VQA literature. Current VQA models only represent relationships between objects as a combination of visual features of the subject and object bounding boxes.

Contribution: To the best of our knowledge, we propose the first general purpose semantic relationship feature extraction model in Chapter 6 and showcase our results which strongly advocate for further investigation on better relationship modeling in the semantic domain, a direction less explored so far in the VQA community. We demonstrate that under an oracle setting, these semantic relationships can bring the performance of a VQA model on par with human-level accuracy. Our contributions regarding semantic relationship modelling are as follows:

- We propose a general purpose semantic relationship parser that takes an image as input and generates semantic relationship label and features that can be used for multi-modal visual-linguistic downstream tasks such as Visual Question Answering.
- We showcase the effectiveness of using the semantic relationship features by reporting superior performance over models employing similar visual relationship features. Further, in an oracle setting where ground-truth relationship labels are available, we obtain a 25% accuracy gain compared to a state-of-the-art model that only uses visual features.

1.4.4 Complexity vs. Accuracy Trade-off

Recently, the accuracy of VQA models has almost saturated in leading VQA tasks. Augmentation with more computationally expensive multi-modal fusion and attention operation often results in trivial improvements in performance. This is a major bottleneck for developing and deploying AI agents performing visual-linguistic tasks.

Knowledge Gap: VQA models consist of several building blocks which individually and collectively contribute to achieving higher VQA accuracy. However, there is a lot of variation and diversity in the literature on how different design choices might affect the overall performance. Building more computationally expensive VQA models often improves the VQA accuracy slightly. However, there is a need for a systematic study on the influence of key components commonly used within VQA models on the efficiency and final performance.

Contribution: To help the community navigate the VQA accuracy vs. complexity trade-off, in Chapter 7 we present our findings and recommendations to help researchers find optimum design choices when building vision and language models. Our contributions regarding this include:

- We establish an unified VQA architecture that supports the three most popular meta-architectures, namely visual features extractor, bilinear fusion and co-attention, and a comparative evaluation protocol by varying these meta-architectures.
- We perform an extensive evaluation on three challenging VQA datasets (i.e., VQAv2, VQA-CPv2 and TDIUC) with 6 visual feature extractor, 7 bilinear fusion model and 2 attention mechanism, and generate elaborate accuracy vs. complexity trade-off curves.

- We provide design recommendations for resource constrained setting as well as for achieving state-of-the-art performance on several challenging VQA datasets.

1.5 Thesis Outline

The remaining chapters of the thesis are organised as follows:

Chapter 2 – Background: This chapter provides a succinct review of the literature relating to the existing VQA datasets and the basic building blocks of a visual linguistic model, namely feature extractor, multimodal fusion and attention. Different variants and combinations of these blocks are employed to achieve superior ability to perform complex multimodal tasks such as VQA; we aim to identity knowledge gaps in this pursuit and highlight the contributions of this thesis to bridge these gaps.

Chapter 3 – Reciprocal Attention Fusion: In this chapter, we introduce reciprocal attention fusion that co-attends to both object and image-level features. We propose a complementary attention mechanism that looks at the objects and the whole image, while answering questions. To showcase the effectiveness of this approach we perform experiments on large scale VQA datasets.

Chapter 4 – Question Agnostic Attention: In this chapter, we propose a question agnostic attention model that forces the VQA model to learn an additional attention distribution over the objects in the image, irrespective of the input question. We show that this approach allows simple VQA models approach near state-of-the-art accuracy and pushes the performance of superior VQA models even higher.

Chapter 5 – Exemplar based Knowledge Transfer: In this chapter, we enable VQA models with the ability to reason about the unknown by finding a similar known concept from the training set, and transferring the knowledge to perform reasoning about unknown concept. Further, we also propose a new VQA dataset for facilitating the development of VQA models in an Open-World setting.

Chapter 6 – Semantic Relationship Parser: In this chapter, we propose a semantic relationship parser, that converts the visual relationships in an image to semantic relationship labels and complements the visual features of the image for a higher level visual reasoning.

Chapter 7 – Accuracy vs. Complexity Trade-off: We develop a guide for navigating the VQA accuracy vs. computational complexity trade-off curve. Through extensive evaluation of different VQA meta-architectures, we proposed two design recommendations, one where lower computationally complex model with moderate VQA accuracy is desirable, and another, where accuracy is critical with a relatively higher computational cost.

1.6 List of Publications

In this section, the publications associated with this dissertation are listed:

- **Moshiur R. Farazi** and Salman Khan, *Reciprocal Attention Fusion for Visual Question Answering.* Published in: Proceedings of the British Machine Vision Conference (BMVC). 2018. [Farazi and Khan, 2018].
- **Moshiur R. Farazi**, Salman Khan and Nick Barnes, *Question-Agnostic Attention for Visual Question Answering.* Publised in: Proceedings of the International Conference on Pattern Recognition (ICPR), 2020. [Farazi et al., 2020d].
- **Moshiur R. Farazi**, Salman Khan and Nick Barnes, *From Known to the Unknown: Transferring Knowledge to Answer Questions about Novel Visual and Semantic Concepts.* Published in: Image and Vision Computing (IVC) Journal, Elsevier. [Farazi et al., 2020c].
- **Moshiur R. Farazi**, Salman Khan and Nick Barnes, *Attention Guided Semantic Relationship Parsing for Visual Question Answering.* Currently under review for a journal publication. Preprint: [Farazi et al., 2020a]
- **Moshiur R. Farazi**, Salman Khan and Nick Barnes, *Accuracy vs. Complexity: A Trade-off in Visual Question Answering Models.*Currently under review for a journal publication. Preprint: [Farazi et al., 2020b].

Chapter 2

Background

In this chapter, we review the existing literature relating to vision and language tasks where high-level reasoning on the image is required to generate a natural language response. We first look at the existing VQA datasets, and discuss their features and shortcomings towards the evaluation and development of VQA models. Then we focus on three major components of VQA models, namely visual attention, learning about the unknown concepts and understanding visual relationships to identify knowledge gaps, which are addressed in this dissertation.

2.1 Datasets for Visual Question Answering

To accurately capture the complexity, difficulty and diversity of visual and semantic concepts in the real world, a VQA dataset needs to be adequately large. The generation of the dataset has to be carefully curated so a model cannot exploit visual or semantic bias to achieve *fake* higher accuracy. Further, there needs to be natural ambiguity in the dataset where humans use *common sense* to infer the most probable answer. This would ensure an AI agent is also able to pick up the salient visual and semantic cues for answering the question correctly. In the following sections, we discuss some of the notable visual question answering datasets, put their merits and shortcomings into prescriptive, which necessitated the newer and better VQA datasets and evaluation metrics.

2.1.1 DAQUAR

Malinowski and Fritz [2014] proposed the first notable VQA dataset called the DAtaset for QUestion Answering on Real-world images (DAQUAR). Comparing with today's standard, it was relatively small, consisting of 1449 RGBD images from NYU-depth-v2 dataset [Silberman et al., 2012], paired with questions asking about type, color and count of objects. The full DAQUAR dataset consists of 894 object classes with 6794 training and 5674 testing image question pairs. There is also an even smaller version called reduced-DAQUAR with 3825 training and 297 testing image-question pairs related to only 37 object classes.

The main limitations of the DAQUAR dataset are three fold. **First,** the scope of the dataset is only limited to indoor scene and it does not capture variety of reasoning skills required to answer question about natural scenes. **Second,** the dataset is of very limited size in terms of number of images and associated questions. Sufficiently large models will easily overfit on the dataset and would not generalize well to new scenarios. Furthermore, the images is the dataset have significant overlaps, occlusion and clutter between objects which makes it harder even for humans to answer correctly (i.e., human accuracy is only 50.2%). **Third,** WUPS score [Wu and Palmer, 1994] is used to evaluate the accuracy of the predicted answer, which measures the similarity between the predicted answer and the ground truth on their longest common subsequence in the taxonomy tree. This measure provides ambiguous results while analysing the comparative performance as it is not a direct measure of accuracy. Even though [Malinowski et al., 2017] further extended the DAQUAR dataset by extending the ground truth answers and consensus based evaluation metric, the image quality, limited scenes and number of question made it harder for the community to use this as a benchmark set for VQA models. Nonetheless, this dataset sets up the ground work for the development of future VQA datasets.

2.1.2 COCO-QA

Ren et al. [2015a] proposed COCO-QA dataset where the images are sourced from the MS-COCO dataset [Lin et al., 2014] with their associated human annotated captions. The images contain indoor and outdoor natural scenes. The questions are automatically generated from image captions in 4 categories, namely Object, Number, Color and Location. For example, if an image is accompanied with a caption *'A person standing next to a person sitting down holding a rainbow color umbrella'*, one of the possibly generated question would be 'What color is the umbrella?'. The dataset consists of 123,287 images associated with 78,736 training and 38,948 testing QA pairs. As for evaluation, similar to DAQUAR dataset, WUPS score is calculated to quantify the accuracy of the predicted response.

Even though compared to DAQUAR dataset, COCO-QA is larger in size and have natural images, the main shortcoming of the COCO-QA dataset is its automatic question generation process. As the questions were generated directly from the COCO captions through NLP algorithms, often questions contained grammatical errors and were unintelligible. This is mainly due to the NLP models limitation to process longer captions into smaller independent clauses and generate question from them. Further, the vocabulary of the dataset was also limited to the MS-COCO captions, which limits the models ability to comprehend wide variety of semantic concepts.

2.1.3 VQA Dataset

One of the most widely used benchmark dataset for visual question answering task is Visual Question Answering dataset, commonly dubbed as the VQA dataset. Two

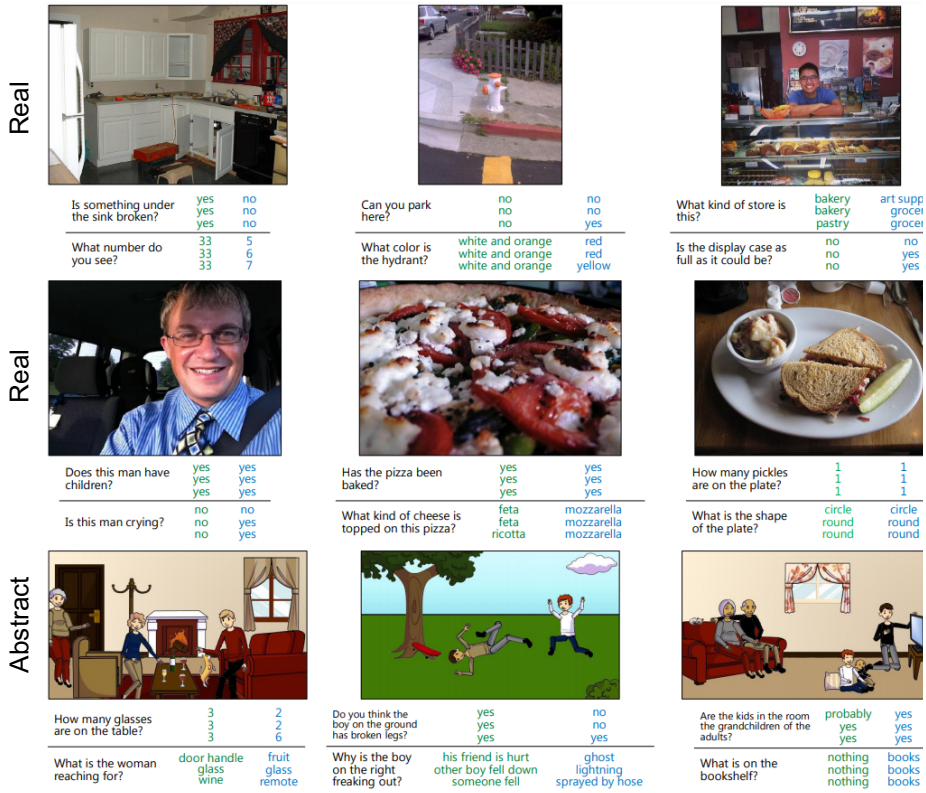


Figure 2.1: *Example image, question and answer triplet from VQA version 1 dataset.* The first two rows contains the examples from the 'real' portion of the dataset, and the last row shows examples from the 'abstract' portion of the VQA v1 dataset. The answers in green are given when looking at the image and the answers in blue are given when not looking at the image. Images are adopted from Antol et al. [2015].

versions of this dataset have been proposed, VQA-v1 [Antol et al., 2015] and VQA-v2 [Goyal et al., 2017].

VQA-v1: This dataset consists of two parts; the first part containing natural images is called VQA-real, and the second part consisting of rendered images is called VQA-abstract (examples from VQA-v1 dataset is shown in Fig. 2.1). For VQA-real potion of the dataset, 82,783 training, 40,504 validation and 81,434 test images were sourced from the MS-COCO [Lin et al., 2014] dataset. The question and answers were annotated by humans using the Amazon Mechanical Turk (AMT)[1]. The human annotators were first shown an image and asked to produce a question about the image, that a human can easily answer but would be hard for a smart robot. The generated questions were given to another set of human annotators to provide natural language answers. For each question at least 10 responses were collected from different annotators. Thus the VQA-v1 real dataset has 289,349 training, 121,512 validation and 244,302 test image-question pairs, which is much larger than DAQUAR and COCO-QA dataset. On the other hand, VQA-abstract dataset has 50,000 ren-

[1] https://www.mturk.com/

Figure 2.2: *Complementary Image-Question pairs in VQA v2 dataset.*

dered images containing *paper doll* models of 20 human, 30 animal and 100 object cartoons. The questions and answers for VQA-abstract also involved human annotators following the same protocol as the VQA-real dataset. Fig. 2.1 shows examples from both 'abstract' and 'real' components of VQA-v1 dataset.

'Real' portion of the VQA-v1 dataset was adopted as the benchmark test for visual question answering task. This is largely due to the fact that it encapsulated the natural complexity and diversity of question answering task in the real world. Further, it provided a strong human baseline accuracy of 83.30% which allowed VQA models to compare their performance against humans. Further, to quantify the performance, VQA dataset proposed a simple consensus based accuracy metric as follows:

$$\text{VQA}_{\text{acc}} = min\left(\frac{\#\text{ of humans provided the predicted answer}}{3}, 1\right) \tag{2.1}$$

which means the predicted answer will be given 100% accuracy if at least 3 human annotators who helped create the VQA dataset gave the exact answer predicted by the model.

VQA-v2: The VQA-v1 dataset had significant *Language Bias* that prompted the release of a second version of the dataset, called VQA-v2 dataset (examples from VQA-v2 dataset is shown in Fig. 2.2). The language bias was due to the very fact that made the dataset popular in the first place, which is the use of human annotators.

For example, if the question is 'Is there an umbrella in the image?', the most probable answer in the dataset is 'Yes', as the human annotators are most likely to ask about an object present in the image. Further, when asked counting question like 'How many umbrellas are in the image?', the most probable answers for overwhelmingly large number questions about counting questions are 'three' or 'two', not 'seventeen'. These kind of language biases are side effects of using natural images and human annotators. Such biases allow VQA models to *cheat*, allowing them to achieve *fake* accuracy gain.

The VQA-v2 dataset was proposed as a more balanced version of the original VQA-v1 real dataset. The language bias was mitigated by pairing every question with complementary images, where the same question will have different answers for the complementary images. Fig. 2.2 shows a pair of balanced question image-question-answer triplets from the VQA-v2 dataset. Such balancing prevented VQA models from *memorizing* the common answers. After the addition of complementary image-questions pairs, the VQA-v2 almost doubled in number of question size (total $1.1M$, training $440K$, validation $214K$ and test $447K$) compared to VQA-v1 real dataset. The inclusion of the complementary image-question pairs reduced the language bias to some degree, but the training and test question-answer distribution remained the same. This allowed VQA models to leverage training set priors that benefited the model at test time.

2.1.4 Visual Genome

Visual Genome [Krishna et al., 2016] dataset consists of $108K$ images selectively sourced from MS-COCO [Lin et al., 2014] and YFCC100M [Thomee et al., 2016] datasets. This is one of the largest natural image dataset containing dense annotations of objects, attributes, and relationships between objects, as well as crowd-sourced question-answer pairs associated with the image (examples from Visual Genome dataset is shown in Fig. 2.3). The images in the dataset are as graphs where each node is an object and the edge connecting two nodes is the relationship between them. Such graph based representation of visual content is commonly known as scene-graphs (the term coined by Johnson et al. [2015]). Each image has multiple region based scene-graphs and a holistic scene-graph representation of the whole image. This structured representation is visually grounded and richer than image-level annotation (i.e., caption) or object-level annotation (i.e., object label). Further, the QA part of the dataset has $1.8M$ image-question pairs, commonly referred as 6W questions (*what, where, how, when, who and why*).

The visual genome dataset is widely used in VQA models for complementary training as they are non-overlapping with the testing image-question pair of the VQA datasets [Antol et al., 2015; Goyal et al., 2017] and several other visual language task as scene-graph generation [Xu et al., 2017; Yang et al., 2018], image captioning [Gu et al., 2019] and many more. One of the major differences between traditional QA datasets and visual genome dataset is that it does not have any binary (i.e., Yes/No) question. The creators of the dataset argue that the binary questions are easy for the

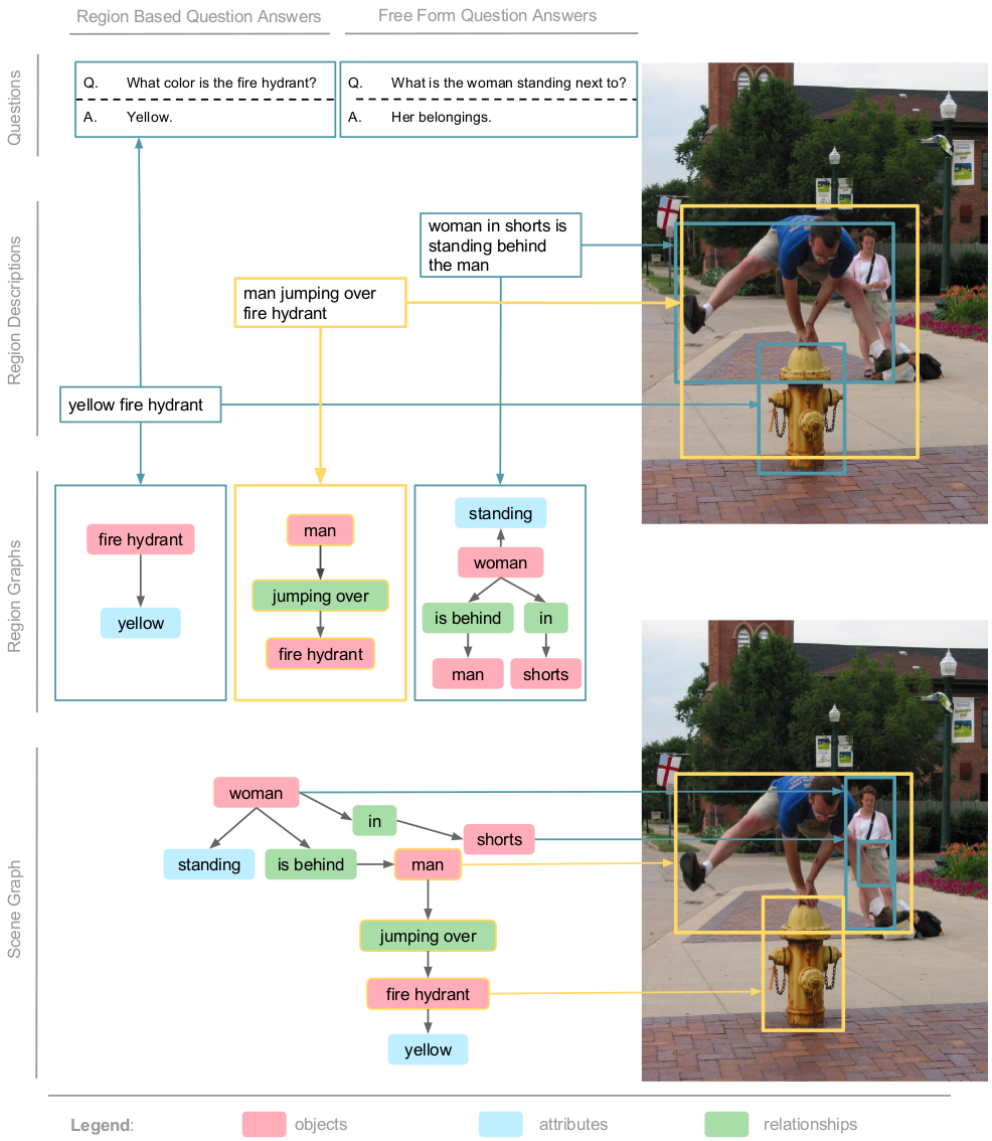


Figure 2.3: *An example from Visual genome dataset.* Each image contains a region-level scene graph that describes a localized portion of the image. There are two types of question-answer (QA) pairs: one freeform QAs and another region-based QAs. Figure adopted from Krishna et al. [2016].

models to memorize and model can easily take advantage of the language bias while answering these type of questions. On the contrary, Zhang et al. [2016] and Andreas et al. [2016] showed that with balanced binary questions, spatial reasoning and high-level inference is hard for VQA models to answer. Nonetheless, visual genome dataset paved the way to perform structured reasoning over the image and its parts which has been a major factor in improving the performance of visual question answering models.

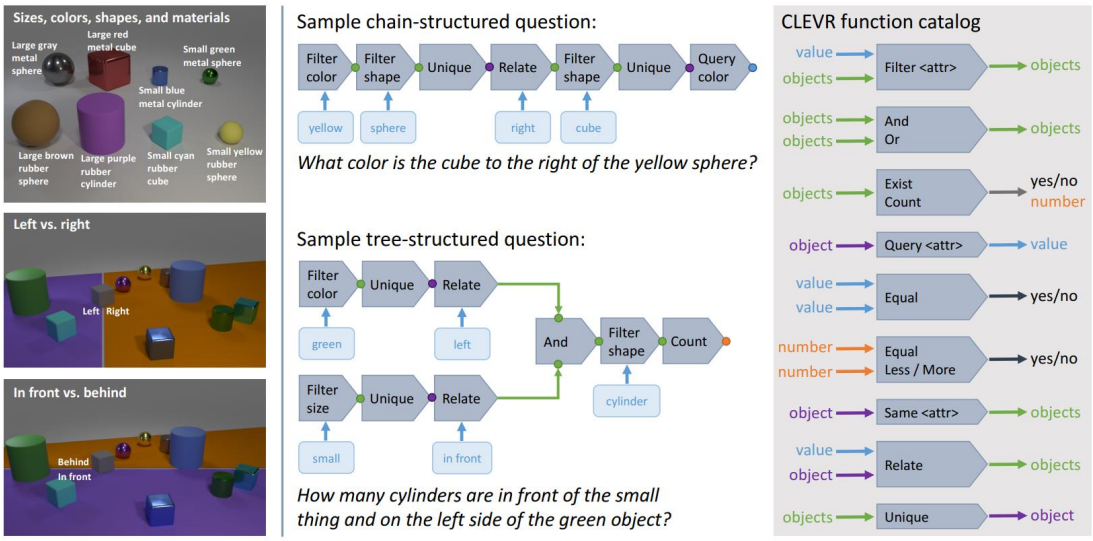


Figure 2.4: *Example from CLEVR dataset [Johnson et al., 2016].* Left: Shapes, attributes, and spatial relationships. Center: Examples of questions and their associated functional programs. Right: Catalog of basic functions used to build questions. Figure adopted from Krishna et al. [2016].

2.1.5 CLEVR

It is often hard to pinpoint the reasoning task (i.e., recognition, reasoning, commonsense knowledge) in which the VQA model needs improvement due to the holistic nature of the VQA task. Johnson et al. [2016] proposed Compositional Language and Elementary Visual Reasoning (CLEVR) dataset as a diagnostic dataset for evaluating the visual reasoning prospect of VQA systems where one can test the model's ability to answer questions about a diverse set of reasoning tasks (examples from CLEVR dataset is shown in Fig. 2.3). CLEVR dataset consists of rendered simple shapes, sizes and colors that reduce the complexity in recognition task and allow the VQA system to focus on reasoning tasks. The CLEVR diagnostic dataset has $100K$ rendered images and $\sim 1M$ automatically generated questions, $853K$ of which are unique. An example from Visual Genome dataset is presented in Fig. 2.4. This dataset features:

- **Simple scenes:** The rendered scenes were simple, consisting of three object shapes (cube, spare, cylinder), two object size (small and large), two materials

(shiny – 'metal' and matte – 'rubber) and eight colors (gray, red, blue, green, brown, purple, cyan, and yellow). The objects locations were defined by four spatial relations (left, right, in-front and behind) defined by projecting the camera view point vector into the scene.

- **Scene Graphs:** As the images were rendered, each image was accompanied by a ground truth scene graph. The scene-graph represented the nodes as objects with attribute annotations and edges as spatial relationships between objects.
- **Functional Program:** Each question is associated with a functional program to be executed on the image's scene graph to get the answer of the question. These functional programs represent elementary visual reasoning operations such as *querying* object attribute like color, size, shape, *counting* set of objects or *comparing* values and spatial locations.
- **Question Family:** The functional programs can have endless combinations for representing a question to achieve an answer. As the question size increases (average question size in CLEVR dataset is about 20 words), intuitively, the representation using the questions functional program becomes increasingly complex. To address this problem, the CLEVR dataset formulated 90 question families, each represented with a template with an average of 4 text templates per question family.

The questions presented in the CLEVR dataset are quite complex and require superior reasoning abilities which would require skills like counting, comparing and short term memory. This helps in figuring out which aspect of the VQA model needs improvement and how well it can reason about complex questions, but the scenes used to render the images are rather simple. Thus a model can improve its reasoning skills on a limited set of visual inputs, but it does not generalize well to the inherent complexities of the natural scenes of the real world. The authors also stress that improving accuracy on CLEVR dataset is not the end goal rather it should be used in conjunction with other VQA systems in order to evaluate the reasoning ability in question. Nonetheless, this dataset helped create the need of real image datasets with complex reasoning questions which we cannot always get from crowd-sourced human annotators.

2.1.6 GQA

The GQA dataset [Hudson and Manning, 2019] can be considered as the 'real image' version of the CLEVR dataset. The CLEVR dataset had rendered images and machine generated questions which limited its application and scope. However, its goal to test VQA models reasoning ability is still valid and GQA dataset paves the way towards that research direction by focusing on real-world reasoning and compositional question answering. This is the largest VQA dataset to-date with $113K$ images and $22M$ questions (examples from GQA dataset is shown in Fig. 2.5). Similar to CLEVR dataset, each image in GQA dataset has dense scene-graph annotation

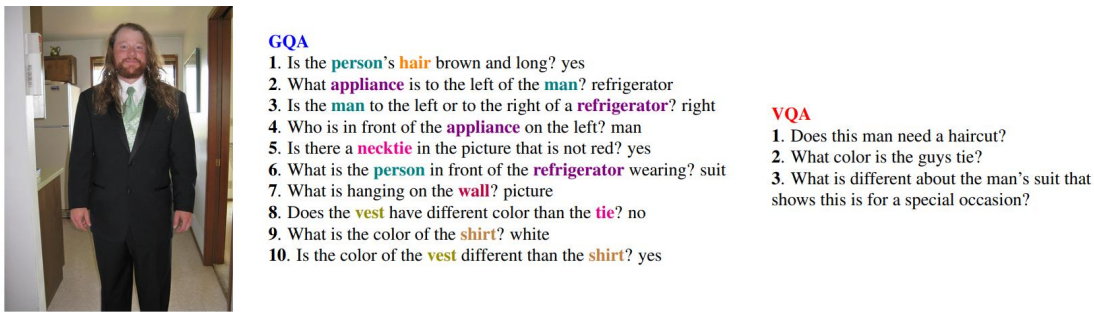


Figure 2.5: *Comparison of question-answer pair between VQA and GQA dataset.* The question-answer pairs in GQA dataset are more complex and require more visual and semantic concepts from image to predict the correct answer. Image adopted from Hudson and Manning [2019] .

describing the objects, attributes and relationships in the image, and each question is associated with a functional program that enumerates the reasoning steps required to answer the question. Unlike CLEVR dataset, each *answer* is accompanied with visual and linguistic justification regarding why the said answer is the correct one. The questions are generated in a two stage process, first, rich semantic information about the objects, relationships and attributes are sourced from rich scene graph, and second, structural patterns learned from VQA-v1 dataset Antol et al. [2015] are applied to generate questions that have natural diversity and compositionality of crowd sourced questions. To avoid question-conditional bias, question are curated based on the functional program associated with the question.

One shortcoming of the GQA dataset is its limited vocabulary size. The questions are partly generated from scene-graph, thus the question and the answers do not capture the diverse vocabulary of real world. There are only 3097 unique words in its $22M$ question. Such limited vocabulary does not scale well with the size of the dataset. Even with this shortcoming, GQA offers an exhaustive platform for evaluating VQA models.

2.1.7 VQA-CP

Both versions of the VQA dataset [Antol et al., 2015; Goyal et al., 2017] have problems with language priors. Agrawal et al. [2018] re-purposed both versions of VQA dataset to eliminate the effect of language prior, and proposed a new version for both of the VQA datasets called Visual Question Answering under Changing Priors (VQA-CP). **First,** the image-question-answer triplets of training and validation splits of VQA datasets are merged together. The ground truth annotations of the test split are not available, thus they are not included in the VQA-CP dataset. **Second,** the merged train-val dataset undergoes *question grouping* where similar questions are grouped together. The question similarity is determined if the first few words are similar and if they have the same answer. For example, the questions 'Q: What color is the

umbrella?, A: White' and 'Q: What color is the wall?, A: White' will be grouped together, whereas, 'Q: What color is the umbrella?, A: White' and 'Q: What color is the wall?, A: Rainbow' will be in separate groups. **Third,** the grouped questions are split into train and test sets following a *greedy re-splitting* approach. The image-question-answer triplets are distributed into the train and test sets so that question with the same type and ground truth are not common between the two sets with uniform coverage of the test concepts. This process is separately done for VQA-v1 dataset [Antol et al., 2015] and VQA-v2 dataset [Goyal et al., 2017] to generate VQA-CP v1 and VQA-CP v2 datasets. VQA-CP v1 dataset has a training set of 118K images with 245K questions and the test set consists of 87K images with 125K questions. On the other hand, VQA-CP v2 dataset has a training set of 121K images with 438K questions and the test set consists of 98K images with 220K questions.

The VQA-CP dataset sets up a challenging platform for evaluating VQA models. The train and test sets are semantically different so that VQA models cannot leverage the language bias. However, as the train and test separation is performed only based on questions types and answers; the same object can be present in train and test sets. For example, the image of a cat can exist in both the train and test sets even though they are separated by different question types. Further, as the questions are re-purposed from train-val split of the VQA datasets, the number of test questions are limited compared to other benchmark VQA datasets. Nonetheless, VQA-CP dataset offers a strong evaluation platform to check a VQA model's ability to perform without language bias.

2.1.8 Proposed Open-World VQA dataset

We summarize the comparison between existing VQA datasets in Tab. 2.1. It is evident from the review about the existing VQA datasets that in order to model real world ambiguities and commonsense knowledge, we need to build VQA datasets using natural images with crowd-sourced question-answer annotations. Further, we need to have non-overlapping training and testing splits, both considering semantic and visual inputs. Agrawal et al. [2018] proposed the VQA-CP which was only separated by semantic concepts, but not visual concepts. As VQA is a multimodal problem, it is only natural to have training and evaluation settings where one can evaluate a model's performance by training on known visual-semantic concepts, and evaluating on unknown concepts. This measures a VQA model's ability to infer about the unknown by transferring knowledge and linking it with the known concepts. We propose an Open-World VQA dataset, dubbed OW-VQA, in Chapter 5 to provide such an evaluation platform. The main features of our proposed OW-VQA dataset are as follows (more details in Sec. 5.3):

- OW-VQA dataset has two versions, OW-VQA v1 and OW-VQA v2, created from VQA v1 and VQA v2 respectively. Our OW-VQA dataset consists of a total a 82,783 images and 224,040 questions in v1, and for the same number of images 402,961 questions in v2.

Dataset ↓	Statistics		Source		Major Limitation
	Image	I-Q Pair	Image	Question-Answer	
DAQUAR	1,449	12,468	NYUVv2	Human+Algo.	Scene, size and evaluation metric
COCO-QA	123,287	117,684	COCO	Algorithm	Algorithmic question generation
VQA-v1	204,721	614,163	COCO	Human	Language bias
VQA-v2	204,721	1,132,904	COCO	Human	Train-test distribution bias
VG	108,000	1,773,258	COCO+YFCC	Human	No binary questions
CLEVR	100,000	864,968	Rendered	Algorithm	Artificial scene
GQA	$\sim$ 113,000	$\sim$ 22,000,000	COCO+VG	Algorithm	Vocabulary size
VQA-CP v1	$\sim$ 205,000	$\sim$ 370,000	VQA-v1	VQA-v1	Splits based only on semantic separation
VQA-CP v2	$\sim$ 219,000	$\sim$ 658,000	VQA-v2	VQA-v2	

Table 2.1: *Comparison between existing VQA datasets.* In this table, we compare the existing VQA datasets based on dataset statistics (i.e., total number of images and associated questions), image source (i.e., natural image dataset or rendered image), question-answer source (i.e., if human annotators were employed or NLP based algorithms were used), and major limitation of each dataset.

- We provide the four splits for both versions of our OW-VQA dataset. The *trainset* split consists only of known visual-semantic concepts, the *valset-known* split consists of known concepts and the *valset-unknown* split consists of unknown concepts of the validation set, and the *testset* split consists of subset of all unknown visual-semantic concepts.
- We recommend two evaluation protocols for our dataset. **First,** for ablation and fine-tuning, we recommend to train a VQA model on the respective *trainset* and evaluate on *valset-known* and *valset-unknown*. This unique feature provides insight about how VQA models can reason about both known and unknown concepts. **Second,** for benchmark VQA accuracy, we recommend training on *trainset* + *valset-known* and report the accuracy when evaluated on *testset*.

2.2 Attention in VQA models

2.2.1 Image-level attention

Given the success of deep learning, image-level visual features from images are extracted using deep Convolutional Neural Networks (CNNs) (e.g., VGGNet [Simonyan and Zisserman, 2014], ResNet [He et al., 2016]), and semantic features are extracted using word-embeddings [Pennington et al., 2014] or Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997] networks or its modern variants [Chung et al., 2014]. Once the features are extracted from the input, the VQA model needs to identify which part of the image is most important to answer the given question. To do so, some approaches train RNNs to generate top-K candidate answers and use a multi-class classifier to choose the best answer [Antol et al., 2015; Zhu et al., 2016; Zhou et al., 2015]. For an improved VQA capability, attention mechanisms have been focused on either or both the image [Shih et al., 2016] and the natural language questions [Lu et al., 2016b]. A number of attention mechanisms have been incorporated within deep networks to automatically focus on arbitrary regions in the image based on the given question [Lu et al., 2016b; Jabri et al., 2016]. One of such attention model is stacked attention network [Yang et al., 2016] which generates an attention map on the image by recursively attending to salient image details. Memory networks have also been incorporated in many top performing models [Xiong et al., 2016; Sukhbaatar et al., 2015; Zhu et al., 2017] where the questions required the system to compare attributes or use a long reasoning chain. All of these attention models try to learn a spatial attention distribution over the image using image-level visual features [Xu et al., 2015; Yang et al., 2016; Jabri et al., 2016; Shih et al., 2016; Lu et al., 2016b].

2.2.2 Object-level attention

Different multimodal fusion methods [Xu et al., 2015; Gao et al., 2016; Ben-Younes et al., 2017] have been used to compute the attention maps (called soft attention) on the spatial image grid locations. All these techniques explore top-down attention and

only focus on the image grid. Anderson et al. [2018] used object detection algorithms (i.e., Faster R-CNN [Ren et al., 2015b]) to get visual feature of the top object proposals of an image. Using a set of object-level features instead of grid-level features provides bottom up attention by identifying all the objects in the image. Using this additional bottom up attention provides accuracy boost for VQA models [Anderson et al., 2018; Ben-Younes et al., 2019]. The feature maps generated from the object proposals are discrete and do not encode the spatial relationships between the objects present in the image. Thus, there exists a semantic gap since the two sets of approaches look at different kinds of features, one from the image-level and one from the object-level.

2.2.3 Self Attention

A large portion of the VQA literature focuses on learning a multi-modal representation of image and question features to generate an attention distribution over the input visual feature representation [Fukui et al., 2016; Vinyals et al., 2015; Ben-Younes et al., 2017, 2019; Farazi and Khan, 2018; Yu et al., 2017; Kim et al., 2018]. These approaches have been very successful in learning the multi-modal interactions, however they do not learn mono-modal attention distributions over the inputs themselves e.g., identifying correlation between different image regions or relationships between different words of the question. Inspired by the success of self-attention mechanism [Vaswani et al., 2017] in capturing long range dependencies, Yu et al. [2019] proposed to use self-attention to capture the mono-modal interactions in a VQA setting. Using such self attention, Yu et al. [2019] achieved better VQA accuracy. However, for achieving high-level visual understanding, one needs to learn both mono-modal and multi-modal interactions.

Inspired by human perception, in Chapter 3, we combine the image-level and object-level attention through an efficient attention mechanism to reason about the objects in local and global context, which enables that model not only to look at the whole image or only at objects, rather look at both. Further, as the visual linguistic attention models are limited to learning the attention distribution based on the questions in the dataset, in Chapter 4 we propose a question agnostic attention which helps answer questions that require the model to perform a reasoning task such as counting. Furthermore, in Chapter 6, we propose to use complementary mono-modal and multi-modal attention approaches to identify the salient visual and semantic cues by applying self and mutual attention.

2.3 Learning about the unknown

2.3.1 Training with Novel Concepts

A VQA engine is highly likely to encounter questions about completely unknown objects and semantic concepts when operating in an *Open-World* setting. Some recent attempts to propose novel concept splits for VQA only consider the language side [Agrawal et al., 2018; Teney and Hengel, 2016; Ramakrishnan et al., 2017; Agrawal

et al., 2017]. Goyal et al. [2017] showed that existing VQA datasets have highly correlated answers on train and test sets. As a result, VQA models tend to remember the popular answers instead of attending to the correct image details for predicting the correct answer. They subsequently proposed new protocols with distinct distributions of answers in both sets to have a fair evaluation protocol. On similar lines, Agrawal et al. [2018] proposed a new split for VQA where train and test sets have different prior distributions for each question type. Teney and Hengel [2016] also highlighted that current VQA models are biased towards rare and unseen concepts and proposed a zero-shot split only for language content (i.e., Q&A). We note that the above-mentioned methods only suggest a language based split and the visual concepts may still appear visually during the training process. Therefore, they do not satisfy the *Open-World* assumption.

2.3.2 Incorporating supplementary information

Although most VQA approaches only work with a given training set, some efforts explore the use of supplementary information to help the VQA system. Generally, such methods employ external knowledge sources (both textual and visual) to augment the training set. For example, a couple of approaches used web searches to find related images which were used for answer prediction [Teney and Hengel, 2016; Teney et al., 2018]. Language based external knowledge bases were used by Wang et al. [2017a] and Wu et al. [2016] to provide logical reasons for each answer choice and to answer a more diverse set of questions. More recently, Teney and van den Hengel [2018] proposed a meta-learning approach that learns to use an externally supplied support set comprising of example question-answers. Patro and Namboodiri [2018] proposed a differential attention mechanism that uses an exemplar from the training set to generate human-like attention maps, however does not consider a transferable attention function that can reason about new visual/semantic concepts.

In contrast to these approaches, we develop a VQA model for the open-world in Chapter 5 that does not use any external data, rather learns an attention function to use similar examples from the training set to provide better inference-time predictions. The VQA model first identifies similar known visual and semantic concepts from the training set and transfers the leaned joint feature embeddings to reason about the unknown concept. Our approach does not require external knowledge base and/or expensive pretraining, rather employs an efficient search and retrieval on its training states, which makes it generalizable across datasets.

2.4 Understanding visual relationships

2.4.1 Visual relationships

To improve visual reasoning performance, understanding object relationships in a scene is important. The concept of scene-graph was introduced in [Johnson et al., 2015] for image retrieval, where it was defined as a way of describing the contents of

a scene by encoding object instances, attributes of objects, and relationships between objects. In the VQA domain, Visual Genome [Krishna et al., 2016] and CLEVR [Johnson et al., 2016] datasets contain scene-graphs representing the relationships between objects and attributes, and some object clusters that usually come together. These representations are often generated manually by humans [Krishna et al., 2016] for real images and automatically for rendered images [Johnson et al., 2016]. However, a recent approach has been proposed [Xu et al., 2017] to generate such scene-graphs automatically. This approach uses state-of-the-art object detection algorithms (Faster R-CNN [Ren et al., 2015b]) to detect the objects in an image; rather than predicting local relationship predicates among the objects, it passes messages between different regions of the image to capture the global scene context. Additional knowledge sources, such as language priors [Lu et al., 2016a] and pairs of images [Zhang et al., 2017a] have also been used to predict object relationships. Different from our work, top-down visual factors and associated object features have not been used for the VQA task, where such information is highly desirable to generate an informed answer.

2.4.2 Semantic relationship modelling

The two major obstacles in utilizing visual relationships in a VQA model are the lack of ground-truth relationship labels and a way to represent the relationship features.

Several recent VQA models [Xu et al., 2017; Li et al., 2019b; Hu et al., 2019; Zhang et al., 2019a] resorted to a graph neural network approach where the object pairs representing the nodes and relationship features were represented by some combination of the object features. This approach has two practical limitations, **first,** it relies heavily on the graph representation and the model's ability to reason over the graph representation. **Second,** the lack of a real-world VQA dataset that has ground-truth graph representations of images to train and test the models. A few models [Teney et al., 2016; Santoro et al., 2017] tried to capture the relationships from rendered synthetic VQA datasets (Abstract Scene VQAv1 [Antol et al., 2015], CLEVR [Johnson et al., 2016]), which do not generalize well to real scenes. Even though, the Visual Genome [Krishna et al., 2016] dataset has scene graph annotations, the lack of scene graph representations in benchmark VQA datasets (e.g., VQAv1[Antol et al., 2015], VQAv2 [Goyal et al., 2017], VQA-CP[Agrawal et al., 2018]) limit a model's ability to generate graph representations.

We propose a tangential approach in Chapter 6, first of its kind, where we treat the visual relationship feature not as a combination of visual features or a graph, rather as a semantic mono-modal feature representation from its subject, predicate and object labels. We use semantic relationship parser that generates relationship triplets directly from the image, and use them along with other input visual features. Such enriched semantic relationship modeling is an important missing piece in the existing VQA models. We demonstrate that under an oracle setting, these semantic relationships can bring the performance on par with the human-level accuracy for the VQA task.

2.5 Conclusion

In summary, this chapter provides an overview of existing VQA literature and identifies knowledge gaps around existing datasets, visual attention, reasoning about the unknown and understanding visual relationships. In the next chapters, we discuss how we developed and improved VQA state-of-the-art by bridging the aforementioned knowledge gaps, and proposing better VQA models.

Chapter 3

Reciprocal Attention Fusion

In this chapter, we address the problem of attending to both image-level and object-level visual features for Visual Question Answering (VQA). Existing attention mechanisms either attend to local image-grid or object-level features. Motivated by the observation that questions can relate to both object instances and their parts, we propose a novel attention mechanism that jointly considers reciprocal relationships between the two levels of visual details. Our design hierarchically fuses multi-modal information i.e., language, object- and grid-level features, through an efficient tensor decomposition scheme. The bottom-up attention thus generated is further coalesced with the top-down information to only focus on the scene elements that are most relevant to a given question. This chapter is based on our published work [Farazi and Khan, 2018], previously mentioned in Sec. 1.6.

3.1 Introduction

An AI agent equipped with visual question answering ability can respond to intelligent questions about a complex scene. This task bridges the gap between visual and language understanding to realize the longstanding goal of highly intelligent machine vision systems. Recent advances in automatic feature learning with deep neural networks allow joint processing of both visual and language modalities in a unified framework, leading to significant improvements on the challenging VQA problem [Antol et al., 2015; Krishna et al., 2016; Johnson et al., 2016; Zhu et al., 2016; Goyal et al., 2017].

To deduce the correct answer, an AI agent needs to correlate image and question information. A predominant focus in the existing efforts has remained on attending to local regions on the image grid based on language input [Xu et al., 2015; Lu et al., 2016b; Yang et al., 2016; Jabri et al., 2016; Shih et al., 2016]. Since these regions do not necessarily correspond to representative scene elements (objects, attributes and actions), there exists a "semantic gap" in such attention mechanisms. To address this issue, Anderson et al. [2018] proposed to work at the object level, where model attention is spread over a set of possible object locations. However, the object proposal set considered in this way is non-exhaustive and can miss important aspects of a scene. Furthermore, language questions can pertain to local details about objects parts and

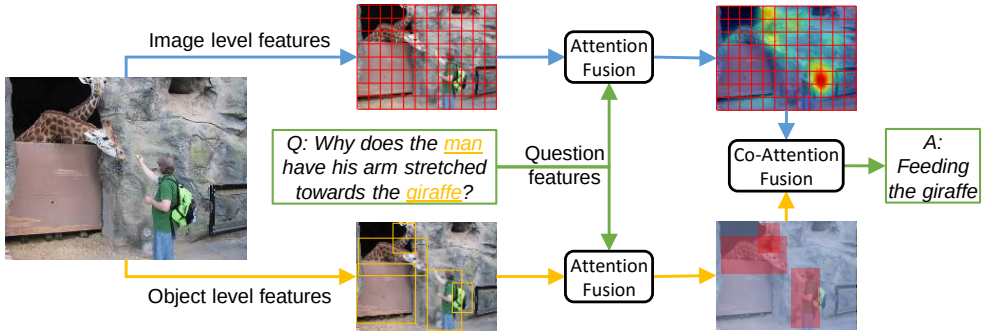


Figure 3.1: Applying attention to reciprocal visual features allow a VQA model to obtain the most relevant information required to answer a given visual question.

attributes, which are not encompassed by the object-level scene decomposition.

In this chapter, we propose to simultaneously attend to both low-level visual concepts as well as the high-level object based scene representation. Our intuition is based on the fact that the questions can be related to objects, object-parts and local attributes, therefore focusing on a single scene representation can degrade model capacity. To this end, we jointly attend to two reciprocal scene representations that encompass local information on the image gird and the object-level features. The bottom-up attention thus generated is further combined with the top-down attention driven by the linguistic input. Our design draws inspiration from the human cognitive psychology, where attention mechanism is known to be a combination of both exogenous (bottom-up) and endogenous (top-down) factors [Desimone and Duncan, 1995; Borji and Itti, 2013].

Given the multi-modal inputs, a critical requirement is to effectively model complex interactions between the multi-level bottom-up and top-down factors (Fig. 3.1). For this purpose, we propose a multi-branch CNN architecture that hierarchically fuses visual and linguistic features by leveraging an efficient tensor decomposition mechanism [Tucker, 1966; Ben-Younes et al., 2017]. Our experiments and extensive ablative study proves that a language driven attention on both image-grid and object level representation allows a deep network to model the complex interaction between vision and language as our model outperforms the state-of-the-art models in VQA tasks.

In summary, in this chapter we make the following key contributions:

- A hierarchical architecture incorporating both the bottom-up and top-down factors pertaining to meaningful scene elements and their parts.
- Co-attention mechanism enhancing scene understanding by combining local image-grid and object-level visual cues.
- Extensive evaluation and ablation on balanced and imbalanced versions of large-scale VQA dataset achieving single model state-of-the-art performance in both.

3.2 Methods

The VQA task requires an AI agent to generate a natural language response, given a visual (i.e., image, video) and natural language input (i.e., questions, parse). We formulate VQA task as a classification task, where the model predicts the correct answer ($\hat{a}$) from all possible answers for a given image ($\mathbf{v}$) and question ($\mathbf{q}$) pair:

$$\hat{a} = \underset{a \in A}{\operatorname{argmax}}\, p(a|\mathbf{v}, \mathbf{q}; \theta), \tag{3.1}$$

where θ denotes the set of parameters used to predict the best answer from the set of all possible answers A.

Our proposed architecture to perform VQA task is illustrated in Figure 3.2. The key highlights of our proposed architecture include a hierarchical attention mechanism that focuses on complementary levels of scene details i.e., gird of image regions and object proposals. The relevant co-attended features are then fused together to perform final prediction. We name our model as the *'Reciprocal Attention Fusion'* because it simultaneously attends to two complementary scene representations i.e., image gird and object proposals. Our experimental results demonstrate that both levels of scene details are reciprocal and reinforce each other to achieve the best single-model performance on challenging VQA task. Before elaborating on the hierarchical attention and feature fusion, we first discuss the joint feature embedding in Section 3.2.1.

3.2.1 Joint Feature Embedding

Let V be the collection of all visual features extracted from an image and Q be the language features extracted from the question. The objective of joint embedding is to learn the language feature representation $q = \chi(Q)$ and multilevel visual features $v_k = \zeta(V)$. These feature representations are used to encode the multilevel relationships between question and image which in turn is used to train the classifier to select the correct answer.

Multilevel visual features: The multilevel visual embedding v_k consists of image level features v_I and object level features v_O. Our model employs ResNeXt [Xie et al., 2017] to obtain image level features, $v_I \in \mathbb{R}^{n_v \times G}$ by taking the output of convolution layer before the final pooling layer, where G denotes the number of spatial grid locations of the extracted visual feature with n_v dimensions. This convolution layer retains the spatial information of the original image and enable the model to apply attention on the image grid. On the other hand, our model employs object detectors to localize object instances and pass them through another deep CNN to generate object level features $v_O \in \mathbb{R}^{n_v \times N}$ for N object proposals. We use Faster R-CNN [Ren et al., 2015b] with ResNet-101 [He et al., 2016] backbone and pretrain the object detector on ImageNet [Deng et al., 2009] and again retrain it on Visual Genome Dataset [Krishna et al., 2016] with class label and attribute features similar to Anderson et al. [2018].

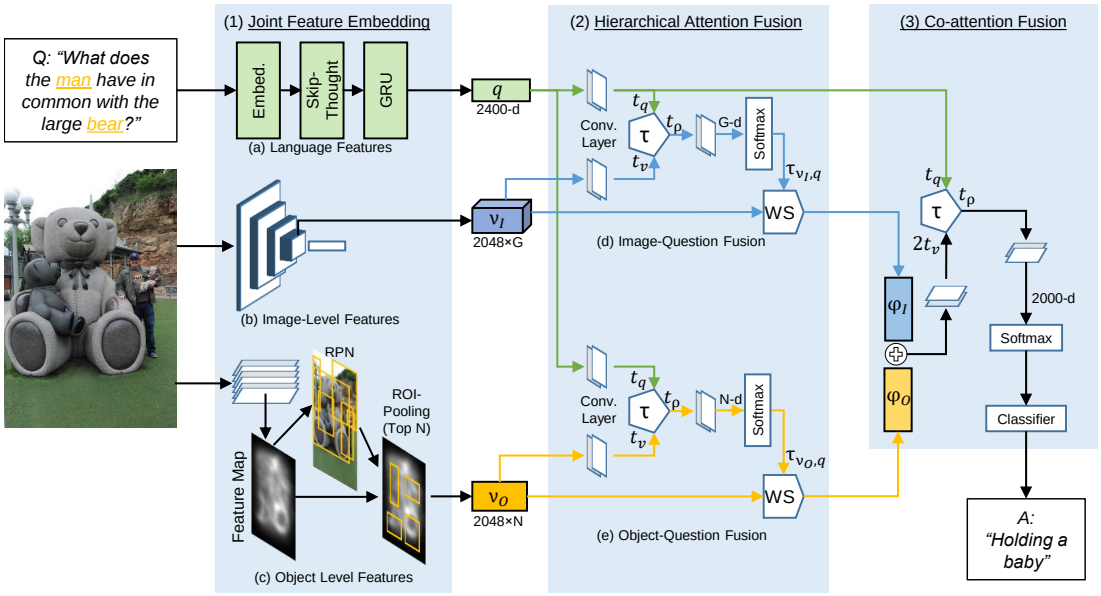


Figure 3.2: *VQA model architecture of Reciprocal Attention Fusion(RAF model).* Given an image-question pair, our model employs (1) Joint Feature Embedding (Sec.3.2.1) to embed (a) Language Feature q, (b) Image-Level Feature v_I and (c) Object-Level Feature v_O. Further, these embeddings undergo (2) Hierarchical Attention Fusion (Sec.3.2.2) which consists of (d) Image-Question and (e) Object-Question Fusion followed by top-down attention. These multi-modal representations are combined together by (3) Co-attention Fusion (Sec.3.2.3) that predicts an answer for the given Image-Question pair. Overall, the proposed model attends to complementary levels of scene details and fuses multi-modal information to predict highly accurate answers.

Bottom-up Attention: In order to focus on the most relevant features, two Bottom-up attention mechanisms are applied during multilevel feature extraction. The image-grid attention is generated using ResNeXt [Xie et al., 2017] pretrained on ImageNet [Deng et al., 2009] to obtain $v_I \in \mathbb{R}^{2048\times14\times14}$, which represents 2048 dimensional features vectors for $G = 14 \times 14$ image grid over the visual input. The size and scale of the image grid can be changed by using different CNN architecture or taking the output of a different convolutional layer to generate a different sized BU attention. Meanwhile, object proposals are generated in a bottom up fashion to encode object level visual features v_O. We select a total of top $N = 36$ object proposals whose $n_v = 2048$ dimensional feature vectors are obtained from the ROI pooling layer in the Region Proposal Network.

Language features: To represent the questions embedding in an end-to-end framework, GRUs [Cho et al., 2014] are used in a manner similar to [Fukui et al., 2016; Ben-Younes et al., 2017]. The words in questions are encoded using one-hot-vector representation and embedded into vector space by using a word embedding matrix. The embedded word vectors are fed to the GRU with n_q units initialized with

pretrained Skip-thought Vector model [Kiros et al., 2015]. The output of the GRU is fine-tuned to get the language feature embedding $q = \{q_i \ : \ i \in [1, n_q]\}$ where $n_q = 2400$. The language feature embedding is used to further refine the spatial visual features (i.e., image grid and object level) by incorporating top-down attention discussed in Section 3.2.2.

3.2.2 Hierarchical Attention Fusion

The hierarchical attention mechanism takes spatial visual features v_I, v_O and language feature q as input and a learns multi-modal representation W to predict answer embedding ρ. This step can be formulated as an outer product of the multi-modal representation, visual and language embeddings as follows:

$$\rho = W \times_1 \ q \ \times_2 \ v, \tag{3.2}$$

where, $\times_n$ denotes n-mode tensor-matrix product. However, this approach has some serious practical limitations in terms of learnable parameters for $W \in \mathbb{R}^{n_v \times n_q \times n_\rho}$ as the visual and language feature are very high dimensional, which results in huge computational and memory requirements. To counter this problem, our model employs a multi-modal fusion operation $\tau(v, q)$ to encode the relationships between these two modalities, which is discussed next.

Multi-modal Fusion: Multi-modal fusion aims to reduce the number of free parameters in tensor $W \in \mathbb{R}^{n_v \times n_q \times n_\rho}$ for a fully parameterized VQA bilinear model. Our model achieves this by using Tucker Decomposition [Tucker, 1966] which is a special case of higher-order principal component analysis to express W as a core tensor T_c multiplied by a matrix along input mode. The decomposed tensors are fused in a manner similar to [Ben-Younes et al., 2017] that encompass the multi-modal relationship between language and vision domain. The tensor W can be approximated as:

$$W \approx T_c \times_1 T_q \times_2 T_v \times_3 T_\rho = [\![T_c; T_q, T_v, T_\rho]\!] \tag{3.3}$$

where $T_v \in \mathbb{R}^{n_v \times t_v}$, $T_q \in \mathbb{R}^{n_q \times t_q}$ and $T_\rho \in \mathbb{R}^{n_\rho \times t_\rho}$ are factor matrices similar to principal components along each input and output embeddings and $T_c \in \mathbb{R}^{t_v \times t_q \times t_\rho}$ is the core tensor which encapsulates interactions between the factor matrices. The notation $[\![\cdot]\!]$ represents the shorthand for Tucker decomposition. In practice, the decomposed version of W is significantly smaller number of parameters than the original tensor [Bader and Kolda, 2007].

After reducing the parameter complexity of W with tucker decomposition, the fully parametrized outer product representation in Eq. 3.2 can be rewritten as:

$$\rho = T_c \ \times_1 \ \tilde{q} \ \times_2 \ \tilde{v} \ \times_3 \ T_\rho \tag{3.4}$$

where $\tilde{v} = v^\intercal \ T_v \in \mathbb{R}^{t_v}$ and $\tilde{q} = q^\intercal \ T_q \in \mathbb{R}^{t_q}$. We define a prediction space $\rho = \tau^\intercal \ T_\rho \in \mathbb{R}^{n_\rho}$ where the multi-modal fusion τ is:

$$\tau = T_c \ \times_1 \ \tilde{q} \ \times_2 \ \tilde{v} \ \in \mathbb{R}^{t_\rho} \tag{3.5}$$

The Tucker decomposition allows our model to decompose W into a core tensor T_c and three matrices. The first two matrices, T_q and T_v project the question and visual embeddings to lower t_q and t_v dimensional space that learns to model the multi-modal interaction and projects the resulting output to t_ρ dimensional vector. We set the input projections dimension to $t_q = t_v = 310$ and output projection dimension as $t_\rho = 510$. The input and output tensor projection dimensions determine the complexity of the model and the degree of multi-modal interaction which in turn affects the performance of the model. These values are set empirically by testing them on VQAv1 validation dataset. It has been reported in the literature [Fukui et al., 2016; Ben-Younes et al., 2017] that applying nonlinearity to the input feature embeddings improve performance of multi-modal fusion. Therefore, we encode $\tilde{v}$ and $\tilde{q}$ with tanh nonlinearity during fusion. The output of the multi-modal fusion $\tau \in \mathbb{R}^{t_\rho}$ passes through convolution and softmax layers to create $1 \times G$ and $1 \times N$ dimensional representation for image-question and object-question embedding respectively. Thus, by employing hierarchical attention fusion, we embed question with spatial visual features to generate image-question $\tau_{v_I,q} \in \mathbb{R}^{1\times G}$ and object-question $\tau_{v_O,q} \in \mathbb{R}^{1\times N}$ embedding.

Top-down Attention The image level and object level features are used alongside image-question and object-question embeddings to generate an attention distribution over spatial grid and object proposals respectively. We take weighted sum (WS) of the spatial visual features (i.e.,v_I and v_O) vectors using the attention weights (i.e.,$\tau_{v_I,q}$ and $\tau_{v_O,q}$) to generate φ_I and φ_O which are top-down attended visual features,

$$\varphi_I = \sum_i^G \tau_{v_I,q}^i \, v_I^i \quad \text{and} \quad \varphi_O = \sum_i^N \tau_{v_O,q}^i \, v_O^i. \tag{3.6}$$

3.2.3 Co-attention Fusion

The attended image-question and object-question visual features represent a combination of visual and language features that are most important to generate an answer for a given question. We concatenate these two bimodal representations to create the final visual-question embedding $\varphi = \varphi_I \oplus \varphi_O$. The visual-question embedding, φ and original question embedding q again undergo same multi-modal fusion as Eq.3.5. The only difference is now $t_\varphi = 2 \times t_v$ as our model uses two glimpse attention which was found to yield better results [Fukui et al., 2016; Ben-Younes et al., 2017; Kim et al., 2016]. The output of the final fusion is then passed on to the classifier that predicts the best answer $\hat{a}$ from the answer dictionary A given question **q** and visual input **v**.

Methods	Test-dev				Test-standard			
	Y/N	No.	Other	All	Y/N	No.	Other	All
RAF (Ours)	**85.9**	**41.3**	**58.7**	**68.0**	**85.8**	41.4	58.9	**68.2**
ReasonNet[Ilievski and Feng, 2017]	-	-	-	-	84.0	38.7	**60.4**	67.9
MFB+CoAtt+Glove [Yu et al., 2018]	85.0	39.7	57.4	66.8	85.0	39.5	57.4	66.9
Dual-MFA [Lu et al., 2017]	83.6	40.2	56.8	66.0	83.4	40.4	56.9	66.1
MLB+VG [Kim et al., 2016]	84.1	38.0	54.9	65.8	-	-	-	-
MCB+Att+GloVe [Fukui et al., 2016]	82.3	37.2	57.4	65.4	-	-	-	-
MLAN [Yu et al., 2017]	81.8	41.2	56.7	65.3	81.3	**41.9**	56.5	65.2
MUTAN [Ben-Younes et al., 2017][1]	84.8	37.7	54.9	65.2	-	-	-	-
DAN (ResNet) [Nam et al., 2016]	83.0	39.1	53.9	64.3	82.8	38.1	54.0	64.2
HieCoAtt [Lu et al., 2016b]	79.7	38.7	51.7	61.8	-	-	-	62.1
A+C+K+LSTM[Wu et al., 2016]	81.0	38.4	45.2	59.2	81.1	37.1	45.8	59.4
VQA LSTM Q+I [Antol et al., 2015]	80.5	36.8	43.1	57.8	80.6	36.5	43.7	58.2
SAN[Yang et al., 2016]	79.3	36.6	46.1	58.7	-	-	-	58.9
AYN [Malinowski et al., 2017]	78.4	36.4	46.3	58.4	78.2	37.1	45.8	59.4
NMN [Andreas et al., 2016]	81.2	38.0	44.0	58.6	-	-	-	58.7
DMN+ [Xiong et al., 2016]	60.3	80.5	48.3	56.8	-	-	-	60.4
iBOWIMG [Zhou et al., 2015]	76.5	35.0	42.6	55.7	76.8	35.0	42.6	55.9

Table 3.1: Comparison of the state-of-the-art methods with our single model performance on VQAv1.0 test-dev and test-standard server. All models reported in this comparison use ResNet [He et al., 2016] (except for iBOWIMG [Zhou et al., 2015] which uses GoogleLeNet [Szegedy et al., 2015]) to extract image-level visual features.

Methods	Test-dev				Test-standard			
	Y/N	No.	Other	All	Y/N	No.	Other	All
RAF (Ours)	**84.1**	**44.9**	**57.8**	**67.2**	**84.2**	**44.4**	**58.0**	**67.4**
BU, adaptive K [Teney et al., 2018]	-	-	-	-	81.8	44.2	56.0	65.3
MFB [Yu et al., 2018]	-	-	-	64.9	-	-	-	-
ResonNet[Ilievski and Feng, 2017]	-	-	-	-	78.9	42.0	57.4	64.6
MUTAN[Ben-Younes et al., 2017]†	80.7	39.4	53.7	63.2	80.9	38.6	54.0	63.5
MCB [Fukui et al., 2016; Goyal et al., 2017]	-	-	-	-	77.4	36.7	51.2	59.1
HieCoAtt [Lu et al., 2016b; Goyal et al., 2017]	-	-	-	-	71.8	36.5	46.3	54.6
Language only[Goyal et al., 2017]	-	-	-	-	67.1	31.6	27.4	44.3
Common answer[Goyal et al., 2017]	-	-	-	-	61.2	0.4	1.8	26.0

Table 3.2: Comparison of the state-of-the-art methods with our single model performance on VQAv2.0 test-dev and test-standard server. † Performance on VQAv2 is evaluated from their publicly available repository. All models reported in this comparison use ResNet [He et al., 2016] to extract image-level visual features.

3.3 Experiments

3.3.1 Dataset

We perform experiments on **VQAv1** [Antol et al., 2015] and **VQAv2** [Goyal et al., 2017] both of which are large scale VQA datasets. VQAv1 contains over 200K images from the COCO dataset with 610K natural language open-ended questions. VQAv2 [Goyal et al., 2017] contains almost twice as many question for the same number of images. VQAv2 has a balanced image-question pair to mitigate the language bias that allows a more realistic evaluation protocol. **Visual Genome** is another larger scale dataset that has image question pair with dense annotation of objects, attributes [Krishna et al., 2016]. We train a pretrained faster RCNN model (on ImageNet) again on Visual Genome dataset with class and attribute labels to extract object level features from the input image.

3.3.2 VQA Model Architecture

Question Feature Embedding: Our model embeds the question features by first generating the questions and answer dictionary from training and validation set of the VQA datasets. We make the question and answers lower case, remove punctuation and perform other standard preprocessing steps before tokenizing the words, and representing them into one-hot vector representation. As mentioned in section 3.2.1, these question embeddings are fed to GRUs pretrained with Skip-thoughts [Kiros et al., 2015] model that generates 2400-d language feature embeddings for the given question. When experimenting with VQAv1 and VQAv2, we parse questions respectively from training and validation sets to create the question vocabulary.

Answer Encoding: We formulate the VQA task as a classification task. We create an answer dictionary from the training data and select the top 2000 answers as the different classes. We pass the output of the final fusion layer through a convolutional layer that outputs a 2000d vector. This vector is passed through the classifier to predict $\hat{a}$.

We use Adam solver [Kingma and Ba, 2014] with base learning rate of 10^{-4} and batch size of 512 for our experiments. We keep the training parameters same for all our experiments. We use NVidia Tesla P100 (SXM2) GPUs to train our models and report our experimentation results on VQAv1 [Antol et al., 2015] and VQAv2 [Goyal et al., 2017] dataset representing 1500 GPU hours of computation.

3.4 Results

We evaluate our performance on the VQA test servers which ensures blind evaluation on the VQAv1 [Antol et al., 2015] and v2 [Goyal et al., 2017] test sets (i.e., test-dev, test-standard) following the VQA benchmark evaluation approach. The accuracy y

[1]Single model performance is evaluated using their publicly available code.

of the predicted answer $\hat{a}$ is calculated with the following formulation:

$$y = min\left(\frac{\text{\# of humans answered } \hat{a}}{3}, 1\right) \tag{3.7}$$

which means that answer provided by the model is given 100% accuracy if at least 3 human annotators who helped create the VQA dataset gave the exact answer.

In Table 3.1, we report VQAv1 test-dev and test-standard accuracies for our proposed RAF model and compare it with other single models found in literature. Remarkably, our model outperforms all other models in the over all accuracy. We report a significant performance boost of 1.2% on the test-dev set and 0.3% on the test-standard set. It is to be noted that using multiple ensembles and data augmentation with complementary training in Visual Genome QA pairs can increase the accuracy performance of the VQA models. For instance, MCB [Fukui et al., 2016], MLB [Kim et al., 2016], MUTAN [Ben-Younes et al., 2017] and MFB [Yu et al., 2018] employ similar model ensemble consisting of 7,7,5 and 7 models respectively, and report overall 66.5, 66.9, 67.4 and 69.2 on the test-standard set. It is interesting to note that except for MFB (7) all other ensemble models are $\sim$ 1% less than our reported single model performance. We do not ensemble our model or use data augmentation with complementary dataset as it makes the best results irreproducible and most of the models in the literature do not adopt this strategy.

We also evaluate our model on VQAv2 test-standard dataset and compare it with state-of-the-art single model performance in Tab.3.2, illustrating that our model surpasses the closest method [Teney et al., 2018] in all question categories and overall by a significant margin of 2.1%. The bottom up, adaptive-k[Teney et al., 2018] is the same model whose 30-ensemble version [Anderson et al., 2018] reports currently the best performance among on VQAv2 test-standard dataset. This indicates our models superior capability to interpret and incorporate multi-modal relationships for visual reasoning.

In summary, our model achieves state-of-the-art performance on both VQAv1 and VQAv2 dataset which affirms the robustness of our model against language bias without the need of data augmentation or the use of ensemble model. We also show qualitative results in Fig. 3.4 to demonstrate the efficacy and complimentary nature of attention focused on image-grid and object proposals.

3.4.1 Ablation Study

We perform an extensive ablation study of the proposed model on VQAv2 [Goyal et al., 2017] validation dataset and compare it with the best performing model in Table 3.3. This ablation study helps to better understand the contribution of different components of our model towards the overall performance on the VQA task. The objective is to show that when the language features are combined with image grid and object-level visual features, the accuracy of the high-level visual reasoning task (i.e., VQA) increases in contrast to only combining language with image or object-level features. The models reported in Category I in Table 3.3 use only

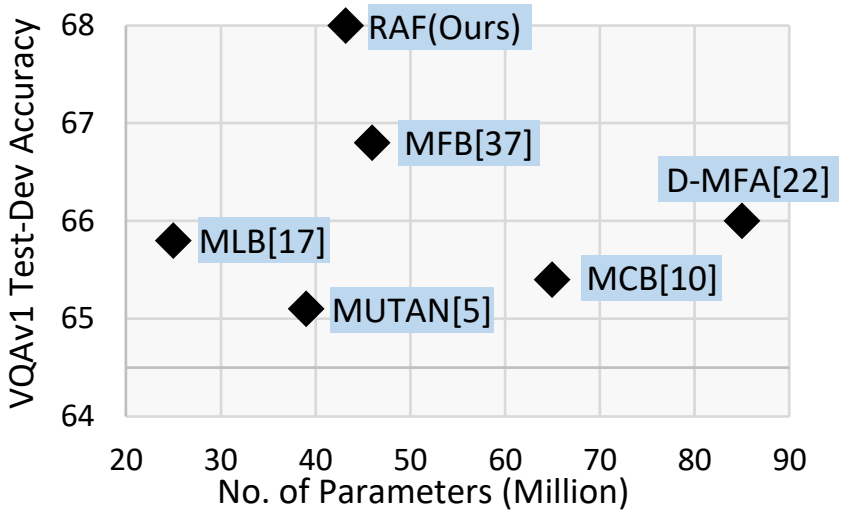


Figure 3.3: *Comparison of Accuracy vs No. of Parameters with other bilinear models.*

image-level features extracted with deep CNNs and we compare RAF-I which is a variant of our proposed RAF architecture only using image-level features. We observe RAF-I achieve comparable performance in this category. In Category II, RAF-O model extracts only 36 object-level features but outperforms the models in Category I. Anderson et al. [2018] also used only object-level features and this variant of our model achieves comparable performance to that model. When we combine image and object-level features together in Category III, we observe that the best results are obtained. This proves our hypothesis that the questions relate to both objects, object parts and local attributes, which should be attended for jointly an improved VQA performance.

The recent Dual-MFA [Lu et al., 2017] model also uses complementary image and object-level features. In contrast, our model uses bimodal attention fusion and also evaluates on the balanced VQAv2 [Goyal et al., 2017] on which they do not report performance. We also study the accuracy vs. size (no. of parameters) trade off in Fig. 3.3 on VQAv1 test-dev set as most of the bilinear models do not report on VQAv2. Remarkably, our RAF model achieves significant performance boost over Dual-MFA (66% to 68%) with around half the number of parameters.

3.5 Conclusion

We build our proposed model based on the hypotheses that multi-level visual features and associated attention can provide an AI agent additional information pertinent for deep visual understanding. As VQA is a standard measure of image understanding and visual reasoning, we propose a VQA model that learns to capture the bimodal feature representation from visual and language domain. To this end,

Cat.	Methods	Val-set
I	RAF-I(ResNet)	53.9
	HieCoAtt [Lu et al., 2016b; Goyal et al., 2017]	54.6
	RAF-I(ResNeXt)	58.0
	MCB [Fukui et al., 2016; Goyal et al., 2017]	59.1
	MUTAN [Ben-Younes et al., 2017]	60.1
II	Up-Down[Anderson et al., 2018]	63.2
	RAF-O(ResNet101)	63.9
	RAF-O(ResNet152)	63.4
III	RAF-IO(ResNet-ResNet)	64.0
	RAF-IO(ResNeXt-ResNet)	64.2

Table 3.3: Ablation Study on VQAv2 [Goyal et al., 2017] validation set.

we employ state of the art CNN architectures to obtain visual features for local regions on the image grid and object proposals. Based on these feature encodings, we develop a hierarchical co-attention scheme that learns the mutual relationships between objects, object-parts and given questions to predict the best response. We validate our hypotheses by evaluating the proposed model on two large scale VQA dataset servers followed by an extensive ablation study reporting state-of-the art performance. Our model improves the state-of-the-art single model performances from 67.9% to 68.2% on VQAv1 and from 65.7% to 67.4% on VQAv2, demonstrating a significant boost.

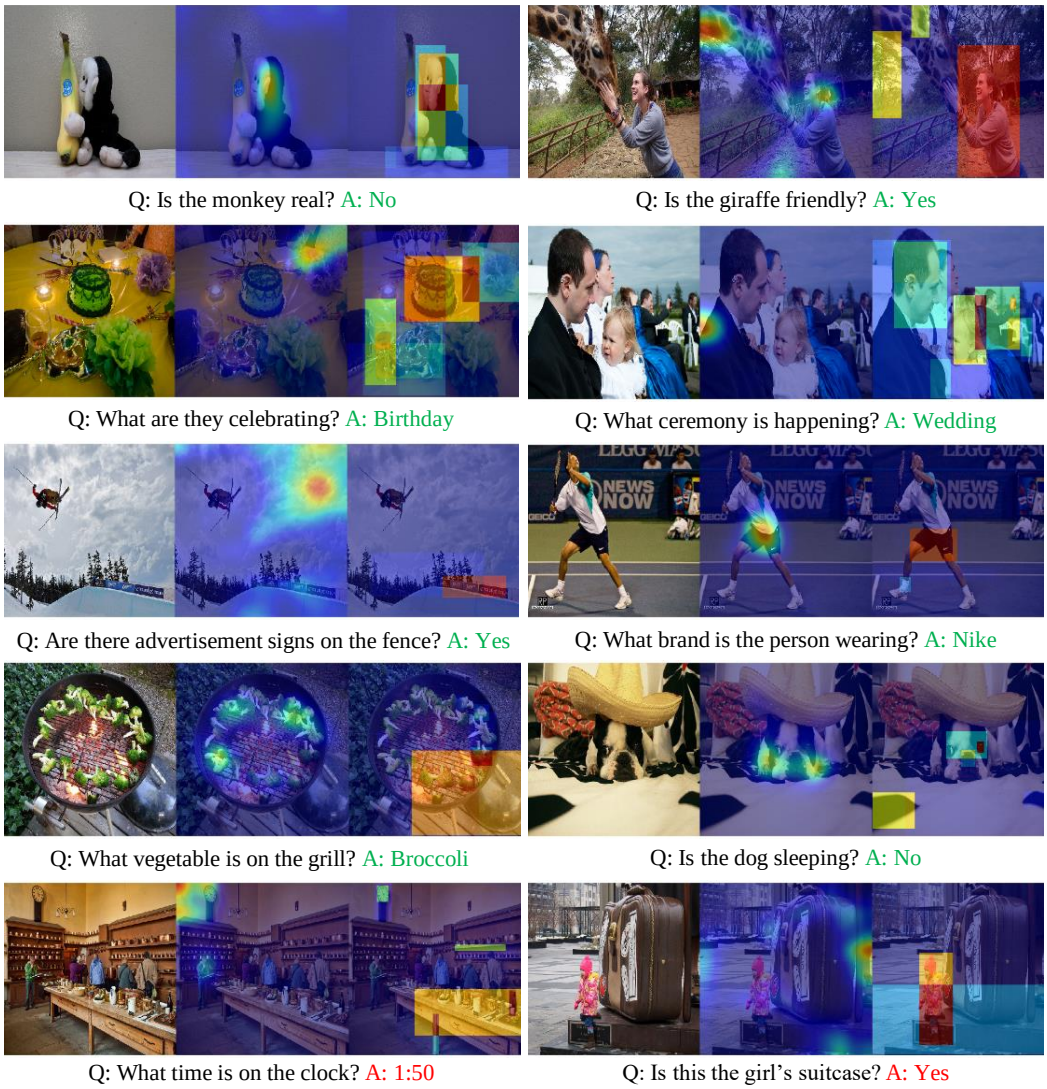


Figure 3.4: *Qualitative results of the proposed Reciprocal Attention Fusion mechanism for Visual Question Answering.* Given a question and an image (columns: 1, 4), attention based on image-grid (columns: 2, 5) and object proposals (columns: 3, 6) is shown above. Correct and incorrect answers are shown in green and red, respectively. Remarkably, the two attention levels provide complementary information about localized regions and objects that in turn help in obtaining the correct answer (rows: 1, 2, 3, 4). In some failure cases of our technique, ambiguous attention maps lead to incorrect predictions (row: 5).

Chapter 4

Question Agnostic Attention

In this chapter, we propose an attention mechanism that is complementary to any existing question-dependent attention mechanisms of a Visual Question Answering (VQA) model. VQA models employ attention mechanisms to discover image locations that are most relevant for answering a specific question. For this purpose, several multimodal fusion strategies have been proposed, ranging from relatively simple operations to more complex ones. The resulting multimodal representations define an intermediate feature space for capturing the interplay between visual and semantic features, that is helpful in selectively focusing on image content. Our model parses object instances to obtain an 'object map' and applies it on the visual features to generate QAA features. In contrast to question-dependent attention approaches that are learned end-to-end, the proposed QAA does not involve question-specific training, and can be easily included in almost any existing VQA model as a generic light-weight pre-processing step, thereby adding minimal computation overhead for training. Further, when used in complement with the question-dependent attention, QAA allows the model to focus on regions containing objects that can potentially be overlooked by the learned attention. Through extensive evaluation on VQAv1, VQAv2 and TDIUC datasets, we show that incorporating complementary QAA allows state-of-the-art VQA models to perform better, and provides significant boost to simplistic VQA models, enabling them to perform on par with highly sophisticated fusion strategies. This chapter is based on our published work [Farazi et al., 2020d], previously mentioned in Sec. 1.6.

4.1 Introduction

An attention mechanism in a VQA system identifies the relevant visual information to intelligently answer a given question. Therefore, attention is central to recent state-of-the-art VQA models. Existing VQA models generally use grid-level or object-level convolutional features to *learn* an attention distribution over the given image. In the former case, this attention is dispersed over the spatial grid [Antol et al., 2015; Fukui et al., 2016; Yang et al., 2016] while in the later case, attention is applied on a set of object proposals [Anderson et al., 2018; Ben-Younes et al., 2019]. Recent best performing methods combine the strengths of both these approaches to obtain

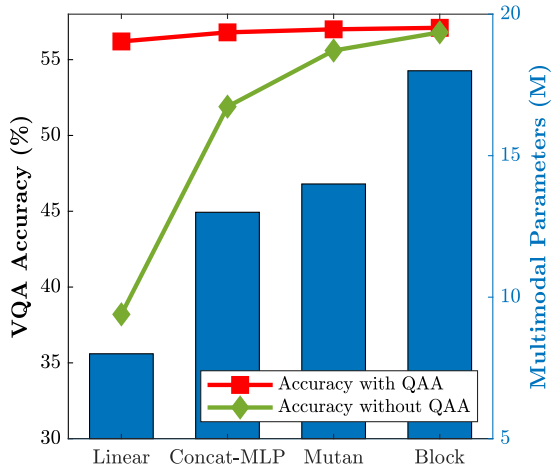


Figure 4.1: *A comparison of various multimodal fusion schemes for VQA evaluated on VQAv2 validation dataset.* In general, methods with low-parametric complexity (such as linear sum, concatenation followed by MLP) deliver low performance compared to more sophisticated ones (e.g., Mutan [Ben-Younes et al., 2017], Block [Ben-Younes et al., 2019]). Using our proposed Question-Agnostic Attention, we observe a consistent boost for all fusion mechanisms. The improvement is especially more pronounced for simple models, bringing them on par with highly sophisticated methods.

attention maps with a better context [Farazi and Khan, 2018; Lu et al., 2018].

Learned attention mechanisms have been shown to significantly enhance the performance of VQA systems. However, learning attention on dense grid- and object-level features is a computationally demanding task that results in increased model complexity. Furthermore, learned attention is tuned for a specific dataset and thereby fails to generalize well to novel scenarios. To address these problems, we undertake a tangential path and propose a Question-Agnostic Attention (QAA) approach that is independent of a given question. Our approach is based on the insight that questions generally relate to the state, number, type and actions of the '*objects*' present in an image and their '*mutual relationships*'. Therefore, we propose to use an object parsing module to generate question-agnostic attention maps based only on the given images. This attention generation procedure acts as a simple pre-processing step that encodes salient instance-centric visual cues (e.g., location, shape) and object-relationship information which in turn leads to a performance boost for all evaluated models and difficult question types (e.g., *'What sports are they playing?'*, *'What kind of furniture is in the picture?'*)

We note that our proposed question-agnostic attention has some resemblance to bottom-up saliency based attention methods. The saliency maps can be obtained from human-viewers or predicted using learned machine learning models. The literature shows that human eye fixations as an attention mechanism, works poorly for VQA and has less correlation with machine attention [Das et al., 2017a]. On the other hand, several efforts in VQA literature show the importance of object-aware visual

attention for improved VQA [Lu et al., 2016b; Anderson et al., 2018; Farazi and Khan, 2018] which emphasizes the notion that better localization of object instances results in higher VQA accuracy. However, these attention procedures are learned on top of object proposals while we propose an attention approach with minimal training cost. Our approach uses instance segmentation to generate an *object map* on the spatial image grid in a bottom-up fashion that is demonstrated to improve performance for simple as well as complex VQA models.

Our results provide an interesting perspective on VQA showing that question-agnostic attention can help achieve competitive VQA performance and provides complementary information for existing VQA models, that results in notable performance gain. In an extreme case, when we apply a fixed attention map computed from a prior based on the training data, the VQA model still performs on par with existing models with learned attention (Fig. 4.1). *Firstly,* this highlights the performance-complexity trade-off that is offered by recent multimodal fusion mechanisms for VQA task. Our results show that even with very simple multimodal operations, a VQA model can perform as well as more sophisticated models if question-agnostic attention is used. *Secondly,* the performance improvement across all the models illustrates the complementary nature of QAA, that highlights the room for improvement in learned 'question-aware' attention. *Finally,* the relatively stronger improvement for simpler models shows that the information learned with QAA features is somewhat similar to the high-order representation modelling through complex multimodal fusion techniques.

The main contributions of this chapter include:

- An inexpensive VQA pre-possessing step, dubbed Question-Agnostic Attention (QAA), that localizes object instances in an image irrespective of the question.
- A modular co-attention architecture that allows any off-the-shelf VQA model to incorporate complementary QAA features.
- An extensive set of experiments on large scale VQA datasets and the TDIUC dataset to showcase the effectiveness of using complementary QAA features, especially helping simplistic VQA models achieve near state-of-the art performance.

4.2 Method

Given a question Q about an image I, an AI agent designed for the VQA task will predict an answer a^* based on the learning acquired from training examples. Benchmark VQA models frame this task as a multi-class classification problem in the candidate answer space, and the models learn to predict the correct answer for a given Image-Question (IQ) pair. This task can be formulated as:

$$a^* = \underset{\hat{a} \in \mathcal{D}}{\arg\max}\, P(\hat{a}|Q, I; \theta), \tag{4.1}$$

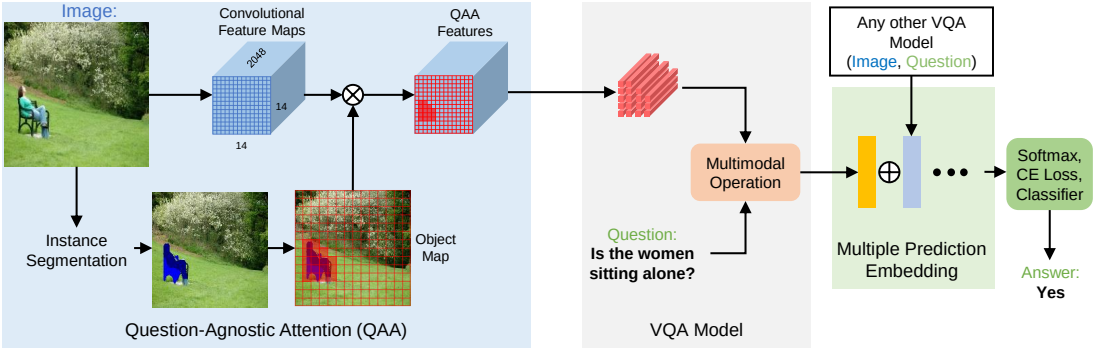


Figure 4.2: *Architecture of our Question-Agnostic Attention (QAA) based VQA model.* QAA features are generated using instance segmentation (using Mask-RCNN) to create a binary *object map* with the same resolution of the convolutional feature map. The *object map* is applied as a mask on the convolutional feature map (generated by ResNet) of the whole image. This 'modular attention' with minimal training cost delivers strong improvement while used in complement with existing VQA models on a number of VQA benchmarks.

where θ denotes the model parameters and a^* is predicted from the dictionary of candidate answers $\mathcal{D}$.

A simplistic VQA model consists of two major parts: (1) Feature extraction module, and (2) Multimodal feature embedding. The **first** part of the model extracts visual features from an Image I and semantic features from a Question Q. The visual features from an image are extracted using deep CNN based object recognition models (e.g., ResNet [He et al., 2016]) which are pretrained on large-scale image recognition datasets such as ImageNet [Deng et al., 2009]. The image feature map from the last convolution layer of the model is extracted as the visual feature $\boldsymbol{v} \in \mathbb{R}^{g \times d_v}$, where g is the index of the spatial location in the image over a coarse grid and d_v is the feature embedding dimension for each spatial grid location. On the other hand, for extracting the language feature from a Question, each word is fed to a pretrained encoder (e.g., GloVe [Pennington et al., 2014], Skip-thought [Kiros et al., 2015]) to get vector embeddings of the question words. These vectors are then passed through a language model which consists of Gated Recurrent Units (GRUs) to generate a semantic feature $\boldsymbol{q} \in \mathbb{R}^{d_q}$.

In the **second** part, extracted visual and semantic features are combined into a multimodal representation, which in turn is used to minimize a loss function to predict the correct answer. A VQA model employs a joint embedding function $\Psi(\cdot)$ to merge the extracted features in a common multimodal space. The function $\Psi(\cdot)$ can be a simple fixed function (e.g., a linear sum, concatenation followed by MLP) or a complex operation (e.g., multimodal pooling [Fukui et al., 2016] or fusion [Ben-Younes et al., 2017, 2019]). Most importantly, the multimodal embedding is used to selectively attend to visual features using a learned attention mechanism. This attention is derived jointly from the given question and image pair. Different from these attention approaches, we propose a pre-prcoessing step that estimates an atten-

tion map *without* considering at-all the input questions. This simple approach with no-training cost surprisingly gives highly compelling results.

Our proposed question-agnostic attention model is illustrated in Fig. 4.2. We first employ an attention mechanism that focuses on different object instances by creating an '*object map*' with which the question-agnostic features are generated (Sec. 4.2.1). The question-agnostic attention enables the model to focus on arbitrary object shapes and object parts which results in an improved model attention. Instead of the original CNN extracted spatial grid feature map, the question-agnostic features are passed through the VQA model where the given language query is used to further refine the visual features. These refined visual features are used to generate final predictions for classification. The modular architecture of our model enables it to combine predictions from other VQA models that aggregates multiple predictions to generate an intelligent answer for the given question (Sec. 4.2.2).

4.2.1 Question-Agnostic Attention

The input image is passed through a pre-trained instance segmentation module to predict the pixels that correspond to object instances. Notably, we ensure that the pre-trained network has not seen any of the test images for the evaluated datasets and is pre-trained on an altogether different task (i.e., instance segmentation as opposed to VQA). These instances have arbitrary shape and size which makes it harder to encode them and computationally infeasible for a VQA model to train with instance-level features. To remedy this, a coarse representation of the object instances is generated by creating a grid of size g over the whole image and the object instances are mapped onto this grid. A binary representation of this grid is called the *object map* $\mathcal{M} \in \mathbb{R}^g$, which essentially identifies if an object instance occupies a grid location or not.

One can learn a non-linear mapping function that maps the object instances to an arbitrary high-dimensional space. However, we adopt a simplistic approach to set the size of the *object map* equal to spatial grid size g for primarily three reasons. *Firstly*, having the grid size equal to the CNN features allows our approach to establish a one-to-one correspondence to the spatial grids of the CNN extracted convolutional feature map, which enables the model to access the visual features of that gird region without requiring another explicit ROI pooling like Faster-RCNN. This avoids expensive computations in our model. *Secondly*, the binary g-dimensional *object map* can be applied as a mask to select only the visual features at grid locations that have an object instance with a computationally inexpensive element-wise multiplication between $\boldsymbol{v}$ and $\mathcal{M}$ to generate question-agnostic feature $\boldsymbol{v}^{\mathcal{M}} \in \mathbb{R}^{g \times d_v}$. *Finally*, as the question-agnostic features have the same size as CNN extracted visual features, it can be easy for any VQA model to incorporate the question-agnostic features by only adjusting the size of *object map* equal to the size of CNN spatial grid. Thus, this simplistic approach fashions question-agnostic attention mechanism as an inexpensive pre-processing step that is easily applicable to any CNN based VQA model.

4.2.2 Multiple Prediction Embedding

The modular architecture of QAA enables it to jointly consider predictions from any other VQA model to generate a final prediction vector. In order to further validate the complementary nature of our proposed QAA model, we include a simple spatial attention mechanism, commonly used in most VQA models [Fukui et al., 2016; Ben-Younes et al., 2017, 2019] to refine the visual features according to the question. In addition to a fixed *object map* used to generate question-agnostic features, this optional module can be used to refine the question-agnostic features according to the question, providing flexibility to incorporate a spatial attention mechanism on top of QAA. We achieve this by calculating a similarity measure between each question-agnostic feature grid location $\boldsymbol{v}_i^{\mathcal{M}} \in \mathbb{R}^{d_v}$ and $\boldsymbol{q}$ by projecting them in a common space by a joint embedding function $\Psi(\cdot)$. This, in general, represents the relevance of a spatial grid location for answering that input question. This similarity measure is applied as a semantic weighting function, called Spatial Attention $\alpha \in \mathbb{R}^g$, that takes a weighted sum over the spatial grids of input visual features. It can be expressed as:

$$\tilde{\boldsymbol{v}}^{\mathcal{M}} = \sum_{i=1}^{g} \alpha_i \boldsymbol{v}_i^{\mathcal{M}} \quad \text{where} \;\; \alpha_i = \text{softmax}(\Psi(\boldsymbol{q}, \boldsymbol{v}_i)). \tag{4.2}$$

Here $\tilde{\boldsymbol{v}}^{\mathcal{M}} \in \mathbb{R}^{d_v}$ represents a combination of question-agnostic attention features that are emphasized by the input question. Finally, it undergoes a second multimodal embedding with the question feature $\boldsymbol{q}$ to generate a prediction vector P which has the same dimension as the candidate answer dictionary $\mathcal{D}$. Predictions from any other VQA model can be concatenated with the prediction of our model. The concatenated predictions are passed through a multiple prediction embedding layer that learns to generate a $\mathcal{D}$ dimensional final prediction vector.

4.3 Experiments and Results

In this section, at first, we describe the experimental setup that includes our instance segmentation pipeline, VQA model architecture, dataset and evaluation metric. Then we discuss the findings from our ablative experiments to study effectiveness of our proposed approach in different settings. Finally, we present the qualitative and quantitative results of our model and do a comparative analysis with other state-of-the-art models.

4.3.1 Experimental Setup

In this section, we provide additional details of the datasets that we experimented on and our model architecture setup. In Fig. 4.1 we show the VQA accuracy gain achieved while different VQA models use complementary QAA features.

VQA dataset: Firstly, we evaluate our QAA model on two large scale benchmark VQA datasets, namely VQAv1 [Antol et al., 2015] and VQAv2 [Goyal et al., 2017]. Among the two datasets, VQAv2 contains complementary question-answer

			VQAv1 Dataset				VQAv2 Dataset			
		Spatial	Multimodal Operation				Multimodal Operation			
	Visual Feature	Attention	Linear	C-MLP	Mutan	Block	Linear	C-MLP	Mutan	Block
(1)	Spatial Grid (SG)	✗	39.7	57.2	56.3	58.2	38.2	51.9	55.6	56.8
(2)	QAA	✗	41.4	40.5	57.3	58.4	39.7	53.2	56.3	56.5
(3)	Ours(SG+QAA)	✗✗	**57.9**	**58.3**	**57.5**	**58.4**	**56.2**	**56.8**	**57.0**	**57.1**
			18.2 ↑	1.1 ↑	1.2 ↑	0.2 ↑	18.2 ↑	4.9 ↑	1.4 ↑	0.3 ↑
(4)	Spatial Grid (SG)	✓	41.8	60.4	58.6	61.2	41.0	54.4	57.9	60.1
(5)	QAA	✓	41.4	59.6	57.9	60.5	37.3	57.3	56.5	59.3
(6)	Ours(SG+QAA)	✓✓	**60.6**	**60.7**	**59.2**	**61.6**	**59.1**	**59.5**	**58.2**	**60.5**
			18.8 ↑	0.3 ↑	0.6 ↑	0.3 ↑	18.1 ↑	5.2 ↑	0.3 ↑	0.4 ↑
	Multimodal Parameters		8M	13M	14M	18M	8M	13M	14M	18M

Table 4.1: *Comparison of different multimodal operations when using complementary QAA features on VQA datasets.* The models are evaluated on validation sets of VQAv1[Antol et al., 2015] and VQAv2[Goyal et al., 2017] dataset, and we report the overall accuracy (higher the better). Models in rows (1)-(3) do not have any spatial attention mechanism whereas the models in rows (4)-(6) learn spatial attention as described in Sec.4.2.2

pairs which mitigates language bias present in the VQAv1 dataset, making VQAv2 a more challenging test setting. Both versions of the VQA dataset contain over 200K real images sourced from the MS COCO dataset [Lin et al., 2014] and placed into respective train/val/test splits. These images are paired with complex open-ended natural language questions and answers. The ground truth answers for train and val split are publicly available, but the test split is not. To evaluate on the test split (both test-dev and test-std), the prediction needs to be submitted to the VQA test server. We perform ablation experiments on validation sets of VQAv1 and VQAv2 (Tab. 4.1) and compare with other state-of-the-art methods on VQAv2 test-dev and test-std dataset (Tab. 4.2). Following the standard evaluation strategy, we calculate the accuracy $\hat{a}$ of the predicted answer a^* as $\hat{a} = \min(\#\text{ of humans answered } a^*/3, 1)$, which means that the answer provided by the model is given 100% accuracy if at least 3 human annotators who helped create the VQA dataset gave the exact answer.

TDIUC dataset: Task Directed Image Understanding Challenge (TDIUC) dataset [Kafle and Kanan, 2017] consists of $1.6M$ questions and $170K$ images sourced from MS COCO and the Visual Genome Dataset. These Image-Question pairs are split into 12 categories and 4 additional evaluation matrices (1^{st} column of Tab. 4.3) which help evaluate a model's robustness against answer imbalance and its ability to answer questions that require higher reasoning capability. We evaluate and perform ablation on TDUIC testset, and report accuracy for all 12 question types along with overall arithmetic mean-per-type (MPT) and harmonic MPT, and overall normalized arithmetic MPT and harmonic MPT in Tab. 4.3.

Model Architecture: We use ResNet [He et al., 2016] pretrained on ImageNet [Deng et al., 2009] to extract the visual features of an image with dimensions 196×2048. Here, $g = 196$ which represents the 14×14 spatial grid corresponding to image regions and 2048 is the dimension of visual features for each grid location. The language model generates a $d_q = 2400$ dimensional feature for each question in a fashion similar to [Fukui et al., 2016; Ben-Younes et al., 2017; Yu et al., 2018]. The question words are first preprocessed, tokenized and encoded through a embedding layer that consists of GRUs and uses a pretrained skip-thought encoder. For the models *without* the optional spatial attention mechanism, the input visual feature is averaged across the spatial grid to generate a 2048-d feature vector from the $2048 \times 14 \times 14$ dimensional feature map and passed on to be jointly embedded with the question feature. On the other hand, the models *with* spatial attention learn to generate 2048-d feature vector as discussed in Sec. 4.2.2. Following the VQA benchmark [Antol et al., 2015], the dictionary of candidate answers $\mathcal{D}$ consists of the top 3000 frequent answers from the respective versions of VQA dataset. A cross entropy loss is minimized to predict the correct answer from the dictionary $\mathcal{D}$. While performing experiments on the TUDIC dataset, dimension of $\mathcal{D}$ is set to 1480.

Instance Segmentation: We employ a pre-trained Mask-RCNN [He et al., 2017] model [1] to generate instance masks by running inference on the input image. Specifically, the Mask-RCNN model was trained on COCO *train* and the *val-minus-minival*

[1] github.com/facebookresearch/maskrcnn-benchmark

split with a ResNet-50-FPN backbone. Although the 'training data' (i.e., images) of the VQA datasets have an overlap with the 'training set' of COCO, none of the test images have been previously seen by the pre-trained model. Also, we do not use any object-level information in our attention map, rather only a simple binary mask showing the location of detected objects is used. Therefore, our setting has no extra advantage or external supervision compared to other approaches.

Baseline Model: We setup our VQA baseline model with four variants where the model employs different multimodal operations for combining the question and image features. All other setup and hyperparameters are kept exactly the same for fair comparison. Each variant can have the optional spatial attention module. More details about the experimental setup and datasets can be found in the supplementary materials. The first two variant of our baseline model incorporate simpler multimodal operation (i.e., liner summation and concatenation followed by MLP). The latter two variants use a more complex multimodal operation, namely Mutan [Ben-Younes et al., 2017] and Block [Ben-Younes et al., 2019], which achieve the state-of-the-art performance for the VQA task, and have a considerably higher number of trainable parameters for multimodal embedding. Mutan and Block operations are implemented using their publicly available code[2]. The following are the four variants of our baseline model:

Linear: The question and image features are projected onto a common space using fully connected layers and the projected vectors are summed to obtain a joint feature representation. This joint representation is projected to the prediction space $P \in \mathbb{R}^{3000}$ which is then passed through the answer prediction network to generate the final prediction. This can be expressed as:

$$P' = \omega_P(\omega_q \boldsymbol{q} + \omega_v \boldsymbol{v}), \tag{4.3}$$

where ω represents the fully connected layer weights used for projection.

Concat-MLP: The question and image features are concatenated and passed through a 3-layer MultiLayer Perceptron (MLP) with ReLU activation and dropout to combine the input features. The resulting vector is projected onto the prediction space for answer classification.

Mutan: The Mutan model learns a multimodal interaction between question and image using rank constrained Tucker tensor decomposition [Ben-Younes et al., 2017]. In this model, the visual and language features are decomposed into three matrices and a core tensor that is somewhat capable of modelling the fully parameterized interaction in the multimodal space.

Block: It employs block-term tensor decomposition following a super-diagonal fusion framework[Ben-Younes et al., 2019]. This is the most computationally expensive model that we experiment with and achieves state-of-the-art performance. The complexity of a multimodal operation is inferred by calculating the number of trainable parameters from attended image features, the question embedding, and the answer prediction.

[2] github.com/Cadene/block.bootstrap.pytorch

Ablation study: In Sec. 4.3.2 we perform ablation to showcase the effectiveness of using complementary QAA on VQA models employing different multimodal operation by evaluating on the VQAv2 Valset [Goyal et al., 2017] and the TDIUC testset [Kafle and Kanan, 2017]. Furthermore, in Sec. 4.3.3 we show that without an explicit *object map* during inference, our model can utilize image independent QAA features generated from a global representation of training examples.

4.3.2 Ablation on Different Multimodal Operations

Simplistic VQA models get a significant performance boost using complementary QAA features and perform on par with the state-of-the-art. In row (1) of Tab. 4.1, we report that our baseline VQA model employing state-of-the-art Block fusion achieves 58.4 and 57.1 accuracy, whereas with a linear-sum operation, the same model achieves accuracy of only 39.7 and 38.2 on VQAv1 and VQAv2 validation sets, respectively. When the Linear model is trained with complementary QAA features, the accuracy increases to 57.9 and 56.2 on the VQAv1 and VQAv2 datasets, respectively; performing very close to the state-of-the-art Block model (row (3)). This pattern also exists when these same models include the optional spatial attention module (comparing rows (4) and (6)). The simpler Linear model benefits from complementary QAA features as it represents a subset of the spatial locations of the whole image that has object instances and encodes visual cues like count, location and attributes which are most important to predict the correct answer. The Linear model with only 8M trainable parameters and relatively simpler multimodal operation cannot learn to identify these visual cues on its own. Thus the performance boost while using complementary QAA feature is more significant ($\sim$18$\uparrow$ vs $\sim$0.5$\uparrow$) for VQA models employing a simplistic multimodal operation (i.e., Linear and Concat-MLP) compared to the models employing a more sophisticated fusion operation (i.e., Mutan[Ben-Younes et al., 2017] and Block[Ben-Younes et al., 2019]). Since more complex multimodal operations learn salient visual cues by modeling the interaction between visual and semantic features through significantly more parameters; VQA models employing such complex operations benefit less from using complementary QAA features. Overall, Tab. 4.1 and Fig. 4.1 shows that all variants of our VQA baseline employing different multimodal operations, with or without optional spatial attention, benefit from using complementary QAA features.

Complementary QAA features help answer rare question more accurately. In Tab. 4.3, we valuate our baseline models with and without complementary QAA features on the TDIUC testset and compare it against other state-of-the-art models using spatial grid features. The baseline models reported in this table use the optional spatial attention module. We can see that the accuracy for the difficult question categories (e.g., Object Utility, Object Presence) increased when using QAA features, and this improvement is more prominent for models using Linear and Concat-MLP operations. Further, for all variants of the baseline model, both versions of Arithmetic and Harmonic MPT improved, and this improvement is more significant for Harmonic MPT and Harmonic N-MPT. This is particularly important as Harmonic MPTs is a more strict

	Model	Test-dev All	Test-Standard All	Y/N	Num.	Other
	MCB	-	62.3	78.8	38.3	53.3
(1)	Mutan	63.2	63.5	80.9	38.6	54.0
	Ours(SG+QAA)	**64.7**	**65.0**	**81.8**	**43.6**	**55.4**
	Up-Down	65.3	65.7	82.2	43.9	56.3
(2)	Block	66.4	66.9	83.8	45.7	57.1
	Ours(BU+QAA)	**66.7**	**67.0**	**83.8**	**45.9**	**57.1**

Table 4.2: *Comparison with state-of-the-art VQA models on VQAv2 Test-dev and Test-std dataset.* In (1), we compare MCB [Fukui et al., 2016] and Mutan [Ben-Younes et al., 2017] models, which are trained with spatial grid (SG) features with our SG+QAA model; and in (2) we compare Up-Down [Anderson et al., 2018] and Block [Ben-Younes et al., 2019] model, trained with Bottom-Up features with our BU+QAA model. With QAA, in both cases, our model outperform contemporary VQA models.

metric as it measures the ability of a model to have high scores across *'all'* question-types and it consequently puts an emphasis on lowest performing categories. In the last row of Tab. 4.3, we report the traditional VQA accuracy and observe that the Block variant of our(SG+QAA) model achieves higher accuracy than other state-of-the-art methods. Furthermore, the Concat-MLP model achieves almost same traditional VQA accuracy with or without QAA features ($\sim$ 84.0). Interestingly, one can notice that, even with same VQA accuracy, our model achieves a significant boost in both versions of Arithmetic and Harmonic MPT. These findings support our hypothesis that the QAA features encode salient object-level information that helps consider high-level visual concepts when answering difficult questions.

4.3.3 Inference with Global Representation

We further experiment with Image-Question-Agnostic Attention (IQAA) where the attention feature is generated without looking at the input question *and* image. To do so, **first**, we create a global representation of *object maps* by counting object presence at each spatial grid location for all images in the dataset. In Fig. 4.4, we show such a global representation from the count of object presence, $\mathcal{C} \in \mathbb{R}^{14\times14}$, of VQA dataset training images (i.e., COCO trainset 2014 images) on a 14×14 grid. We can see from this figure that most objects present in an image occupy the center grids. We leverage this centre bias to create fixed *object maps*, that in turn is used to generate IQAA features. **Second**, the count vector is min-max normalized between $[0, 1]$ (x-axis of Fig. 4.3). The left y-axis shows the number of grid locations selected when applying different thresholds on the normalized count measures. It ranges from 191 to 22 grid locations when the threshold is varied between 0.1 to 0.9. **Third**, we treat the selected grid location for a set threshold as fixed *object maps* and apply fixed map on the input visual feature as discussed in Sec. 4.2.1 for generating IQAA features. These IQAA features can be used instead of QAA features in a similar fashion to

	MCB	NMN	RAU	Linear	Ours (SG+QAA)	Concat -MLP	Ours (SG+QAA)	Mutan	Ours (SG+QAA)	Block	Ours (SG+QAA)
Scene Recog.	93.0	91.9	94.0	50.9	93.1	92.5	93.0	92.2	92.4	92.8	92.8
Sport Recog.	92.8	90.0	93.5	19.0	93.7	93.4	94.1	93.0	92.9	93.5	93.5
Color Attributes	68.5	54.9	66.9	55.7	67.1	65.4	68.2	66.3	66.2	68.6	64.5
Other Attributes	56.7	47.7	56.5	0.1	54.9	56.3	56.4	52.1	52.4	57.9	56.1
Activity Recog.	52.4	44.3	51.6	0.0	50.9	52.3	53.0	49.3	50.2	53.2	52.4
Pos. Reasoning	35.4	27.9	35.3	7.3	33.4	32.2	35.4	29.4	29.9	36.1	34.7
Sub-Obj Recog.	85.4	82.0	86.1	23.8	85.7	86.1	86.5	85.2	85.5	86.2	85.9
Absurd	84.8	87.5	96.0	90.3	88.2	92.4	92.4	90.0	89.1	90.7	92.1
Object Utility	35.0	25.1	31.6	15.2	29.3	26.2	35.7	27.4	30.4	34.5	37.4
Object Presence	93.6	92.5	94.4	93.5	94.3	94.3	94.4	93.8	93.9	94.1	94.2
Counting	51.0	49.2	48.4	50.1	51.2	53.0	52.6	51.2	50.4	51.1	51.2
Sentiment Undstd.	66.3	58.0	60.1	56.3	65.8	65.7	66.3	63.2	61.0	66.0	63.5
Arithmetic MPT	67.9	62.6	67.8	38.5	68.3(29.8↑)	67.6	**69.0**(1.4↑)	66.2	66.3(0.1↑)	68.4	68.8(0.4↑)
Harmonic MPT	60.5	51.9	59.0	0.0	58.1(58.1↑)	57.3	**61.3**(4.0↑)	55.1	56.7(1.6↑)	60.0	61.1(1.1↑)
Arithmetic N-MPT	42.5	34.0	41.0	29.8	54.1(34.3↑)	53.4	**56.4**(3.0↑)	53.1	53.7(0.6↑)	54.7	55.9(1.2↑)
Harmonic N-MPT	27.3	16.7	24.0	0.0	32.3(32.3↑)	28.2	**38.8**(3.0↑)	32.3	32.8(0.6↑)	34.1	38.2(1.2↑)
Simple Accuracy	81.9	79.6	84.3	72.9	82.6(9.7↑)	83.9	84.0(0.1↑)	82.5	82.7(0.2↑)	84.5	**84.6**(0.1↑)

Table 4.3: *Evaluation of our QAA models on the testset of TDIUC [Kafle and Kanan, 2017] dataset and comparison with state-of-the-art MCB[Fukui et al., 2016], NMN[Andreas et al., 2016] and RAU[Noh and Han, 2016] methods.* The first 12 rows report the unnormalized accuracy for each question-type. The Arithmetic MPT and Harmonic MPT are unnormalized averages, and Arithmetic N-MPT and Harmonic N-MPT are normalized averages of accuracy scores for all question type. The last row shows the simple VQA accuracy for all models. Using complementary QAA features, the models ability to answer rare questions increased significantly (i.e., higher Harmonic MPT and N-MPT) for all cases.

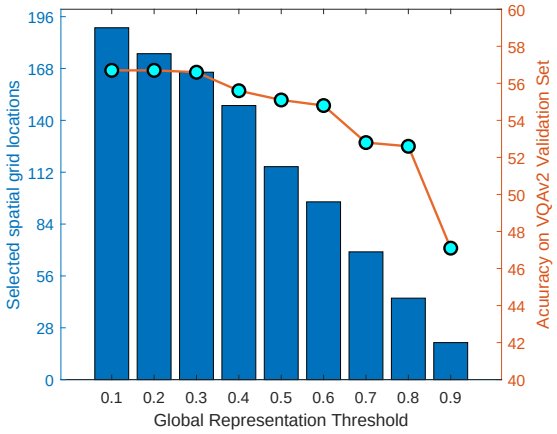


Figure 4.3: *VQA accuracy (right y-axis) using Image-Question-Agnostic Attention (IQAA) features.* IQAA feature is generated by selecting a global representation threshold (x-axis) and corresponding spatial grid locations (left y-axis).

train any VQA model.

By only using complementary IQAA features VQA models achieve reasonable performance. We report the VQA accuracy score on VQAv2 validation set using our Block baseline model without spatial attention on the right y-axis of Fig. 4.3. When using IQAA features, the VQA accuracy ranges from 53 to 56 when the global representation threshold is varied between 0.1 to 0.9. This means if one selects a fixed set of 24 spatial grid locations at the center of the image, and trains a state-of-the-art VQA model with visual features of only these grid locations; the model can still achieve VQA accuracy comparable to when it looks at the whole image. A similar finding was reported by Judd et al. [2009] where they show that humans tend to focus the object at the center when they take picture. Our finding further adds to that notion of *Center Prior* by showing that humans also tend to ask questions about objects that are at the center of the image. Even though we run inference with pre-trained Mask-RCNN to generate QAA features as a light-weight prepossessing step, by modeling the object presence prior in the dataset, one can further reduce pre-training computational burden by replacing QAA with IQAA, and achieve reasonable performance.

4.3.4 Evaluation on the VQAv2 Testset

We evaluate our model's performance on the VQAv2 Test server and report accuracy for different question types on the Test-dev and Test-std dataset to compare with other contemporary state-of-the-art VQA models. For fair comparison, in Tab. 4.2 (1), we separate models that use spatial grid features (i.e., visual features extracted by ResNet) and compare it with our SG+QAA model; in (2) the models that use Bottom-Up [Anderson et al., 2018] features and we compare our BU+QAA model. For both cases, our question-agnostic models employ Block fusion to jointly embed image and question features with a spatial attention mechanism. From Tab. 4.2, we

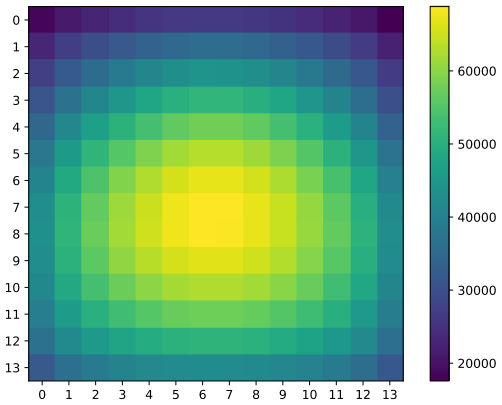


Figure 4.4: *Global Representation of object maps*. Count of object presence at each of the **196 (14×14)** spatial grid locations generated from the training images of VQA dataset.

can see that when the QAA features are used alongside spatial grid features, the gain is more, compared to when used with BU features. As the BU features are a collection of top object bounding box features generated using Faster-RCNN, it also offers some object-level information to the VQA model. Thus, when used in combination with BU features, the overall performance gain in relatively small. However, as the question-agnostic features encompass the *Object Map* of an image, it somewhat encodes the global spatial relationship between object and count information of object instances; it provides accuracy gain when answering *Number* (i.e., *'How many?'*) question (0.2% ↑ in test-standard). Overall, if a parallel branch trained using question-agnostic features is added to an existing VQA model, accuracy of the model increases.

4.3.5 Qualitative Results

We present qualitative results of our SG+QAA model with Block fusion in Fig. 4.5 to showcase the effectiveness of complementary question-agnostic features. In the second row, first example, the model is tasked with a count question: *'How many animals are on the grass?'* The learned spatial attention map is scattered in different image locations whereas the question-agnostic feature localizes five object instances that help the model answer correctly. Overall, from the qualitative results, we can deduce that learned and question-agnostic attention provides complementary information which can be leveraged by VQA models to correctly answer intelligent questions.

4.4 Conclusion

In this chapter, we introduced Question-Agnostic Attention that can be used to augment existing VQA approaches. Rather than using computationally intensive methods to learn question-specific attention, our approach derives attention only from the

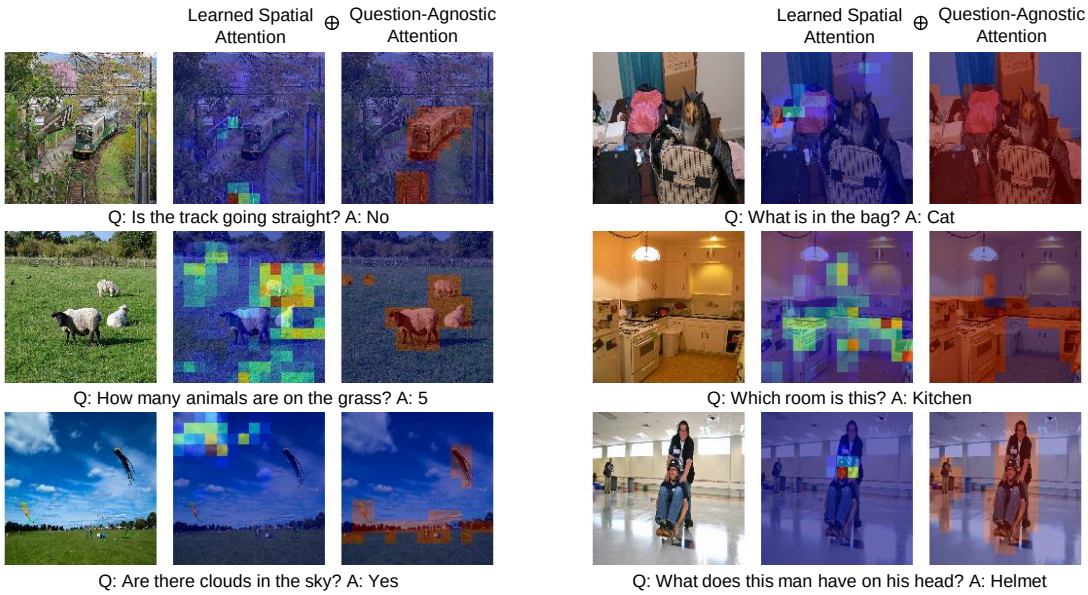


Figure 4.5: *Qualitative results on VQAv2 val-set to demonstrate the effectiveness of using complementary QAA.* The learned spatial attention (2^{nd} and 5^{th} columns) focuses on regions with or without objects, but QAA map is focused on objects.

image, based on the insight that questions generally relate to object instances. We use an object parsing model to automatically generate an *Object Map*, that has the same resolution as the feature map from a pre-existing classification network. The *Object Map* is used to mask the convolutional feature map to generate question-agnostic attention features. When high-performing computationally-intensive VQA models are augmented with QAA, it improves their accuracy to be a new state-of-the-art. When simple linear models are augmented with QAA, they preform significantly better when answering question that require a higher level of visual reasoning (e.g., activity recognition), which a simplistic model cannot learn on its own. This capability provides the simplistic (low-complexity) models a significant boost that brings them close to state-of-the-art.

Chapter 5

Exemplar Based Knowledge Transfer

Current Visual Question Answering (VQA) systems can answer intelligent questions about *known* visual content. However, their performance drops significantly when questions about visually and linguistically *unknown* concepts are presented during inference ('Open-world' scenario) [Bansal et al., 2018; Rahman et al., 2018b; Geng et al., 2020; Rahman et al., 2018a]. A practical VQA system should be able to deal with novel concepts in open-world settings. In this chapter, we propose an exemplar-based approach that transfers learning (i.e., knowledge) from previously *known* concepts to answer questions about the *unknown*. We learn a highly discriminative joint embedding (JE) space, where visual and semantic features are fused to give an unified representation. Once novel concepts are presented to the model, it looks for the closest match from an exemplar set in the JE space. This auxiliary information is used alongside the given Image-Question pair to refine visual attention in a hierarchical fashion. Our novel attention model is based on a dual-attention mechanism that combines the complementary effect of spatial and channel attention. Since handling the high dimensional exemplars on large datasets can be a significant challenge, we further introduce an efficient matching scheme that uses a compact feature description for search and retrieval. To evaluate our model, we propose a new dataset for VQA, separating *unknown* visual and semantic concepts from the training set into the test set. To this end, we do not source new images or questions, rather re-purpose the already available image-question pairs from VQA-v1 and VQA-v2 datasets into visually and semantically non-overlapping train and test sets. Our approach shows significant improvements over state-of-the-art VQA models on the proposed Open-World VQA dataset and other standard VQA datasets. This chapter is based on our published work [Farazi et al., 2020c], previously mentioned in Sec. 1.6.

5.1 Introduction

Machine vision algorithms have significantly evolved various industries such as internet commerce, personal digital assistants and web-search. A major component of machine intelligence comprises of how well it can comprehend visual content.

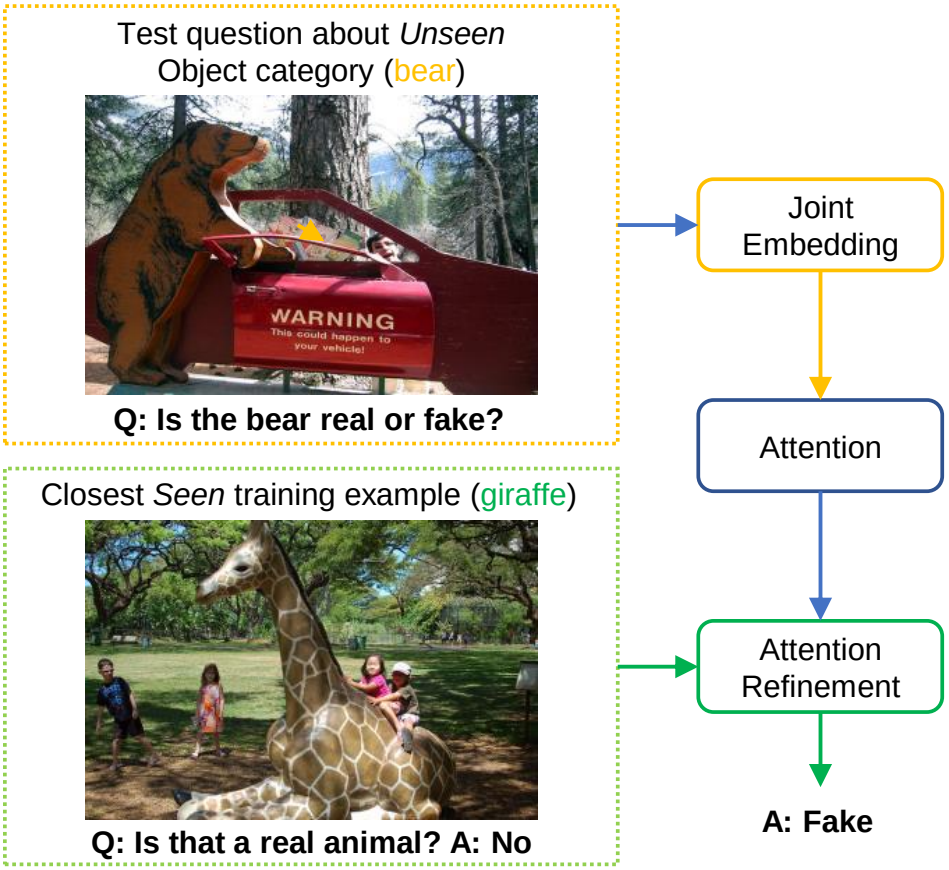


Figure 5.1: *Open World VQA for Novel Concepts.* Our model learns to represent multi-modal information (Image ($\boldsymbol{I}$)-Question ($\boldsymbol{Q}$) pairs) as a joint embedding (Φ). Once presented with *unknown* concepts, the proposed model learns to effectively make use of the Joint Embedding (JE) space by using past knowledge accumulated over the training set to answer intelligent questions.

A Visual Turing Test to assess a machine's ability to understand visual content is performed with the Visual Question Answering (VQA) task. Here, machine vision algorithms are expected to answer intelligent questions about visual scenes. In the real world, humans can easily reason about visual and semantic unknown concepts based on previous knowledge about the known. For instance, having seen visual examples of '*lion*' and '*tiger*', a person can recognize an unknown '*liger*' by associating visual similarities with a new compositional semantic concept and answer intelligent questions about their count, visual attributes, state and actions. However, one key limitation of current VQA paradigm is that the questions asked at inference time only relate to the concepts that have already been known during the training stage (*closed-world* assumption). To address this limitation, we introduce a novel VQA problem setting that evaluates models in an '*Open-World*' dynamic scenario where previously unknown concepts (both visual and semantic) show up during inference (Fig. 5.1).

An open-world VQA setting requires a vision system to acquire knowledge over

time and later use it intelligently to answer complex questions about *unknown* concepts for which no linguistic+visual examples were available during training. In order to design machines to mimic human visual comprehension abilities, we must impart lifelong learning mechanisms that allow them to accumulate and use past knowledge to relate unknown concepts with known concepts. Existing VQA systems lack this capability as they use a 'fixed model' to acquire learning and envisage answers without explicitly considering closely related examples from the knowledge base. This can lead to 'catastrophic forgetting' [McCloskey and Cohen, 1989] as the object/question set is altered with updated categories. We address this out-of-domain knowledge transfer problem by developing a flexible knowledge base from the training examples without using external information. The knowledge base contains joint representation of visual and semantic embedding features of each training Image-Question pair in a highly discriminative latent space, dubbed Joint Embedding (JE) space. Building the knowledge base in the joint embedding space allows our model to search for training examples that are both visually and semantically similar to the *unknown* concepts. As seen in Fig. 5.1, the contextual cue that happy children usually play around a fake wild giraffe (i.e., a known animal class) can be leveraged and used to infer that humans are most likely to be seen near another fake wild bear (i.e., an unknown animal class).

We propose a new deep Convolutional Neural Network (CNN) architecture to perform knowledge transfer between the *known* and *unknown* concepts. Our CNN model has three components. First, a multi-modal feature embedding is automatically learned that jointly models the visual and semantic domains. Secondly, the exemplars are represented in the joint embedding space and consequently matched with the newly presented image-question pairs using an efficient retrieval scheme. This step ensures knowledge transfer from the closely related exemplars to help answer challenging questions regarding the *unknown* concepts. Finally, a dual self-attention mechanism is applied on top of input image-question and exemplar embeddings to refine attention on the visual features.The self-attended features are then used for the answer prediction by our proposed VQA model.

Related to our work, we note a few recent efforts that aim to extend VQA beyond the already known concepts [Wang et al., 2017b; Agrawal et al., 2018; Teney and Hengel, 2016; Ramakrishnan et al., 2017; Agrawal et al., 2017]. A major limitation of these approaches is they introduce novel concepts only on the language side (i.e., new questions/answers), either to re-balance the split or to prevent the model cheating by removing biases [Agrawal et al., 2018; Teney and Hengel, 2016; Agrawal et al., 2017]. Further, they rely on external data sources (both visual and semantic) and consider training splits that contain visual instances of 'novel concepts' [Ramakrishnan et al., 2017; Teney and Hengel, 2016], thereby violating the *unknown* assumption. To bridge this gap, we propose a new Open-World VQA dataset named OW-VQA.

We make the following major contributions in this chapter:

- We reformulate VQA in a transfer learning setup that uses closely related *known* instances from the exemplar set to reason about *unknown* concepts and propose

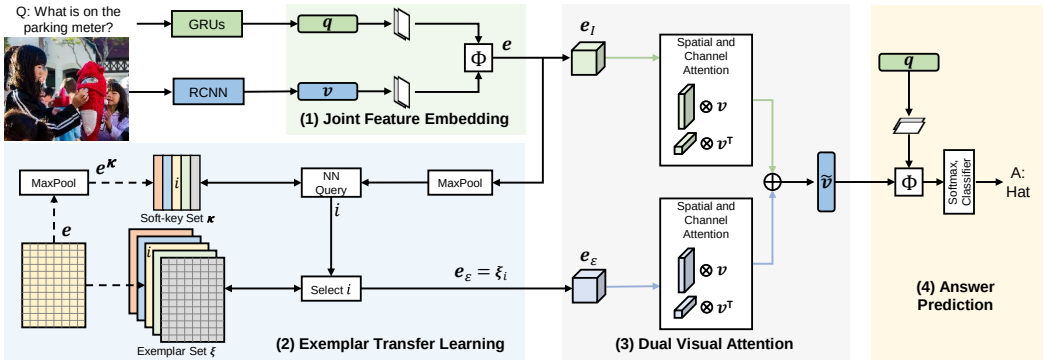


Figure 5.2: *Overview of our proposed Joint Embedding Exemplar (JE+X) model.* Given an Image-Question (IQ) pair, our model populates an exemplar set with visual-semantic joint embedding (represented by dotted lines) during training. When novel concepts (both visual and semantic) are encountered during inference, our model identifies a closely related exemplar to provide complementary information that helps in generating an intelligent answer with an improved dual attention mechanism (spatial and channel).

a new OW-VQA dataset, re-purposing existing VQA datasets, to enable impartial evaluation of VQA algorithms in a real-world setting.

- We present a novel exemplar-based search and retrieval scheme to enable efficient exemplar matching on a rich feature embedding space that aggregates visual and semantic information.
- We propose a network architecture that incorporates dual visual attention (spatial and channel) in the joint embedding space to intelligently attend to relevant features for accurate VQA.

5.2 Joint Embedding Exemplar Model

An ideal VQA system must effectively model the complex interactions between the language and visual domains to acquire useful knowledge and use it to answer newly presented queries at test time. A VQA engine can encounter questions about previously known as well as unknown concepts. Towards this end, we propose a framework to answer questions about novel concepts in Fig. 5.2. The overall pipeline is based on four main steps: *(1) Joint Feature Embedding:* An Image-Question (IQ) pair is processed to extract visual v and language q features. These features are jointly embedded into a common space through multi-modal fusion. *(2) Exemplar Transfer Learning:* We propose an exemplar-based model that learns to reason from similar examples known during training time. When presented with a test image containing an unknown concept, our model transfers the knowledge acquired on closely related examples to novel cases. *(3) Dual Visual Attention:* Our model applies spatial

and channel attention on joint embeddings obtained from the input IQ pair (output of Step 1) and retrieved exemplar (output of Step 2) to generate complementary visual cues. These visual cues are used to apply attention on the input visual features which ensures that our model identifies salient visual features for both known and unknown concepts, relevant to the input question. *(4) Answer Prediction:* The attended visual features undergo a final fusion with the question features that generate a refined joint embedding, which is then passed through a classifier to predict the correct answer a^* from an answer set $\boldsymbol{A}$.

5.2.1 Joint Feature Embedding

For a given image $\boldsymbol{I}$, the n_v-dimensional visual feature embedding $\boldsymbol{v} \in \mathbb{R}^{n_v}$ is the set of top object proposals obtained by applying bottom-up attention following Anderson et al. [2018]. The language feature embedding $\boldsymbol{q} \in \mathbb{R}^{n_q}$ is generated from $\boldsymbol{Q}$ by first encoding the question as a one-hot-vector representation and then embedding into vector space using Gated Recurrent Units (GRUs) [Cho et al., 2014; Fukui et al., 2016]. In order to predict a correct answer, a VQA model needs to generate a joint embedding: $\boldsymbol{e} = \Phi(\boldsymbol{v}, \boldsymbol{q}; \boldsymbol{\tau}) \in \mathbb{R}^{n_e}$ given parameters $\boldsymbol{\tau}$. A naive approach that models visual-semantic interactions using a tensor $\boldsymbol{\tau} \in \mathbb{R}^{n_q \times n_v \times n_e}$ will result in an unrealistic number of trainable parameters (e.g., ~ 9 billion for our baseline model).

To reduce the dimensionality of the tensor, we use Tucker decomposition [Tucker, 1966] that works as a high-order principal component analysis operation. This technique has been proven effective in embedding visual and textual features for VQA [Ben-Younes et al., 2017; Farazi and Khan, 2018]. It approximates $\boldsymbol{\tau}$ as by:

$$\begin{aligned} \boldsymbol{\tau} &= \sum_{i=1}^{t_q} \sum_{j=1}^{t_v} \sum_{k=1}^{t_e} \omega^{ijk} \tau_q^i \circ \tau_v^j \circ \tau_e^k \\ &= \boldsymbol{\omega} \times_1 \boldsymbol{\tau}_q \times_2 \boldsymbol{\tau}_v \times_3 \boldsymbol{\tau}_e = [\![\boldsymbol{\omega}; \boldsymbol{\tau}_q, \boldsymbol{\tau}_v, \boldsymbol{\tau}_e]\!] \end{aligned} \tag{5.1}$$

where $\times_i$ denotes n-mode tensor-matrix product, $[\![\cdot]\!]$ are Iverson brackets and $\circ$ denotes the outer vector product. Eq. 5.1 shows that tensor $\boldsymbol{\tau}$ is decomposed into a core tensor $\boldsymbol{\omega} \in \mathbb{R}^{t_q \times t_v \times t_e}$ and orthonormal factor matrices $\boldsymbol{\tau}_q \in \mathbb{R}^{n_q \times t_q}$, $\boldsymbol{\tau}_v \in \mathbb{R}^{n_v \times t_v}$, $\boldsymbol{\tau}_e \in \mathbb{R}^{n_e \times t_e}$. Intuitively, by setting $t_q < n_q$, $t_v < n_v$ and $t_e < n_e$, one can approximate $\boldsymbol{\tau}$ with only a fraction of the originally required parameters.

The output embedding $\boldsymbol{e}$ from the Tucker decomposition effectively captures the interactions between semantic and visual features for a given Image-Question-Answer (IQA) triplet. Such joint embedding for VQA is specific to the given IQ pair because the same visual feature associated with different semantics (and vice-versa) will result in a different joint embedding specific for that pair. For example, given an image that captures children playing in the backyard, when asked *'How many children are in the picture?'* and *'Are the children happy?'*, requires two very different joint embeddings $\boldsymbol{e}$ even though they use the same visual feature, $\boldsymbol{v}$. Building on this rich joint embedding, we develop a transfer learning module using exemplars.

5.2.2 Exemplar Transfer Learning

Given a question about an *unknown* concept, our model identifies a similar joint embedding of *known* concepts from the training set. Since, visual/semantic examples of *unknown* concepts are not available to us during training, first we learn a generic attention function $\mathcal{A}$ that can transfer knowledge from the *known* concepts to *unknown*. The attention function is learned on the training set, where it identifies the useful features from closely related exemplars to answer questions. The function $\mathcal{A}$ is agnostic to specific IQ pairs and provides a generalizable mechanism to identify relevant information from related examples. Therefore, at inference time, we use the same exemplar based attention function to obtain refined attention maps by using the closely related joint embedding of *known* concepts. We design the training schedule in two stages to facilitate knowledge transfer. During the **first** stage, only the Visual-Semantic embedding part of the network is trained end-to-end and the joint feature embedding tensor $\boldsymbol{e}$ is stored in memory $\boldsymbol{\xi} \in \mathbb{R}^{d \times n}$, where n is the number of training IQA triplets and d denotes the embedding dimension. In the **second** stage, both the visual-semantic embedding and the exemplar-embedding segment of the model are trained end-to-end where the model performs a nearest neighbour (NN) search on $\boldsymbol{\xi}$ to find the most similar joint embedding $\boldsymbol{e}_{\xi}$. Further, the network learns to use the exemplar embedding to refine the attention on visual scene details, which can be represented as:

$$\tilde{\boldsymbol{v}}_{\mathcal{E}} = \mathcal{A}(\boldsymbol{v}, \boldsymbol{e}_{\mathcal{E}}), \text{ where, } \boldsymbol{e}_{\mathcal{E}} = \mathcal{N}(\boldsymbol{e}, \boldsymbol{\xi}, \boldsymbol{\kappa}). \tag{5.2}$$

The embedding $\boldsymbol{e}_{\mathcal{E}}$ is found using NN search ($\mathcal{N}$) on a set of compact embeddings $\boldsymbol{\kappa}$.

There are two main motivations for not performing the NN search directly on $\boldsymbol{\xi}$ and instead using a set of compact embeddings $\boldsymbol{\kappa}$. Firstly, searching in the joint embedding space would allow the model to overfit when searching for the closest match. However, when searching for the closet match for a compact representation of the joint embedding, the reduced dimensionality of the representation avoids overfitting. Secondly, storing and performing NN search directly on the joint embedding exemplar space is extremely memory and time intensive. For example, if the top 36 bounding box proposal are selected from each image while evaluating on the VQAv2 [Goyal et al., 2017] dataset, $\boldsymbol{\xi}$ will be $\mathbb{R}^{n \times d}$ dimensional where $n \approx 400K$ training examples and $d = t_e \times 36$ and $t_e = 2048$ for a standard setting. Doing a similarity match on such a large space has practical memory and computational limitations.

Due to the above-mentioned reasons, we generate a coarse representation of $\boldsymbol{\xi}$ by passing each of its elements through a max-pooling layer. We empirically found max-pooling to perform well in our case. The set of max-pooled embeddings is represented by $\boldsymbol{\kappa}$, whose entries act as *soft-keys* for the exemplar-embeddings. When a query embedding $\mathcal{E}$ is presented, we calculate its compact feature $\boldsymbol{e}^{\kappa}$ by applying a max-pooling operation. The NN search is performed between $\boldsymbol{e}^{\kappa}$ and each element of $\boldsymbol{\kappa}$ to find the embedding $\boldsymbol{e}^{\kappa}_{\mathcal{E}}$. As the elements of $\boldsymbol{\xi}$ and $\boldsymbol{\kappa}$ have a one-to-one relationship, by matching the maxpooled version of $\boldsymbol{e}$ to $\boldsymbol{\kappa}$, the model finds the exemplar embedding $\boldsymbol{e}_{\mathcal{E}}$. We impose a constraint on NN search to make sure that only ex-

emplars that are less than a specified distance threshold d_{th} are selected. If there is no match within d_{th}, the model only uses the input joint embedding. This ensures that only relevant exemplars are selected for influencing the attention mechanism. Notably, with this setup, we do not require the large set of exemplars $\boldsymbol{\xi}$ to be loaded into memory, instead a much more compact representation is used for efficient search and retrieval.

Exemplar Implementation: To generate the compact embedding or *soft-key* set $\boldsymbol{\kappa}$, spatial 2D Max Pooling on a sliding window is applied on each entry of $\boldsymbol{\xi}_i \in \mathbb{R}^{36\times2048}$ to generate the compressed embedding $\kappa_i \in \mathbb{R}^{\rho}$ for $\boldsymbol{\xi_i}$. For our experiments we set $\rho = 98$. We represent $\boldsymbol{\kappa}$ using a K-D tree data structure. During testing and the second stage of training, we query on $\boldsymbol{\kappa}$ to find the index of the closest match to the max-pooled $\boldsymbol{e}$ and get the joint embedding $\boldsymbol{e}_{\xi}$ from $\boldsymbol{\xi}$ for that index. Ideally, one should perform NN search on all entries of $\boldsymbol{\xi}$, however it was found that if we randomly select 15% of exemplars to create our knowledge base, we get the optimum results considering computational resources and query time on a long K-D tree (ablation experiments in Sec. 5.4.4 and Fig. 5.5).

5.2.3 Visual Attention

Attention is applied in two steps in our proposed network. The first step consists of applying spatial and channel attention on $\boldsymbol{e}_{\mathcal{I}}$ and $\boldsymbol{e}_{\boldsymbol{\mathcal{E}}}$, which represent the JE generated from the given IQ pair and JE of the closest exemplar from the knowledge base. Our model approximates the overall attention using a dual self-attention approach on the JEs that employs both channel and spatial attention mechanisms. To this end, the JEs are passed though fully connected (FC) layers followed by a softmax layer to generate spatial $\boldsymbol{\alpha}^s \in \mathbb{R}^S$ and channel $\boldsymbol{\alpha}^c \in \mathbb{R}^C$ attention vectors. Thus, spatial and channel attention vectors for the JE of a given IQ pair are approximated as $\boldsymbol{\alpha}^s_{\mathcal{I}}$ and $\boldsymbol{\alpha}^c_{\mathcal{I}}$, and for the JE of retrieved exemplar ($\mathcal{E}$) as $\boldsymbol{\alpha}^c_{\mathcal{E}}$ and $\boldsymbol{\alpha}^c_{\mathcal{E}}$, respectively. These attention vectors signify complementary spatial and channel features generated using a given IQ pair and the most similar visual-semantic embedding.

During the second step, all attention vectors are used to take a weighted sum of the input visual feature to create a refined representation $\tilde{v}$ that represents salient visual cues:

$$\begin{aligned}
\tilde{\boldsymbol{v}}^s_{\mathcal{I}} &= \sum_j \boldsymbol{v}_j \alpha^s_{\mathcal{I}_j}, \;\; \tilde{\boldsymbol{v}}^c_{\mathcal{I}} = \sum_k \boldsymbol{v}_k^{\mathsf{T}} \alpha^c_{\mathcal{I}_k}, \quad \text{and} \\
\tilde{\boldsymbol{v}}^s_{\mathcal{E}} &= \sum_j \boldsymbol{v}_j \alpha^s_{\mathcal{E}_j}, \;\; \tilde{\boldsymbol{v}}^c_{\mathcal{E}} = \sum_k \boldsymbol{v}_k^{\mathsf{T}} \alpha^c_{\mathcal{E}_k},
\end{aligned} \tag{5.3}$$

where $j \in [1, 36], k \in [1, 2048]$ represent indices along spatial and channel dimensions respectively, $\boldsymbol{v}_j$ denotes the vector from matrix $\boldsymbol{v}$, $\tilde{\boldsymbol{v}}^s_{\mathcal{I}}$ and $\tilde{\boldsymbol{v}}^c_{\mathcal{I}}$ represent spatial and channel visual cues generated from $\boldsymbol{e}_{\mathcal{I}}$, and $\tilde{\boldsymbol{v}}^s_{\mathcal{E}}$ and $\tilde{\boldsymbol{v}}^s_{\mathcal{E}}$ represent spatial and channel visual cues generated from $\boldsymbol{e}_{\mathcal{E}}$. Such complementary visual cues are leveraged by the model to reason about unknown concepts using the attention calculated from the combined effect of the input IQ pair and further refine it by looking at the closest

example from the exemplar set. Our model learns to combine these four visual cues that generate an overall attended visual feature $\tilde{v}$ and again apply Φ in a similar manner as described in Sec. 5.2.1 to generate the final vision-semantic embedding. We then project the embedding to the prediction space that is passed through a classifier to predict the final answer $a^* \in \boldsymbol{A}$.

5.3 OW-VQA Dataset Generation

When a VQA system is subjected to an open-world setting, it can encounter numerous visual and semantic concepts that it has not seen during training. To help VQA systems develop capability to handle unknown visual and semantic concepts, we propose a new dataset that separates *Known-Unknown* concepts for training and testing respectively. Our dataset generation protocol builds on the fact that images in VQA datasets [Antol et al., 2015; Goyal et al., 2017] are re-purposed from MSCOCO images [Lin et al., 2014] and paired with crowd sourced Q&A. Even though, MSCOCO images have rich object level annotation for 12 super-categories and 80 object categories, the VQA dataset annotations include only information related to Q&A, excluding any link to object level annotation. This constitutes a significant knowledge gap which if addressed, would allow for more subtle understanding of the scene even if it contains previously unknown visual and semantic concepts. To bridge the gap, we propose to use objects categories as the core entity to develop a true *Known-Unknown* split that constitutes both visual and semantic domains. During the first stage, we propose an *Known-Unknown* split for MSCOCO object categories, which leads to a well-founded split that separates known/unknown concepts in IQA triplets on VQA datasets.

5.3.1 Known–Unknown Object Separation

At the first stage, from each MSCOCO super-category (except for `person` which has no sub-category), we select the rarest category as *unknown* and the rest as *known*. This choice is motivated by the fact that rare classes are most likely to be *unknown*. For each category c, we calculate N_i and N_t which represent the total number of images that c appears in, and total number of instances of c respectively. These statistics are calculated after merging the MSCOCO Train2014 and Val2014 splits. We define occurrence measure $N = N_i \times N_t$ for each category and select the category with the smallest N as an *unknown* category, which ensures that categories which appear in the least number of images, the least number of times are selected as *unknown*.

Such a measure is particularly necessary for datasets that are used to perform high level vision tasks associated with a language component. For example, in super-category `vehicle`, `train` is less frequent compared to `airplane` in terms of instances (4,761 vs. 5,278). If the split was solely based on number of instances N_t, then `train` would have been selected as the *unknown* category even though it appears in 662 fewer images than `airplane`. When human annotators are tasked with generating language components (i.e., Q&A or captions), the rarest language cues are often

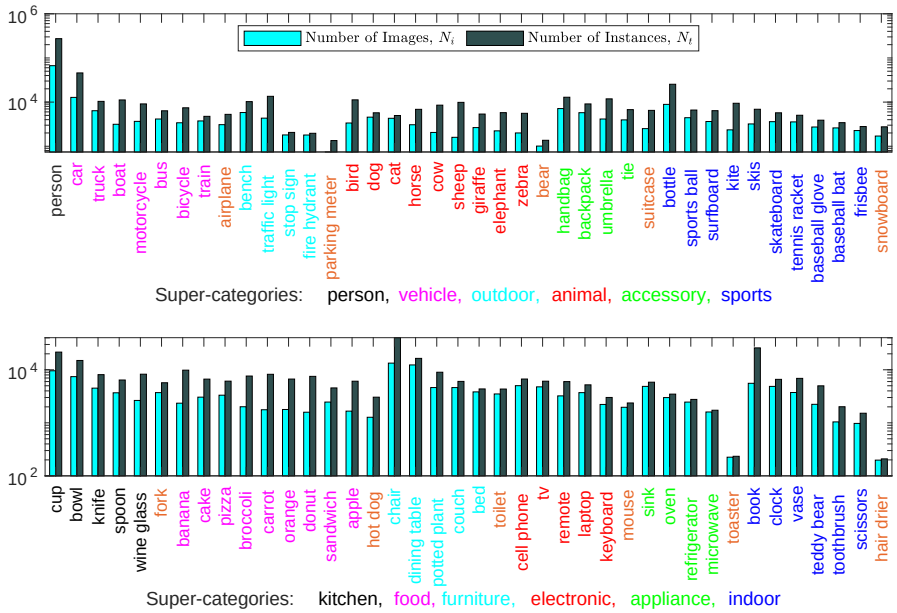


Figure 5.3: *Number of images* $\mathbf{N_i}$ *and number of instances* $\mathbf{N_t}$ *of each object category in the MSCOCO dataset (training and validation set combined).* $\mathbf{N_i}$ and $\mathbf{N_t}$ are used to determine the rarest object category from each super-category as the *unknown*.

associated with categories that appear less frequently. Thus, selecting the category with the least occurrence measure N ensures that categories with least language representation are selected as *Unknown*. Fig. 5.3 shows N_i and N_t for categories within each super-category of MSCOCO [Lin et al., 2014]. The category names are color-coded to represent the super-category labels and respective *unknown* categories. Additionally, Fig. 5.4 shows the normalized version of occurrence measure N for categories in each super-category and respective *unknown* categories.

5.3.2 Known–Unknown IQA Triplet Separation

During the second stage, we build on the *Known-Unknown* object categories to separate Image-Question-Answer (IQA) triplets. We re-purpose IQA triplets from VQAv1 [Antol et al., 2015] and VQAv2 [Goyal et al., 2017], and propose training (*known*) and test (*unknown*) splits for OW-VQAv1 and OW-VQAv2 dataset. For this purpose, we combine training and validation sets of respective VQA datasets (test split cannot be used as they are not publicly available). We employ a two step process to ensure that both visual and semantic concepts associated with the *unknown* are completely absent in the training set. **First**, we place an IQA triplet in the training set if there is no instance of any unknown object category in the image. This ensures that the new visual concepts are not known to the model during training. **Second**, we focus on

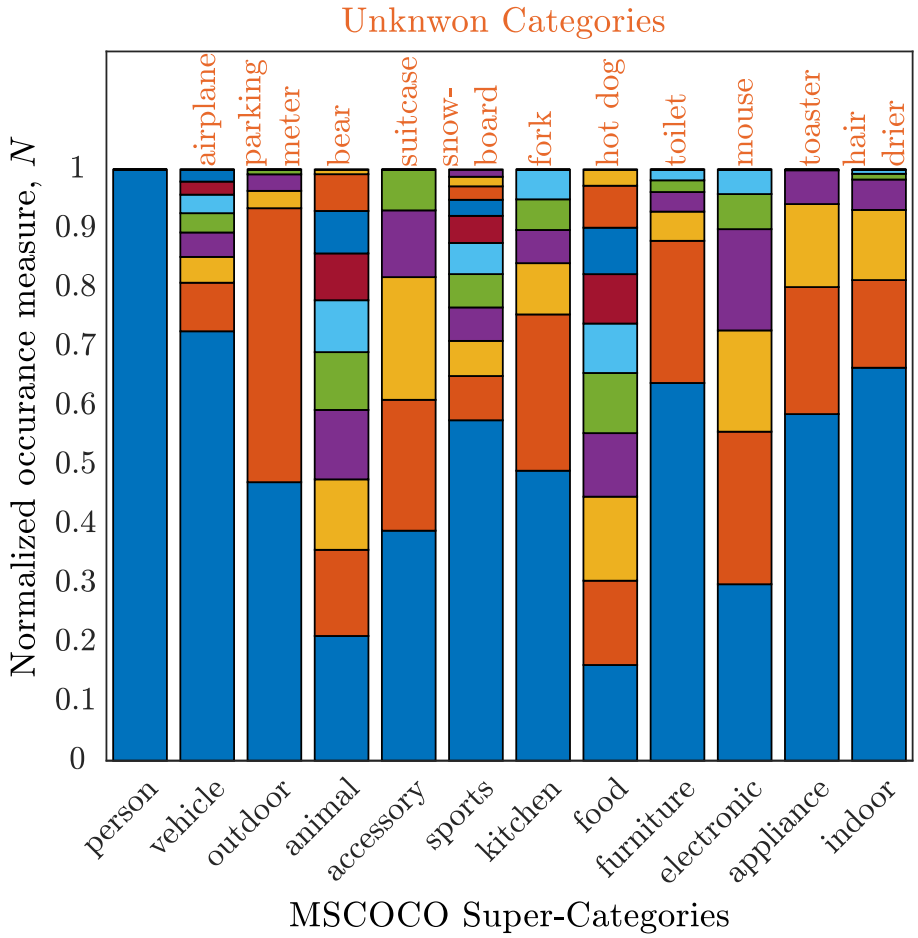


Figure 5.4: *Normalized occurrence measure N of object categories in each super-category.* The *Unknown* categories have the least representation in each super-category.

the semantic part and filter out the IQA triplets from the training set that have any unknown object category names or its synonyms of in the questions or answers. This ensures that even though an unknown object category is not present visually, the training set also does not contain any semantic cues of the object which the model might use as a supervision signal during training.

Tab. 5.1 presents statistics of VQA datasets[Antol et al., 2015; Goyal et al., 2017] following our proposed *known–unknown* concept separation protocol. We can see from the table that *unknown* object categories are present in ∼16% of training and validation images. Furthermore, it can be observed, when IQA triplets from the training and validation splits of the VQA datasets are separated on the basis of *known* and *unknown* semantic concepts, the *unknown* IQA triplets also amount to ∼16% of the total. This is an indication that our dataset preparation protocol uniformly separates *known* and *unknown* semantic concepts even from crowd-sourced, complex, multi-modal dataset like VQA.

Such uniform visual and semantic confinement of concepts in train/test split

		Training Split			
VQA Dataset		Total	Known	Unknown	Unknown%
Images	v1 & v2	82,783	69,557	13,226	15.98
IQA Triplet	v1	224,040	187,986	36,054	16.09
IQA Triplet	v2	402,691	336,124	66,568	16.53
		Validation Split			
VQA Dataset		Total	Known	Unknown	Unknown%
Images	v1 & v2	40,504	34,137	6,367	15.72
IQA Triplet	v1	120,916	101,815	19,101	15.80
IQA Triplet	v2	213,266	178,321	34,945	16.39

Table 5.1: *VQA dataset statistics based on our proposed* known *and* unknown *splits.* Following our proposed dataset generation protocol (Sec. 5.3), we are able to separate visual (i.e., image) and visual+semantic (i.e., IQA triplet) concepts into proportionate *known* and *unknown* splits. Percent of *unknown* Images and IQA triplets are $\sim$ **16**% for training and validation split for both versions of the VQA dataset.

is a major advantage of our proposed dataset over other approaches [Ramakrishnan et al., 2017; Teney and Hengel, 2016; Agrawal et al., 2018], where the *unknown* 'objects/concepts' are only defined at semantic-level. For example, `airplane` is an '*unknown* category' in our proposed dataset and a 'novel object' in the dataset proposed by Ramakrishnan et al. [2017]. A semantically motivated protocol would place an IQA triplet, without keyword `airplane` in the question, in the training set. However, there are several IQA triplets in VQA dataset that shows an `airplane` being serviced by a car, truck or a person at an airport, and do not ask about the `airplane`. Just ensuring that the semantic concepts are not present during training only addresses a naive version of the challenge an open-world VQA system would face.

Dataset ($\rightarrow$) Split ($\downarrow$)		# Image	OWv1 # IQA	OWv2 # IQA
Trainset		69,557	187,986	336,124
Valset	Known	34,117	120,916	178,321
	Unknown	6,367	19,101	34,945
Testset		13,226	36,054	66,568

Table 5.2: *Train, Val and Test splits for proposed OW-VQAv1 and OW-VQAv2 dataset.*

5.3.3 Proposed OW-VQA Dataset Splits

The Trainset and Testset of OW-VQA dataset consists of *known* and *unknown* IQA triplets from corresponding Train splits of VQA datasets. We propose two validation splits called Valset-Known and Valset-Unknown from the Val splits of VQA datasets. The Valset-Known contains *known* IQA triplets and the *Valset-Unknown* contains *unknown* IQA triplets from the Valset of respective VQA dataset version. The subdivision of Valset into *known* and *unknown* splits allows evaluation on both concept types, which is an unique feature of our proposed dataset. Tab. 5.2 lists the number of image and IQA triplets of the proposed splits for the OW-VQAv1 and OW-VQAv2 datasets.

There are two ways to evaluate a models' performance on the proposed OW-VQA dataset: **(a)** For the purpose of debugging and running validation experiments, one can train a VQA model on OW-VQA Trainset and evaluate on Valset-Known or Valset-Unknown or the whole Valset. **(b)** For a more comprehensive evaluation, it is recommended to train a VQA model on the OW-VQA Trainset and evaluate on Testset. We benchmark our proposed model and other state-of-the-art VQA models in this setting (See Tab. 5.3).

5.4 Experiments

Here, we present the results of our experiments which includes benchmarking of VQA models on the OW-VQA dataset, ablation and performance analysis of our proposed model on semantically motivated VQA splits and standard VQA settings along with qualitative results.

5.4.1 Experimental Setup

Feature Extraction and Fusion: We use Bottom-Up features provided by Anderson et al. [2018] to represent visual features of the input image $\boldsymbol{v} \in \mathbb{R}^{S \times C}$, where S is the number of top bounding box object proposals and C is the dimension of each the object feature. We select 36 bounding box proposals and each with $2048d$ feature vector. Our architecture can be easily adapted to incorporate CNN extracted features (i.e., ResNet152 [He et al., 2016]) where $S = 196$, however, we only conduct our experiments with Bottom-Up features for this work. The semantic feature $\boldsymbol{q} \in \mathbb{R}^{2400}$ is generated in a manner similar to [Farazi and Khan, 2018; Ben-Younes et al., 2017; Fukui et al., 2016] where the question is encoded with skip-thought vectors [Kiros et al., 2015] and passed through GRUs. When generating the visual-semantic embedding, we set the output dimensions equal to C to get a JE that has same dimension as the the input visual features.

Answer Classifier: We create the answer set $\boldsymbol{A}$ with the most frequent 3000 answers from the training set and formulate the VQA task as a multi-class classification problem on the answer set $\boldsymbol{A} \in \mathbb{R}^{3000}$ following VQA benchmark [Antol et al., 2015]. The final attended visual-semantic feature representation $\tilde{\boldsymbol{v}}$ is passed through a fully

connected layer to project to the answer embedding space where softmax cross entropy loss is applied to predict the most probable answer from $\boldsymbol{A}$.

5.4.2 Benchmarking VQA models on OW-VQA

We benchmark existing VQA models along with our proposed JE baseline and exemplar based JE+X model on both versions of the OW-VQA Testset. The JE baseline model applies spatial and channel attention on the input image-question joint embedding features, whereas our final JE+X model applies an additional spatial and channel attention on exemplar joint embedding. The models are trained on the Trainset and evaluted on Testset which is the recommended evaluation protocol for our proposed OW-VQA dataset. From Tab. 5.3, we can see VQA models that incorporate multimodal (visual-semantic) embedding (i.e., pooling [Fukui et al., 2016] or fusion [Ben-Younes et al., 2017]) compared to the models which only use semantic embedding to generate visual attention, achieve higher performance in both versions of OW-VQA. Our exemplar based approach further refines the visual attention by transferring knowledge from the exemplar set and we report 0.9% and 0.7% overall accuracy gain over the closet state-of-the-art method on OW-VQAv1 and v2 respectively. Such an improvement without using any external knowledge base (i.e., complementary training on Visual Genome [Krishna et al., 2016], external image and text corpora) and/or model ensemble justifies our approach of transferring knowledge from exemplars. Furthermore, the accuracy scores of VQA models reported in Tab. 5.3 drop significantly when evaluated on OW-VQAv2 compared to v1 as the IQA triplets in v2 have less language bias. Interestingly, our exemplar based JE+X model performs poorly compared to our baseline JE model, when answering *Number* questions (i.e., how many?). This is because, for *Number* questions, the closest exemplar may provide misleading information that negatively affects the accuracy.

Both versions of the OW-VQA dataset have a validation split where one can train a model only on known concepts and evaluate on Valset (Standard VQA setting with *known* + *unknown* concepts), Valset-Known or Valset-Unknown. In Tab. 5.3 we also report accuracy of the JE and JE+X model on OW-VQAv2 Valset (Kn+Unk) and OW-VQAv2 Valset-Unknown (Unk). It can be observed that when incorporating only spatial and channel attention (JE baseline), the Kn+Unk and Unk accuracy is increases by 0.3% and 0.1% respectively from the MUTAN baseline. Further, when incorporating exemplar information in JE+X model, the Kn+Unk and Unk accuracy is further increased by 0.2% and 0.7% compared to JE baseline. This shows that the exemplar feature indeed encapsulates valuable information which provides a performance boost in VQA setting, and is more useful when tasked with answering questions about *unknown* concepts (3.5x more accuracy gain compared to standard VQA setting).

[1]Compared with k=1, where only one nearest neighbour was used.

OW-VQA Dataset (→)	v1-Testset				v2-Testset				v2-Valset	
Model (↓)	All	Y/N	Num	Other	All	Y/N	Num	Other	Kn+Unk	Unk
JE+X(Ours)	**61.4**	**80.5**	42.0	**48.3**	**58.6**	**76.7**	40.2	**47.8**	**60.4**	**55.2**
JE(Ours)	60.9	80.4	**42.1**	47.7	58.3	75.7	**40.5**	47.3	60.2	54.5
MUTAN [Ben-Younes et al., 2017] + BU	60.5	80.0	41.3	47.3	57.9	75.3	39.5	47.2	59.9	54.4
MCB [Fukui et al., 2016]	59.7	73.1	36.9	46.1	55.5	71.8	35.5	45.7	-	-
SAN [Yang et al., 2016]	55.7	76.0	40.2	39.8	50.6	67.2	34.5	39.4	-	-
HieCoAtt [Lu et al., 2016b]	55.6	77.3	42.1	37.7	50.7.	67.4	35.1	38.5	-	-
VQA [Antol et al., 2015]	54.1	77.3	37.2	35.9	49.8	68.1	37.1	35.7	-	-

Table 5.3: *Evaluation on proposed OW-VQAv1-Testset, OW-VQAv2-Testset and Valset.* All models reported in this comparison use ResNet [He et al., 2016] to extract image-level visual feature.

Dataset (→)	VQA-CPv1				VQA-CPv2				Novel-VQA			
Model (↓)	All	Y/N	Num.	Oth.	All	Y/N	Num.	Oth.	All	Y/N	Num.	Oth.
JE+X (Ours)	**39.7**	44.4	**12.9**	**45.9**	**39.0**	40.6	**12.7**	46.0	**54.1**	**79.8**	**39.0**	**40.7**
JE (Ours)	39.3	44.0	12.9	45.5	38.4	37.2	12.6	**46.4**	53.8	78.9	38.7	40.2
MUTAN[Ben-Younes et al., 2017] + BU	39.4	44.0	12.8	45.5	38.1	38.3	12.6	45.5	53.7	79.4	37.5	40.1
GVQA [Agrawal et al., 2018]	39.2	**64.7**	11.9	24.9	31.3	**58.0**	11.7	22.1	-	-	-	-
MCB [Fukui et al., 2016]	34.4	38.0	11.8	39.9	36.3	41.0	12.0	40.6	-	-	-	-
SAN [Yang et al., 2016]	26.9	35.3	11.3	24.7	25.0	38.3	11.1	27.7	-	-	-	-
Novel Arch-1[Ramakrishnan et al., 2017]	-	-	-	-	-	-	-	-	41.8	76.6	28.5	25.7
VQA [Antol et al., 2015]	23.5	34.5	11.4	17.4	19.8	34.3	11.4	14.4	39.4	74.0	27.5	23.1

Table 5.4: *Evaluation on VQA-CP[Agrawal et al., 2018] and Novel-VQA [Ramakrishnan et al., 2017] dataset.* All models reported in this comparison use ResNet [He et al., 2016] to extract image-level visual features.

VQAv2 Val-set →	All	Y/N	Num	Other
JE+X (Exemplar based Model)	**62.2**	**79.9**	38.0	**51.9**
JE (Spatial+Channel Attention)	61.9	79.8	38.3	51.5
JE Spatial (Only Spatial Attention)	59.9	79.4	**41.1**	51.3
JE Channel (Only Channel Attention)	58.2	78.6	40.9	47.3
MUTAN [Ben-Younes et al., 2017] + BU	61.6	79.2	37.8	51.3
Support-Set[Teney and van den Hengel, 2018]	59.9	-	-	-
MCB [Fukui et al., 2016]	59.1	77.3	36.7	51.2
HieCoAtt [Lu et al., 2016b]	54.6	71.8	36.5	46.3
DCN+LQIA[Patro and Namboodiri, 2018][1]	53.3	70.6	34.6	44.4
SAN [Yang et al., 2016]	52.0	68.9	34.6	43.8
GVQA[Agrawal et al., 2018]	48.2	72.0	31.2	34.7

Table 5.5: *Ablation on VQAv2 Validation set.*

5.4.3 Evaluation on semantically separated VQA splits:

We evaluate our exemplar based approach on semantically motivated VQA-CP [Agrawal et al., 2018] and Novel VQA [Ramakrishnan et al., 2017] datasets where the former separated the challenging semantic concepts in the Testset and the latter placed least frequent nouns and associated IQA triplets in the testset. Although, our motivation is orthogonal and our definition of *Novel Concepts* is heterogeneous to these semantically motivated approaches, we showcase the effectiveness of our exemplar based approach on their settings as well. In Tab. 5.4, we evaluate our baseline JE and exemplar based JE+X and model on both versions of the VQA-CP dataset and report performance against other benchmarks and their proposed GVQA [Agrawal et al., 2018] dataset. JE and JE+X outperforms other contemporary models on the VQA-CPv1 dataset on Overall Accuracy, Number and Other Question. GVQA [Agrawal et al., 2018] employs separate question classifiers for Y/N and non-Y/N (i.e., Num, Other) questions that account for its high accuracy in Y/N questions which results in higher Overall accuracy for binary questions. However, when evaluated on both VQA-CPv1 and v2, JE+X outperforms GVQA in Other question accuracy by a significant margin (21% and 23.9%). We further evaluate our models on the Novel-VQA [Ramakrishnan et al., 2017] dataset. Our exemplar based approach outperforms the best variant of Novel Arch-1 which includes external knowledge, both semantic (i.e., books) and visual (i.e., examples from ImageNet [Deng et al., 2009]) by 12.3% and MUTAN baseline by 0.4%.

5.4.4 Ablation study on standard VQA setting

We evaluate different variants of our model on the VQAv2 validation set [Goyal et al., 2017] to perform an ablation study and compare its performance with other attention based models. It is worth noting that we only compare with their single model without data augmentation which is similar to our setting for fair comparison.

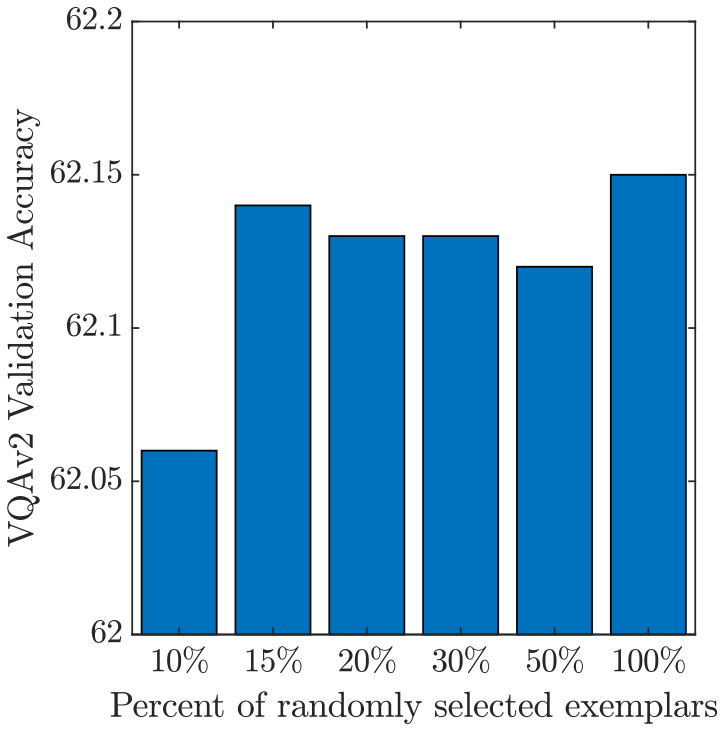


Figure 5.5: *VQA Accuracy vs. Percentage of randomly selected exemplars.*

From Tab. 5.5, it can be seen that our baseline JE and final JE+X model outperforms the state-of-the-art Tucker decomposition based MUTAN model [Ben-Younes et al., 2017] with BU [Anderson et al., 2018] attention, which has a similar multimodal fusions operation to our baseline model. Further, it also outperforms the Support-Set model [Teney and van den Hengel, 2018] in a similar setting where the support set contains example representation of question, answers and image. It is to be noted that some recent models have more powerful reasoning [Zhang et al., 2018; Wu et al., 2018] or fusion [Ben-Younes et al., 2019] mechanism. We aim to demonstrate the effectiveness of the discriminative features of joint embedding space which can be useful for equipping most VQA models in an Open-World setting.

Further, we report the VQA accuracy of spatial or channel attention only variant of the baseline JE model in Tab. 5.5. The model with spatial attention achieves 59.9 overall accuracy whereas the model with channel attention achieved 58.2. This observation provides a couple of important insights about the Joint-Embedding feature space. **First,** the JE space encapsulates multimodal information in a way similar to how deep CNNs represent visual features; applying either spatial or channel attention on JE space yields reasonable VQA accuracy. The accuracy is higher when spatial attention is applied on the JE features than channel attention, because the spatial information encapsulated in the spatial attention has higher importance for the VQA task. **Second,** applying spatial and channel attention on JE space provides

complementary information that a VQA model can learn to answer questions more accurately. By leveraging the complementary nature of the these two attention modules our JE model achieves the overall best accuracy among other variants of our approach in a traditional VQA setting.

Our proposed JE+X model used a knowledge base built from only 15% of randomly selected exemplars (also discussed in Sec. 5.2.2). In Fig. 5.5 we report the overall accuracy in the VQAv2 Validation dataset varying the percent of randomly selected exemplars. Notably, increasing the number of exemplars results in a more computationally expensive model. We can see that VQA accuracy increases by only a small amount when we used the whole exemplar set as a knowledge base compared to when we randomly selected 15% of the exemplars. Further, the accuracy saturates when a percentage higher than 15% of exemplars is used for knowledge transfer. Thus, we used 15% randomly selected exemplars to build our knowledge base as the right compromise between performance and efficiency. Using a small percentage of exemplars enables a more computationally efficient search and retrieval of exemplars while yielding superior VQA accuracy.

5.4.5 Qualitative results

We report some qualitative results in Fig. 5.6 of our baseline JE and exemplar based JE+X models evaluated on OW-VQAv2 Testset. It can be seen that for a given image I and question IQ, JE+X finds an exemplar image X and question XQ that had the most similar representation in the question-image joint embedding space. We visualize only the spatial attention for JE and JE+X models as visualizing channel attention is more ambiguous when the input image features are bounding box features of object or object parts. It can be seen that when the JE model is asked *Is this a fancy restaurant?* (second row of Fig. 5.6) the spatial attention is focused on the chairs at the back table. Our JE+X model finds an exemplar where the question asked *'Is this a vegan meal?'*, which allows the model to generate a complementary attention on the bottles stacked on the table. Such complementary attention from exemplar helps the model to focus on the subtle visual cues that are an indicator of the restaurant not being a fancy one. We also show some failure cases in the Fig. 5.6 3rd row. The first example showcases where the model is unable to recognize high level semantic cues even with an exemplar. The second example shows by using an exemplar the model gets confused when answering counting question as there are no explicit reasoning modules for this type of questions.

5.5 Conclusion

Existing VQA systems lack the ability to generalize their knowledge from training to answer questions about novel concepts encountered during inference. In this chapter, we propose an exemplar-based transfer learning approach that utilizes the closest *known* examples to answer questions about *unknown* concepts. A joint embedding space is central to our approach, that effectively encodes the complex relationships

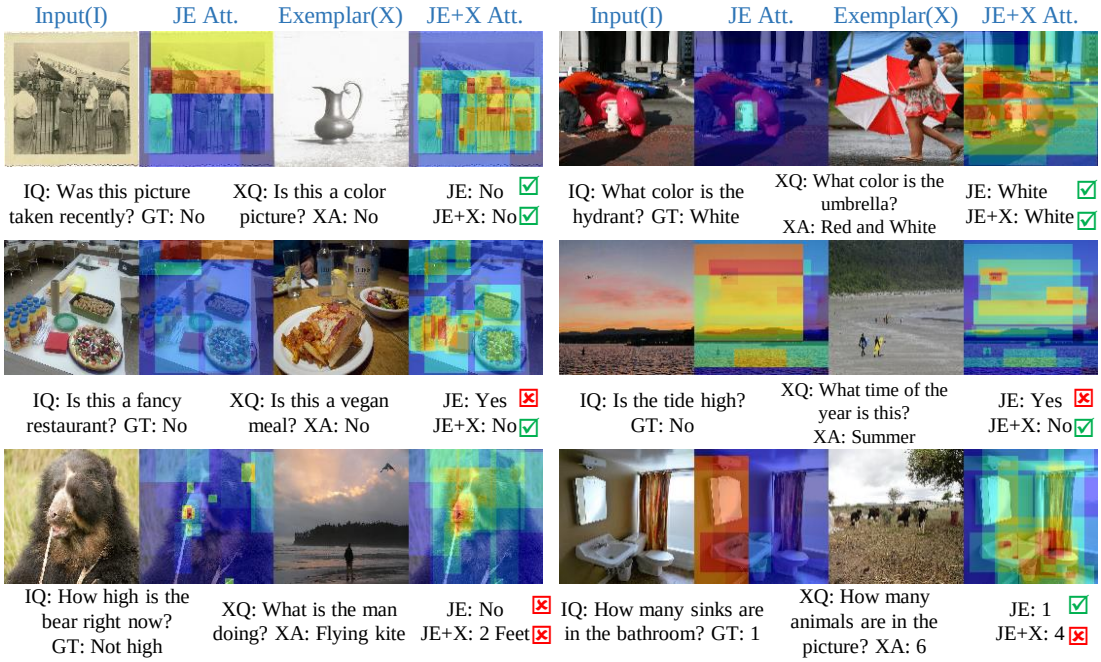


Figure 5.6: *Qualitative results of our baseline JE and exemplar based JE+X model.* Predicted answers and attention maps evaluating JE+X model on OW-VQAv2 Valset-Unknown images. The Grid Attention map (GA) and Exemplar Attention(EA) map provides complementary information for the model to reason about Unknown concepts, where only using GA or EA often leads to wrong prediction.

between semantic, visual and output domains. Given the IQ pair and exemplar embedding in this space, the proposed approach hierarchically attends to visual details and focuses attention on the regions that are most useful to predict the correct answer. We propose a new Open-World VQA dataset to fairly compare the performance of VQA systems on *known* and *unknown* concepts. Our exemplar based approach achieves significant improvements over the state-of-the-art techniques on the proposed OW-VQA setting as well as standard VQA setting, which reinforces the notion of transferring knowledge from rich joint embedding space to reason about *unknown* concepts.

Chapter 6

Semantic Relationship Parsing

In the previous chapters, we designed VQA models that can reason based on visual feature representation. However, Humans explain inter-object relationships with semantic labels that demonstrate a high-level understanding required to perform visual-linguistic task such as Visual Question Answering (VQA). Some of the existing VQA models represent relationships as a combination of object-level visual features which constrain a model to express interactions between objects in a single domain, while the model is trying to solve a multi-modal task. In this chapter, we propose a general purpose semantic relationship parser which generates a semantic feature vector for each subject-predicate-object triplet in an image, and a Mutual and Self Attention (MSA) mechanism that learns to identify relationship triplets that are important to answer the given question. To motivate the significance of semantic relationships, we show an oracle setting with ground-truth relationship triplets, where our model achieves a 25% accuracy gain over the closest state-of-the-art model on the challenging GQA dataset. Further, with our semantic parser, we show that our model outperforms other approaches on VQA and GQA datasets.

6.1 Introduction

Humans can perform high-level reasoning over an image by seamlessly identifying the objects of interest and associated relationships between them. Although objects are central to scene interpretation, they cannot be independently used to develop a holistic understanding of the visual content without considering their mutual relationships. The multi-modal reasoning task of Visual Question Answering (VQA) requires learning precisely encoded relationships between objects. Given the complexity of the task, we advocate for relationship modeling in the semantic space so that a given question can be directly related with the objects and relationships present in an image. Our choice is motivated by two observations. *First*, visual representations for different instances of the same semantic relationship can be very different, making it challenging for the VQA model to relate them with the asked question. *Secondly*, different semantic relationship interpretations can exist for a single visual representation, thereby requiring an enriched mechanism to encode a diverse set of semantic relationships. If a relationship parser can automatically derive representa-

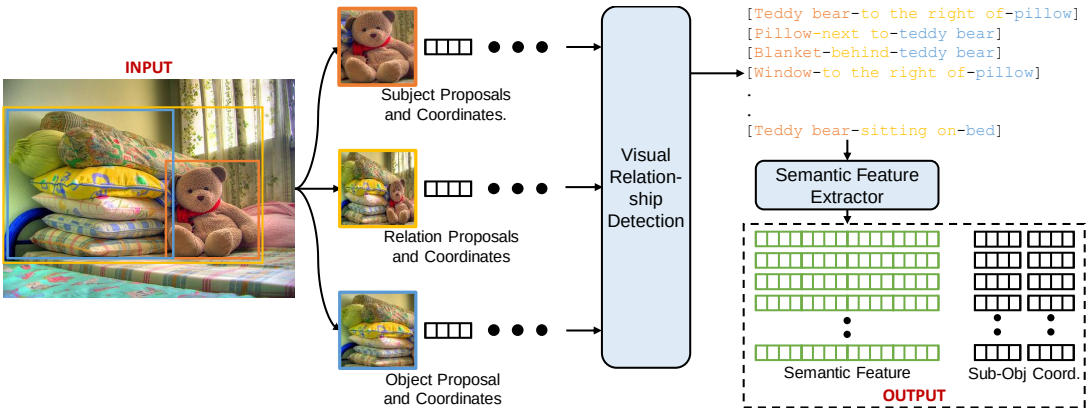


Figure 6.1: *Our proposed Semantic Relationship Parser (SRP).* Given an image, the SRP module generates relationship triplets through a relationship detector, which are then passed through a semantic feature extractor. Each semantic relationship feature is paired with subject and object box coordinates for visual grounding. Our approach is built upon semantic description of relationships, as opposed to a visual representation in existing works, which allows us to accurately model complex relationships.

tions in semantic space and attend to relevant relations, the above challenges can be simplified for VQA.

Based one the hypothesis that a better scene understanding requires a model to generate more discriminative visual and semantic feature representation, recent VQA models employ state-of-the-art visual [He et al., 2016; Hu et al., 2018; Ren et al., 2015b] and semantic [Pennington et al., 2014; Mikolov et al., 2013b; Devlin et al., 2018] feature extractors. Specifically, VQA models use information at grid-level [Antol et al., 2015; Ben-Younes et al., 2017; Yu et al., 2017], object-level [Anderson et al., 2018; Ben-Younes et al., 2019; Yu et al., 2019] or both [Nam et al., 2016; Farazi and Khan, 2018] to extract visual features in an image without considering the relationships between them. Some recent models address this problem by identifying the most relevant object pairs by learning an attention distribution over them with respect to the question [Cadene et al., 2019a; Li et al., 2019b; Hu et al., 2019]. This kind of relationship-aware models achieve better performance compared to the ones that do not consider any kind of relationship. However, as seen in the example shown in Fig. 6.1, the visual feature representation of `teddy bear` and `pillow` remains the same even though the relationship between them can be different (e.g., `on the right`, `near to`). For higher level reasoning, a visual-semantic model needs to identify these subtle differences, which can only be achieved if the model considers semantic relationship features.

While attempting to combine semantic relationship features, a VQA model faces three major challenges. First, the lack of a visual relationship detector that not only detects arbitrary relationships between object pairs in an image, but also generates semantic features for the detected subject-predicate-object triplets for a downstream

task. Second, the effectiveness of semantic relationship features over the visual relationship feature in a complex visual-linguistic task such as VQA is not investigated before. Third, an effective attention mechanism is required to combine rich features encoding visual, semantic and relationship information to predict the correct answer. In this chapter, we contribute towards bridging the gap by addressing these three main challenges. The main contributions of this chapter are:

- We propose a general purpose semantic relationship parser that can be used for multi-modal visual-linguistic downstream tasks such as Visual Question Answering.
- We showcase the effectiveness of using semantic relationship features by reporting superior performance over models employing similar visual relationship features. Further, in an oracle setting where ground-truth relationship labels are available, we obtain a 25% accuracy gain compared to a SOTA model that only uses visual features.
- We further propose a Mutual and Self Attention (MSA) mechanism that utilizes both mono-modal self-attention and multi-modal mutual-attention using visual features (from the image) and semantic features (from both question and relationships), and report superior accuracy on the VQAv2 and GQA datasets.

6.2 Methods

Given an image I and a natural language question Q, the task of a VQA model is to predict the answer $\hat{a}$. Let $\boldsymbol{v}$ and $\boldsymbol{r}$ be the collection of all visual features and semantic relationship features extracted from the image I, and $\boldsymbol{q}$ be the semantic feature representation of the question Q. The VQA problem is typically formulated as a multi-class classification problem:

$$\hat{a} = \underset{a \in \mathcal{A}}{\arg\max}\, p(a|\boldsymbol{v}, \boldsymbol{q}, \boldsymbol{r}; \boldsymbol{\theta}), \tag{6.1}$$

where $\boldsymbol{\theta}$ denotes the parameters of the model and $\mathcal{A}$ is a dictionary of candidate answers.

6.2.1 Question and Image Feature Extraction

The traditional approach [Fukui et al., 2016; Ben-Younes et al., 2017, 2019; Teney et al., 2018; Yu et al., 2019] for extracting question features for the VQA task is by sourcing pretrained semantic embedding vectors for each question words, concatenating and passing them through a recurrent neural network. The hidden state of the last recurrent block is extracted as the question feature. In contrast, we consider the question as a whole instead of separate word entities, thereby providing better contextual modelling. We use Bidirectional Encoder Representations from Transformers (BERT) [Devlin et al., 2018] where we first tokenize each word of the question and then feed

the tokenized question into a Transformer model pretrained for language modeling task. The question feature $\boldsymbol{q} \in \mathbb{R}^{m \times d_q}$ is extracted from the last hidden layer of the BERT model, where m is the number of tokens identified in the question and d_q denotes the feature dimension.

We represent the visual features of an input image as a set of bounding box coordinates and corresponding object-specific features. First, the object proposals are generated using a bottom-up [Anderson et al., 2018] attention approach where a pretrained Faster-RCNN [Ren et al., 2015b] model is employed the get the region proposals and extract visual features using a ResNet [He et al., 2016] backbone. Following [Anderson et al., 2018], we use an adaptive threshold to select a range of region proposals $l \in [10, 100]$ for each image. Further, to visually ground each region proposal we concatenate each region proposal with its bounding box coordinates. Thus, the visual feature representation $\boldsymbol{v} \in \mathbb{R}^{l \times (d_v + 4)}$ of image I consists of features of its object proposals $\{\boldsymbol{f}_j \in \mathbb{R}^{d_v}\}_{j=1}^{l}$ and corresponding bounding box coordinates $\{\boldsymbol{b}_j \in \mathbb{R}^4\}_{j=1}^{l}$, where d_v is the object feature dimension.

6.2.2 Semantic Relationship Parsing

The Semantic Relationship Parser (SRP) module is illustrated in Fig. 6.1. It has three major components. The **first** component is a region proposal network that operates in a similar manner as explained above in Sec. 6.2.1 for visual feature generation. Based on these object-wise features and box coordinates, a visual relationship detector is used in the **second** stage. The visual relationship detector generates subject, relationship and object[1] proposals from the region features which in-turn are used to generate a *semantic relationship triplet* set $\mathcal{T} = \{\boldsymbol{t}_k\}_{k=1}^{q}$. Each relationship triplet $\boldsymbol{t}_k$ consists of class labels predicted for subject, relationship, and object. In order to generate a triplet, a set of candidate subject, relationship and object visual features, denoted by $\boldsymbol{f}_s$, $\boldsymbol{f}_r$ and $\boldsymbol{f}_o$ respectively, are passed through the visual relationship detector. We follow the framework proposed by [Zhang et al., 2019b], where we assume a relationship exists only if a subject-object pair exists, not vice versa. Thus the relationship detector learns two mapping functions from visual feature space to semantic space, one for subject/object and the other one for relationship embedding.

The relationship feature embedding is generated by passing the concatenated version of three visual features $\boldsymbol{f}_s$, $\boldsymbol{f}_r$ and $\boldsymbol{f}_o$ through a two-layer Multi-layer Perceptron (MLP) network. The subject and object feature embeddings are generated in parallel by passing them through the same MLP network. On the other hand, class labels of subject, object and relationship are first converted to word vectors and then to semantic feature embeddings by passing them through a small MLP network. Three triplet losses are minimized [Zhang et al., 2019a] to match visual and semantic embedding for subject, object and relationship respectively. During inference, word vectors of all subject/object and relationship class labels are passed and a nearest neighbour search is performed to find the desired relationship labels. In practice, we perform the visual relationship detection on the input image as a pre-processing step where

[1]Here, the object refers to the grammatical component of a sentence.

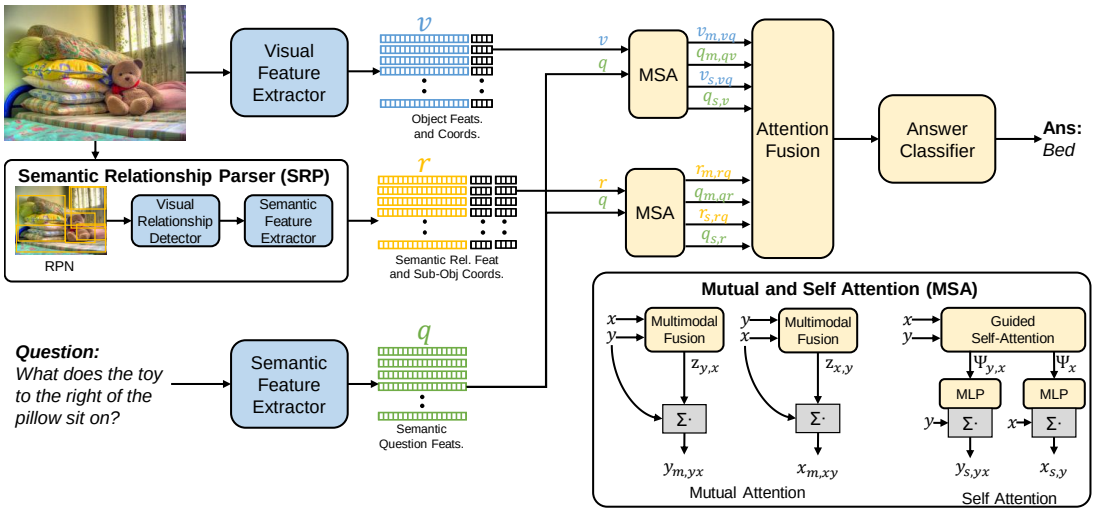


Figure 6.2: *Our proposed Mutual and Self-Attention (MSA) VQA model built on the Semantic Relationship Parser (SRP).* The question feature $\boldsymbol{q}$ is used alongside visual $\boldsymbol{v}$ and relationship $\boldsymbol{r}$ features to generate corresponding mutual and self-attended representations. These rich feature representations are then projected into a common embedding space and sum-pooled to predict the correct answer.

we train the visual relationship detector end-to-end on visual relationship dataset (i.e., Visual Genome [Krishna et al., 2016], VRD [Lu et al., 2016a]) and then run inference on the input images to generate relationship prediction, which consists of subject, object and relationship probability.

The **third** component of the SRP module is the semantic feature extractor which takes the *filtered* semantic relationship triplets $\mathcal{R}$, subject bounding box $\boldsymbol{b}_s$ and object bounding box $\boldsymbol{b}_o$ coordinates, and generates visually grounded semantic relationship features as follows,

$$\boldsymbol{r} = \mathcal{F}_{\text{BERT}}(\mathcal{R}, \boldsymbol{b}_s, \boldsymbol{b}_o). \tag{6.2}$$

Here, $\mathcal{F}_{\text{BERT}}$ denotes the BERT model (similar to the one described in Sec. 6.2.1) for extracting semantic features from the triplets. First, this function takes all entries from the set $\mathcal{R}$ and adds a period ('.') after each element. Each relation triplet $\boldsymbol{t}_k \in \mathcal{R}$ is now considered a complete sentence which is passed through the BERT model separately alongside its subject and object proposal coordinates. This generates corresponding semantic relationship features $\boldsymbol{r} \in \mathbb{R}^{n \times d_r}$ from image I, where d_r is the dimension of the hidden feature of the BERT model.

Notably, the set $\mathcal{R}$ only contains refined relationship triplets obtained after a two stage thresholding and filtering process. At the first stage, we filter out the relationship predictions where the probability of the product of subject and object proposals are higher than a threshold α. This ensures that we only select the high-confidence relationships between subject-object pairs. In the next stage, from the

remaining relationship predictions, we select only those whose relation probability is higher than β. Intuitively, we set α at a higher value compared to β to ensure that we first get the subject and object instances right, and ask the model to predict various relationships between them. The values of α and β are set empirically with the objective to select at least three relationship triplet per image. We further filter out any duplicate relationships and end up with 'n' relationship predictions per image encoded in refined semantic relationship set $\mathcal{R}$.

6.2.3 Mutual and Self Attention

The Mutual and Self Attention (MSA) module consists of two major components. The first component focuses on mutual attention where two separate multimodal fusion operations are performed to learn attention distribution over the input feature vectors. The second component applies self attention on the pair of input features and generates attention distribution over the input features themselves. We illustrate the MSA module in Fig. 6.2. For simplicity lets assume the input to the MSA module are a two feature embeddings $\boldsymbol{x} \in \mathbb{R}^{a \times d_x}$ and $\boldsymbol{y} \in \mathbb{R}^{b \times d_y}$, which will undergo mutual and self attention.

Mutual Attention: To capture the complex interaction between $\boldsymbol{x}$ and $\boldsymbol{y}$ we jointly embed these features by learning a multimodal embedding function. This is achieved by first concatenating the input features and then passing them through a 3 layer MLP network. The MLP model learns to capture the mutual interactions between the input feature vectors and produces a joint feature embedding. For input combinations $(\boldsymbol{y}, \boldsymbol{x})$ and $(\boldsymbol{x}, \boldsymbol{y})$, we have:

$$\boldsymbol{z}_{y,x} = \text{MLP}\ [\boldsymbol{y}\ \oplus\ \boldsymbol{x}] \in \mathbb{R}^{1 \times b}, \quad \text{and} \quad \boldsymbol{z}_{x,y} = \text{MLP}\ [\boldsymbol{x}\ \oplus\ \boldsymbol{y}] \in \mathbb{R}^{1 \times a}, \tag{6.3}$$

where $\oplus$ denotes the concatenation operation of the two vectors. $\boldsymbol{z}_{y,x}$ and $\boldsymbol{z}_{x,y}$ signifies the learned mutual attention distributions over the inputs $\boldsymbol{y}$ and $\boldsymbol{x}$ respectively. These attention distributions are used to take a weighted sum on the corresponding input feature vectors to generate mutually attended feature representation $\boldsymbol{y}_{m,yx}$ and $\boldsymbol{x}_{m,xy}$,

$$\boldsymbol{y}_{m,yx} = \sum_{i=1}^{b} (z^i_{y,x}\ \boldsymbol{y}^i) \in \mathbb{R}^{d_y}, \quad \text{and} \quad \boldsymbol{x}_{m,xy} = \sum_{i=1}^{a} (z^i_{x,y}\ \boldsymbol{x}^i) \in \mathbb{R}^{d_x}, \tag{6.4}$$

where, $\boldsymbol{x}^i$ and $\boldsymbol{y}^i$ denote the i^{th} row from the feature embeddings $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively.

Self Attention: For the self attention component, we follow the guided self attention module used in [Yu et al., 2017]. The input feature $\boldsymbol{x}$ is fed to a Transformer employing multi-head attention [Vaswani et al., 2017] to learn an attention distribution Ψ_x. The other input $\boldsymbol{y}$ undergoes a similar multi-head attention like $\boldsymbol{x}$, except the *query* input is replaced with Ψ_x, which allows the model to learn $\boldsymbol{x}$-guided self attention distribution over $\boldsymbol{y}$, denoted as $\Psi_{y,x}$. Ψ_x and $\Psi_{y,x}$ are passed through separate fully-connected layers for dimensionality reduction and we get $\Psi_x \in \mathbb{R}^a$ and $\Psi_{y,x} \in \mathbb{R}^b$. These self attention maps are used to take a weighted sum over the

corresponding feature representations and generate self attended features $\boldsymbol{y}_{s,yx}$ and $\boldsymbol{x}_{s,y}$,

$$\boldsymbol{y}_{s,yx} = \sum_{i=1}^{b} (\Psi_{y,x}^{i}\ \boldsymbol{y}^{i}) \in \mathbb{R}^{d_y}, \quad \text{and} \quad \boldsymbol{x}_{s,y} = \sum_{i=1}^{a} (\Psi_{x}^{i}\ \boldsymbol{x}^{i}) \in \mathbb{R}^{d_x}. \tag{6.5}$$

In practice, we employ two MSA modules, where we feed $\boldsymbol{q}$, $\boldsymbol{v}$ to the first one and $\boldsymbol{q}$, $\boldsymbol{r}$ to the other. The intuition behind this is the first MSA module learns to identify which region of the image, and words of the question are important to answer the question. Similarly, the second MSA module tries to identify the salient relationship features and question parts for answering the question. For both MSA blocks, we pass question features as $\boldsymbol{x}$ which guides the attention learning process of the input. This is particularly important as the question sets the objective of the task, and the quality of the learned attention distribution depends more on the question than the other inputs. Thus the first MSA module outputs $\boldsymbol{v}_{m,vq}, \boldsymbol{q}_{m,qv}, \boldsymbol{v}_{s,vq}, \boldsymbol{q}_{s,v}$ and the second one outputs $\boldsymbol{r}_{m,rq}, \boldsymbol{q}_{m,qr}, \boldsymbol{r}_{s,rq}, \boldsymbol{q}_{s,r}$.

6.2.4 Attention Fusion

We perform multimodal attention fusion on the outputs of the MSA blocks. Each attended feature is projected to an intermediate space through fully connected layers followed by summation. As the attended features already capture rich feature description, we only use such a simple linear summation technique to capture their interaction before making the final answer prediction. The summed feature vector is then projected to the answer prediction space $d^{|\mathcal{A}|}$ through another fully connected layer where we minimize a cross-entropy loss to predict correct answer from the candidate answer set.

6.3 Experiments

6.3.1 Dataset

We perform experiments on two large-scale VQA datasets, namely VQAv2 [Goyal et al., 2017] and GQA [Hudson and Manning, 2019]. The VQAv2 has 200K images and 1.1M crowed-sourced questions. This is the biggest manually annotated VQA dataset. Further, GQA contains 11K images and 22M auto-generated questions, making it a more challenging evaluation setting.

6.3.2 VQA Model Architecture

The visual feature are extracted using a Faster-RCNN [Ren et al., 2015b] network with ResNet101 [He et al., 2016] backbone with $d_v = 2048$ dimention for each object. To extract the semantic features from the question and relationship triplet, we use

Methods	GQA Validation Set					
	Acc.↑	Binary↑	Open↑	Validity↑	Plaus.↑	Dist.↓
MCAN [Yu et al., 2019]	65.00	82.08	48.98	94.91	91.42	4.21
$\boldsymbol{r}^{vis} + \boldsymbol{q}$	51.89	69.02	35.83	95.13	91.78	7.34
$\boldsymbol{r}^{sem} + \boldsymbol{q}$	50.37	63.66	37.91	95.03	91.83	13.06
$\boldsymbol{r}^{vis} + \boldsymbol{v} + \boldsymbol{q}$	58.62	73.25	44.91	94.95	91.05	12.63
$\boldsymbol{r}^{sem} + \boldsymbol{v} + \boldsymbol{q}$	65.93	82.35	49.27	94.98	91.57	4.88
$\boldsymbol{r}^{oracle} + \boldsymbol{q}$	68.71	71.84	68.71	94.94	92.99	7.29
$\boldsymbol{r}^{oracle} + \boldsymbol{v} + \boldsymbol{q}$	81.15	85.06	77.48	95.34	94.26	1.08

Table 6.1: *On establishing the benefit of semantic relationship parsing for VQA.* We note that using semantic relationship features gives better performance as compared to the visual relationship features (rows 2-5). To demonstrate the richness of semantic features, we also report the upper-bound (oracle case in the last two rows), where our model delivers an absolute gain of ∼16 accuracy points over the MCAN [Yu et al., 2019] model. $\boldsymbol{v}$, $\boldsymbol{q}$ represent visual features and question features respectively. $\boldsymbol{r}^{vis}$, $\boldsymbol{r}^{sem}$ and $\boldsymbol{r}^{oracle}$ represent visual relationship features, semantic relationship features and Oracle semantic relationship features respectively.

a pretrained `bert-large-cased`[2] model. Since a cased version is used, we do not convert the question or relationship triplets to lowercase. The extracted semantic feature dimensions for question and relationship are $d_q = d_r = 1024$. We train the visual relationship detector in the SRP module on VRD dataset with a VGG16 backbone, and use this pretrained model to infer relationship triplets. Following the recommendation of Vaswani et al. [2017] and Yu et al. [2017], the intermediate dimensions d of the multi-head attention in transformer module is set to 512 with 8 heads and latent dimension of 64. Adam optimizer [Kingma and Ba, 2014] with $\beta_1 = 0.9$ and $\beta_2 = 0.98$ is used.

6.3.3 Semantic vs. Visual Relationship Feature

In Tab. 6.1, we first establish the benefit of our proposed semantic relationship feature modeling. To this end, we compare the VQA performance between 'semantic' and 'visual' relationship features to showcase the comparative advantage on the GQA validation dataset. To develop the baseline model with visual relationship features, we train the SRP module (Sec. 6.2.2) on the VRD dataset [Lu et al., 2016a] with 100 objects and 70 predicate categories, and output visual feature of the subject and object relationship proposal alongwith relationship triplet. The visual feature of the subject and object proposals are concatenated and considered as visual relationship feature $\boldsymbol{r}^{vis}$, and the default semantic relationship features (denoted by $\boldsymbol{r}^{sem}$) are extracted

[2] https://huggingface.co/bert-large-cased

from the relationship triplet. The models in Tab. 6.1 employ only the guided self-attention part of the MSA module for simplicity.

Blind models trained with visual relationship feature perform slightly better. In Tab. 6.1, we see that a VQA model trained with only visual relationship features (row 2) performs better than the model trained only with semantic relationship features (row 3). This is because when the visual feature $\boldsymbol{v}$ is not available, the $\boldsymbol{r}^{sem} + \boldsymbol{q}$ model is *blind* to the image and the answer prediction is based only on the relationship labels. On the other hand, the $\boldsymbol{r}^{vis} + \boldsymbol{q}$ model can *see* the image as a set of the visual feature of subject-object proposals, thus performs better than the completely *blind* model (row 3). However, in this extreme setting, the *blind* model perform reasonably well only relying on the semantic relationship labels.

Non-blind models trained with semantic relationship feature perform significantly better. When the visual feature is available, the VQA model with the complementary semantic relationship feature performs significantly better (7.34 ↑) than its counterpart (rows 4, 5 in Tab. 6.1). This demonstrates the complementary effectiveness of the semantic relationship features, since both these settings are identical except for the nature of the relationship feature.

6.3.4 Oracle Setting

We simulate an oracle setting to further evaluate the effectiveness of using semantic relationships for VQA. We build this setting using scene-graph annotations available for GQA [Hudson and Manning, 2019] train and validation sets. Each scene-graph entry consists of ground-truth subject, relationship and object label. We use a scene-graph parser which converts each scene-graph entry into a list of semantic relationship triplets similar to the output of visual relationship detector of Sec. 6.2.2, and denote the extracted semantic ground-truth relationship features as $\boldsymbol{r}^{oracle}$.

Both blind and non-blind oracle models significantly outperform the SOTA. The *blind* VQA model with a ground-truth relationship label $\boldsymbol{r}^{oracle}$ achieves an overall accuracy gain of 3.71 ↑ compared to the state-of-the-art MCAN [Yu et al., 2019] model[3] which is a *non-blind* model (comparing rows 6 and 1 of Tab. 6.1). This is an interesting finding showing if good enough semantic relationship label are available, the VQA model could achieve better performance than SOTA without even *looking* at the image. Further, when visual feature of the image is available in the oracle setting, the model achieves 16.43($\sim$ 25%) accuracy gain over [Yu et al., 2019].

'Open-ended' questions are answered better. Both oracle models report significant accuracy gain (19.73 ↑ and 28.5 ↑ compared to MCAN) for the challenging 'Open' question category. These open-ended questions require diverse and broad reasoning ability to answer correctly. This is a significant finding as it sheds light upon effectiveness of using complementary semantic relationship features as an important line of research to break the bottleneck of VQA models mostly focusing on learning better visual representations.

[3]experimented with `mcan-large` model from https://github.com/MILVLG/openvqa

Input	Attention	VQA-v2 Test-dev				GQA Test-dev		
		Acc.	Y/N	Num.	Other	Acc.	Binary	Open
$r + q$	Mutual	44.00	66.48	31.49	27.21	35.90	54.72	19.93
	Self	53.35	74.16	35.87	39.46	42.72	64.42	29.38
	MSA	53.66	74.71	36.65	39.70	45.53	64.00	29.86
$v + q$	Mutual	45.74	57.18	34.78	29.74	37.20	56.82	20.56
	Self	70.14	86.57	51.59	60.28	57.03	76.02	40.76
	MSA	70.38	86.78	52.05	60.59	57.45	77.08	40.79
$v + r + q$	Mutual	48.29	67.17	33.24	35.49	39.24	55.45	25.48
	Self	70.46	87.14	51.26	60.57	57.72	76.12	40.48
	MSA	70.76	87.10	53.21	60.77	58.37	77.70	40.44

Table 6.2: *Ablation of MSA model on VQAv2 Test-dev and GQA Test-dev set.* $\boldsymbol{v}$, $\boldsymbol{q}$ and $\boldsymbol{r}$ represent visual features, question features and semantic relationship features respectively.

6.3.5 Ablation study

We perform extensive ablation on the VQAv2 test-dev and GQA test-dev datasets and report the results in Tab. 6.2. Our goal here is to identify which input and attention combination contributes to the overall performance of our model. This is a comprehensive setup as the VQA dataset and GQA dataset consist of natural crowed-sourced and auto-generated questions respectively. We use semantic relation features for all our experiments (i.e., $\boldsymbol{r} = \boldsymbol{r}^{sem}$).

Semantic relationship features provide accuracy boost when used in complement with visual features. The *blind* model which only uses parsed relationship features without any visual features (rows 1–3) performs worse compared to other models that explicitly use visual features. However, when used in complement with image and question features (rows 7–9), it helps models achieve better performance on both VQA and GQA datasets.

Guided self attention provides rich attention distribution over its inputs compared to mutual attention. For the three input combinations listed in Tab. 6.2, we ablate the MSA module by only activating mutual or self attention module. We can see that when only guided self attention module is activated, a better VQA accuracy is achieved in both the datasets. This is because the self attention module captures rich semantics over the the input features through its multi-head attention architecture. The mutual attention component works best in a VQA setting when the attention distribution is learned on the visual feature, which undergoes a second multimodal fusion with the question feature [Fukui et al., 2016; Farazi and Khan, 2018; Ben-Younes et al., 2017, 2019]. By design, we want the mutual attention module to capture the multimodal interaction between the inputs and feed it to the attention fusion module (Sec. 6.2.4) for combining with other attention distributions. Thus a standalone setup for our mutual attention module performs sub-par to the guided self attention module.

Methods	VQAv2 Test-Standard			
	Acc.	Y/N	Num.	Other
Ours	**71.1**	87.3	53.3	61.1
MCAN [Yu et al., 2019]†	70.9	-	-	-
ReGAT [Li et al., 2019b]†	70.6	-	-	-
Ban+Counter [Kim et al., 2018]†	70.4	-	-	-
MuRel [Cadene et al., 2019a]	68.4	-	-	-
Counter [Zhang et al., 2018]	68.4	83.6	51.4	59.1
Graph Learner [Norcliffe-Brown et al., 2018]	66.2	82.9	47.1	56.2
Bottom-Up [Anderson et al., 2018]	65.7	82.2	43.9	56.3

Table 6.3: *Comparison of our single MSA model trained only on VQAv2 train+val dataset with other comparable state-of-the-art models on VQAv2 Test-Std dataset.* Our approach performs favorably well against the existing VQA models. † models undergo additional training on Visual Genome [Krishna et al., 2016] dataset which provides an additional gain.

MSA module with both mutual and self attention performs best. The full MSA model with both mutual and self attention modules achieves better performance compared to when a single block is activated (rows 3,6,7 in Tab. 6.2). The mutual attention module provides complementary information that helps in cases where self attention alone is not sufficient.

6.3.6 Comparison with state-of-the-art models

We report the performance of our single MSA model on benchmark VQAv2 Test-Standard dataset in Tab. 6.3. We show that by leveraging the semantic relationship features, our model is able to outperform other comparable state-of-the-art models, even without additional training on Visual Genome dataset [Krishna et al., 2016]. Some recent models resort to ensembling and data augmentation techniques [Yu et al., 2019], cross-modal pretraining on language and vision tasks [Tan and Bansal, 2019; Li et al., 2019a] to achieve superior performance. However, this is tangential to the motivation of our paper, so they are not directly comparable to our approach. [Zhang et al., 2019a; Hu et al., 2019] do not report their performance on the VQAv2 dataset and are thus not included here for comparison.

6.3.7 Qualitative results

We provide some qualitative results in Fig. 6.3 of MSA model on VQAv2 dataset. We visualize the attention distribution over the region proposals and list two relationship triplet with highest attention for better visualization. For simplicity we do not visualize the mutual and self attention distribution over the question words. We can see that the self and mutual attention component provide complementary attention

distribution over the input features. For example, in the second row, when asked *'Are they wearing goggles?'* The visual self attention component focuses more on the `sunglass` of the person on the left. The mutual attention component looks at the person on the left and the right. Similarly, the self attention component gives more attention to a relationship triplet with `sunglass` and `person`, but further looks at `person wear shirt` relationship triplet for getting more semantic context. Such complementary relationship between various attention components helps VQA model to reason better over its input feature representations.

6.4 Conclusion

VQA problem demands an in-depth understanding of the visual and semantic domains. Existing approaches generally focus on deriving more discriminative visual features or modeling the complex multi-modal interactions. In this chapter, we show that an important missing piece in the existing models is that of enriched semantic relationship modeling. We demonstrate that under an oracle setting, these semantic relationships can bring the performance on par with human-level accuracy on VQA task. Further, we propose an automatic semantic relationship parser alongside a complementary attention mechanism that delivers consistent improvements on SOTA across two challenging VQA datasets. Our results strongly advocate for further investigation on better relationship modeling in the semantic domain, a direction less explored so far in the VQA community.

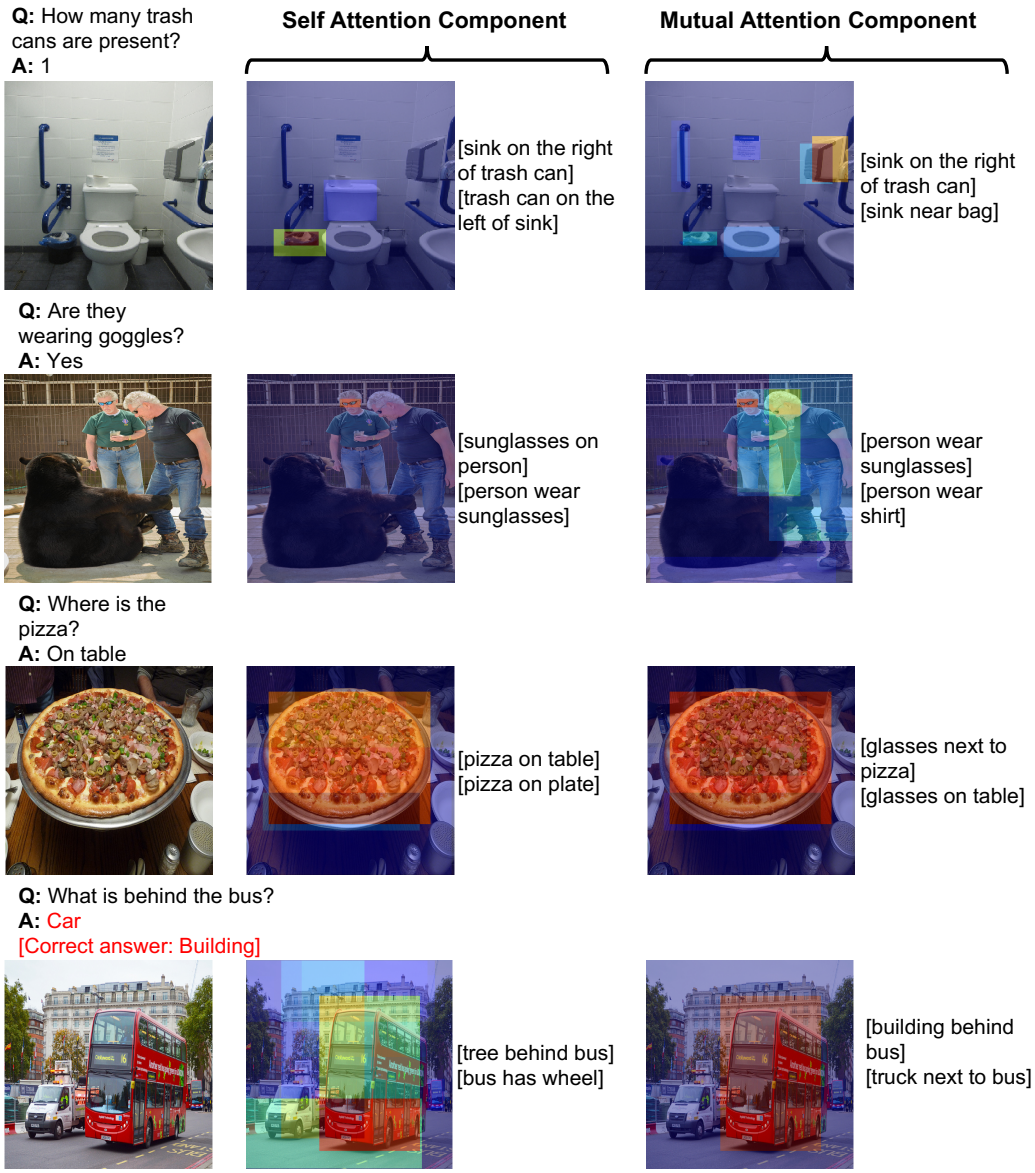


Figure 6.3: *Qualitative results on VQAv2 dataset with semantic relationship parser* Predicted answers and attention distribution over visual features (v) and semantic relationship features (r) employing our MSA model on VQAv2 dataset. We activate both Self and Mutual Attention module and separately visualize their output attention distribution over v and r. The attention distribution over the region proposals are visualized with a heat map and two relationship triplet with highest probability is reported. For both self and mutual attention component provide complementary information for predicting the correct answer.

Chapter 7

Accuracy vs. Complexity Trade-Offs

As discussed in the earlier chapters, the pivot to existing VQA models is the joint embedding that is learned by combining the visual features from an image and the semantic features from a given question. Consequently, a large body of literature has focused on developing complex joint embedding strategies coupled with visual attention mechanisms to effectively capture the interplay between these two modalities. However, modelling the visual and semantic features in a high dimensional (joint embedding) space is computationally expensive, and more complex models often result in trivial improvements in the VQA accuracy. In this chapter, we systematically study the trade-off between the model complexity and the performance on the VQA task. VQA models have a diverse architecture comprising of pre-processing, feature extraction, multimodal fusion, attention and final classification stages. We specifically focus on the effect of *multi-modal fusion* in VQA models that is typically the most computationally expensive step in a VQA pipeline. Our thorough experimental evaluation leads us to two proposals, one optimized for minimal complexity and the other one optimized for state-of-the-art VQA performance. We hope our findings help the community build better VQA models as per design requirements.

7.1 Introduction

The Visual Question Answering (VQA) problem aims to a develop a deep understanding of both vision and language, and the complex interplay between the two, such that a machine is able to answer intelligent questions about a visual scene. The VQA task is inspired by the astounding ability of humans to perceive and process information from multiple modalities and draw connections between them. An AI agent equipped with VQA ability has wide applications in service robots, personal digital assistants, aids for visually impaired and interactive educational tools, to name a few [Antol et al., 2015; Gu et al., 2018].

Given the success of deep learning, one common approach to address the VQA problem is by extracting visual features from an input image or a video using pre-trained Convolutional Neural Networks (CNNs) eg., VGGNet [Simonyan and Zis-

serman, 2014], ResNet [He et al., 2016], ResNeXt [Xie et al., 2017]; and representing language features from the input questions using Recurrent Neural Networks (RNN) eg., [Antol et al., 2015; Fukui et al., 2016; Kiros et al., 2015]. The automatic and generalized feature learning capability of deep neural networks has paved the way towards joint processing of multiple modalities in a single framework, leading to dramatic improvements on the challenging VQA task [Antol et al., 2015; Krishna et al., 2016; Zhu et al., 2016].

To effectively capture the interaction between visual and semantic domains, one must learn a joint representation common between the two domains. Capturing the multimodal interaction between these two modalities is computationally expensive (both in terms of compute and memory footprint), especially when the interactions are learned on high-dimensional visual and language features extracted using deep neural networks. Different multimodal operations ranging from simple linear summation and concatenation to complex bilinear pooling and tensor decomposition have been proposed to effectively model this bi-modal interaction and achieve state-of-the-art VQA accuracy [Fukui et al., 2016; Ben-Younes et al., 2017, 2019].

In this chapter, we specifically focus on studying the trade-off between the complexity and performance offered by different multi-modal fusion mechanisms in VQA models. The multi-modal fusion component is often the most computationally expensive part in a VQA pipeline. It is therefore of interest to analyze its impact on the final performance. Notably, VQA pipelines are often coupled with multi-level, multi-directional attention mechanisms [Lu et al., 2016b; Yang et al., 2016; Jabri et al., 2016; Xu et al., 2015; Kim et al., 2018] that allow the VQA model to identify most salient regions/phrases in the given image/question required to predict a correct answer. Here, we do not analyse different attention mechanisms since they are model-specific and therefore less generalizable across models and different tasks requiring multi-modal integration. However, using a simple attention approach, we demonstrate that attention is helpful in VQA settings across different fusion strategies.

The main contribution of this chapter are as follows:

- We provide a succinct survey of the state-of-the-art VQA models employing multimodal fusion to learn a joint embedding, and describe how most of the leading models leverage a similar high-level architecture.
- We establish a VQA baseline that supports the three most popular meta-architectures (visual features extractor, bilinear fusion and co-attention) and a unified evaluation protocol by varying these meta-architectures (Fig. 7.1).
- We perform an extensive evaluation on three challenging VQA datasets (ie., VQAv2, VQA-CPv2 and TDIUC) for different combinations of feature extractor, bilinear fusion model and attention mechanism to generate an accuracy vs. complexity trade-off curves.
- Our finding suggests VQA models using visual features obtained by Squeeze-and-excitation Network (SeNet [Hu et al., 2018]) mostly outperform models

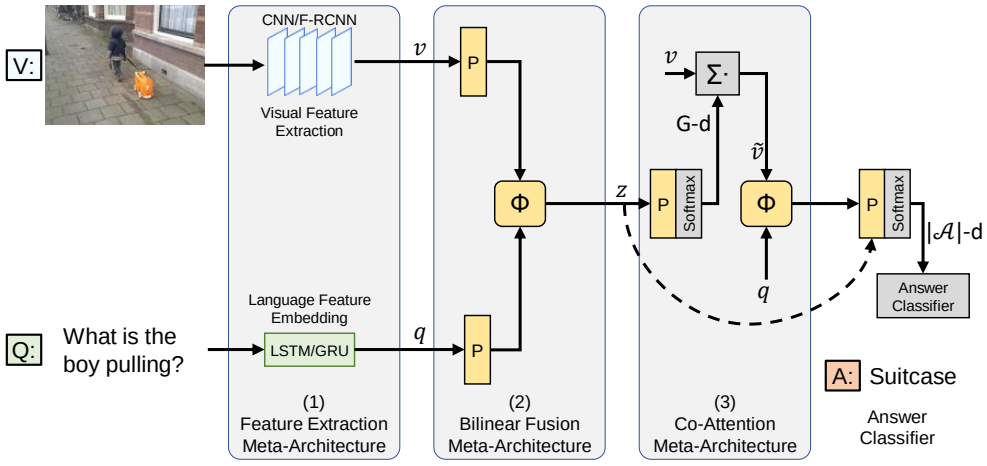


Figure 7.1: *An unified VQA model with three components that co-occur in existing models (we term them meta-architectures).* (1) The feature extraction meta-architecture generates visual feature $\mathbf{v}$ and semantic feature $\mathbf{q}$ from the input image and question respectively. (2) The extracted features are projected to a common-space through P and jointly embedded into z with a bilinear fusion model Φ. (3) The attention meta-architecture takes the joint embedding feature z and learn a spatial attention distribution to generate an attended visual feature representation $\tilde{v}$. The question embedding q and $\tilde{v}$ are again jointly embedded and passed to the answer classifier. The joint embedding feature z can be directly passed to the answer classifier to predict the answer a^* skipping the co-attention meta-architecture (denoted by the dashed line). The trainable blocks are color coded yellow.

using widely adopted ResNet [He et al., 2016] features, irrespective of attention and fusion mechanism. Further, we report that employing MFH fusion facilitates achieving a superior performance over its counterparts.

- We propose a combination of feature extractor and meta-architecture that achieves state-of-the-art performance on three most challenging VQA datasets.

7.2 VQA Model Architecture

The VQA task is modeled as a classification task. Since there exists a long tail distribution of answers in the large-scale VQA datasets, the most frequent answers are placed in a candidate answer set $\mathcal{A}$. The goal is to predict the best possible answer a^* for a natural language question $\mathbf{Q}$ about an image $\mathbf{I}$. This can be formulated as:

$$a^* = \underset{a \in \mathcal{A}}{\arg\max}\, p(a|\mathbf{I}, \mathbf{Q}; \boldsymbol{\theta}) \tag{7.1}$$

where $\boldsymbol{\theta}$ denotes the model parameters.

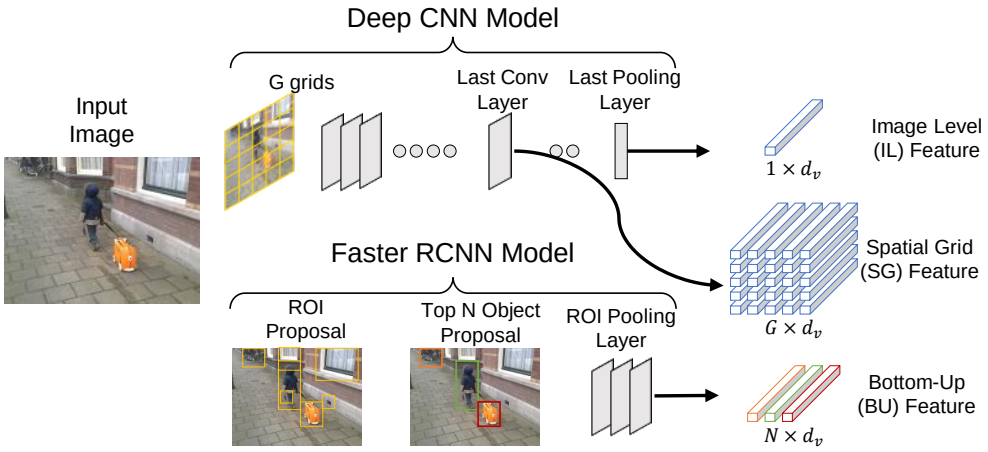


Figure 7.2: *Visual feature extraction meta-architecture illustrating the pipeline for generating Image Level(IL), Spatial Grid (SG) and Bottom-Up(BU) from the input image.*

7.2.1 Feature Extraction Meta-Architecture

The feature extraction component consists of two parts. *First,* a visual feature extraction block takes the input image and extracts visual features. *Second,* a language embedding block generates a semantic embedding from the input features. As these two parts require a trained image and language model on large-scale datasets, these blocks are part of the data pre-processing done before training the VQA model itself.

Visual Feature. To generate discriminative features from images, similar to other high level visual reasoning tasks (e.g., image captioning, visual dialog and relationship prediction), VQA models employ deep neural networks pretrained for object recognition and detection. These deep CNN models generate a feature representation of the image $\mathbf{I}$, denoted as v. It can be formulated as:

$$v = \text{CNN}(\mathbf{I}) = \{v_i,\ s.t.,\ i \in [1, G]\}, \tag{7.2}$$

where $v_i \in \mathbb{R}^{d_v}$ is the feature vector of i^{th} image location and G is the total number of image locations in a grid. The dimension of d_v and G depend on how the features are extracted using a particular CNN model. The extracted visual features can be categorized into three main types (see Fig. 7.2):

i) **Image Level (IL):** These features are extracted from the last pooling layers before the classification layer (eg., 'pool5' layer of ResNet[He et al., 2016]). IL features are $1 \times d_v$ dimensional as they represent the features of the whole image (ie., $G = 1$). When only these features are available, the additional visual attention (discussed in Sec. 7.2.3) is not used as these features have no spatial information.

ii) **Spatial Grid (SG):** The Spatial Grid (SG) features are extracted from the last convolutional layer (eg., 'res5c' if using ResNet-152). The spatial grid feature is $G \times d_v$ dimensional where each feature map corresponds to a uniform grid

location on the input image. While using SG features, an additional attention mechanism is often used to generate a more refined visual representation based on the input question (Sec. 7.2.3).

iii) **Bottom-Up (BU):** Anderson et al. [2018] proposed to use features maps of different object proposals instead of IL or SG features. The object proposals are obtained by passing each input image through an object detector (eg., Faster–RCNN [Ren et al., 2015b]) that is pretrained on large-scale object detection datasets. The extracted features are $G \times d_v$ dimensional, where $G = N$ is the number of top object proposals in an image.

Language Feature. The question is first tokenized into words and encoded in to word embeddings using a pretrained sentence encoder (eg., GLoVe [Pennington et al., 2014], Skip-thought [Kiros et al., 2015]). The length of the word embedding is set to l, determined from the question length distribution in the dataset, where unusually longer question are clipped and short question are zero padded to get a fixed-sized word embedding $\boldsymbol{w}_l$. The word embeddings are passed through LSTMs [Hochreiter and Schmidhuber, 1997] (or its variants) to obtain the semantic features $\boldsymbol{q}$ from the input question:

$$\boldsymbol{q} = \text{LSTM}(\boldsymbol{w}_l), \tag{7.3}$$

where $\boldsymbol{q}$ is the output feature of the last word from the LSTM network and is of d_q dimension. The dimension of the semantic feature embedding is determined by size of the hidden state of the LSTM unit. In the scope of this work, we use a fixed language feature extraction meta-architecture for all our experiments since our goal is to study the trade-off provided by multi-modal fusion strategies. However, a more advanced word embedding, such as Bidirectional Encoder Representations from Transformers (BERT) [Devlin et al., 2018] can provide additional performance gains.

7.2.2 Fusion Model Meta-Architecture

The second meta-architecture (common to VQA models) jointly embeds the extracted visual and semantic features into a common space. To this end, a multimodal embedding function Φ is learned:

$$\boldsymbol{z} = \Phi(\boldsymbol{q}, \boldsymbol{v}) \tag{7.4}$$

where $\boldsymbol{z}$ is the learned joint embedding from the input question and image. The simplest way to project $\boldsymbol{q}$ and $\boldsymbol{v}$ into the same space is by taking Hadamard product of the inputs, $\boldsymbol{z} = \boldsymbol{v} \odot \boldsymbol{q}$. However, this operation requires the inputs to be of equal dimension and is limited to a linear model.

To fuse visual and semantic features of equal or different dimension and capture the complex interaction between these two modalities, one can adopt bilinear fusion models and take the outer product of the two input feature vectors:

$$\boldsymbol{z} = \mathcal{W}[\boldsymbol{v} \otimes \boldsymbol{q}] \tag{7.5}$$

where $\mathcal{W} \in \mathbb{R}^{d_v \times d_q \times \mathcal{A}_d}$ is the learned fusion model, $\mathcal{A}_d$ is the number of entries in the candidate answers set, while $\otimes$ and $[\,]$ denote outer product and vectorization operations, respectively. This operation allows the model $\mathcal{W}$ to learn the interactions between the inputs in a multiplicative manner. One major limitation of this approach is $\mathcal{W}$ is very high dimensional. For example, if one is using SG features with a ResNet-152 backbone, LSTM with 2048 hidden dimensions and an answer classifier with 3000 candidate answers, the leaned model $\mathcal{W}_i$ for i^{th} image grid location will be $\mathbb{R}^{2048 \times 2048 \times 3000}$. As a result, even with a simplistic design, the VQA model will have over 12 billion learnable parameters which is expensive both computationally and memory-wise. Several models have been proposed to tackle this problem and in this chapter we aim to investigate the trade-off between complexity and accuracy of VQA models by using a variety of bilinear fusion methodologies.

We first establish two simple baseline multi-modal models and then formulate different bilinear models proposed in the literature in our experimental setting. The baseline models we experiment with in the scope of this work are as follows.

Linear: Linear Summation is the simplest multi-modal fusion model that we experiment with. It first transforms the input feature vector into an intermediate space through fully connected layers. The intermediate features are then added together and projected back to the answer prediction space through another fully connected layer. This operation only uses a linear operation (ie., summation) in the intermediate space to capture the interaction between the visual and language features, thus is dubbed Linear.

C-MLP: The second baseline fusion model is called Concatenation-MLP (C-MLP). We first concatenate the input features in their native space and pass the resulting visual-semantic features through a 3 layer Multi-layer Perceptron (MLP). The MLP model learns to non-linearly encode the concatenated features and produces a joint-embedding feature in the answer prediction space.

MCB: Multi-modal Compact Bilinear (MCB) pooling [Fukui et al., 2016] introduced the use of bilinear models to perform fusion between visual and semantic feature vectors in a VQA setting. First, the input feature vectors are approximated as $\boldsymbol{q}'$ and $\boldsymbol{v}'$ by using count-sketch projection [Charikar et al., 2002] and then their element wise product is taken in the spectral domain. The spectral domain transformation is achieved via a Fast Fourier Transform (FFT).

$$\boldsymbol{z} = \text{FFT}^{-1}(\text{FFT}(\boldsymbol{v}') \odot \text{FFT}(\boldsymbol{q}')). \tag{7.6}$$

This operation leverages the property that convolution in the time domain is equivalent to element-wise product in the frequency domain; and the frequency domain product is converted back to the original domain by an inverse Fourier transformation. However, the model is still quite expensive to train as it requires the resultant joint embedding vector z to be high-dimensional (precisely $16,000d$) to have a superior VQA accuracy.

MLB: To reduce the dimensions of the output feature vector, Kim etal. [Kim et al., 2016] proposed Multimodal Low-rank Bilinear Pooling (MLB). MLB uses a

low-rank factorization of input features vectors during bilinear operation. The input feature vectors, $\boldsymbol{v}$ and $\boldsymbol{q}$ are projected to a joint embedding space $z \in \mathbb{R}^{d_z}$ and their Hadamard product is taken as follows:

$$\boldsymbol{z} = (P_v^T \boldsymbol{v}) \otimes (P_q^T \boldsymbol{q}) \tag{7.7}$$

where P_v and P_q are projection matrices of dimension $\mathbb{R}^{d_v \times d_z}$ and $\mathbb{R}^{d_q \times d_z}$ respectively. Here, the output joint embedding size d_z is set to $1,000$. Generally, the VQA model thus developed achieves better accuracy than the former MCB approach.

MFB: Even though MLB achieves comparable performance with MCB, it takes longer to converge. Multi-modal Factorized Bilinear Pooling (MFB) [Yu et al., 2018] proposed to add a pooling operation on the jointly embedded feature vector. This process is divided in two stages. First, during the *Expand* stage, the projection dimension is expanded by a factor k and the input visual feature vectors are projected onto $k \times d_z$ dimension. Second, during the *Squeeze* stage, a sum-pooling operation is performed with size k of non-overlapping windows, which squeezes the joint feature embedding by the same factor k.

$$\boldsymbol{z} = \text{Sum-Pool}\big((\tilde{P}_v^T \boldsymbol{v}) \otimes (\tilde{P}_q^T \boldsymbol{q}),\ k\big) \tag{7.8}$$

where the new projection matrices are denoted by $\tilde{P}_v \in \mathbb{R}^{d_v \times d_z \times k}$ and $\tilde{P}_q \in \mathbb{R}^{d_q \times d_z \times k}$. After sum-pooling over k windows, the joint embedding feature vectors again become d_z dimensional. It can be see that setting $k = 1$, MLB can be considered as a spacial case of MFB. The inclusion of the sum-pooling operation with a factor k improves the convergence of the VQA model and provides boost in VQA accuracy compared to MLB.

MFH: To model a more complex interactions, Multi-modal Factorized High-order pooling (MFH) [Yu et al., 2018] uses a series of cascading MFB blocks. Each MFB block takes the input feature vectors and internal feature of the previous MFB block. The internal feature of i^{th} MFB block among a total of m cascaded MFB blocks can be formulated as:

$$\boldsymbol{z}_{\text{int}}^i = \begin{cases} \mathbb{1} \otimes \big((\tilde{P}_v^T \boldsymbol{v}) \otimes (\tilde{P}_q^T \boldsymbol{q})\big), & \text{when } i = 1 \\ z_{\text{int}}^{i-1} \otimes \big((\tilde{P}_v^T \boldsymbol{v}) \otimes (\tilde{P}_q^T \boldsymbol{q})\big), & \text{when } i > 1 \end{cases}, \tag{7.9}$$

where $i \in [1, m]$ and $\mathbb{1}$ is a $d_z \times k$ dimensional matrix of all ones. $\boldsymbol{z}_{\text{int}}^i \in \mathbb{R}^{d_z k}$ is similar to the output of the *Expand* stage of MFB except for the additional input from the previous MFB block. The output joint embedding of the i^{th} MFB block is formulated as:

$$\boldsymbol{z}^i = \text{Sum-Pool}(\boldsymbol{z}_{\text{int}}^i) \tag{7.10}$$

Finally, the final output of a MFH operation with m MFBs block is obtained by concatenating the output of each MFB block:

$$\boldsymbol{z} = [\boldsymbol{z}^1, \boldsymbol{z}^2, ...\boldsymbol{z}^m]. \tag{7.11}$$

Here, the output joint embedding vector $\boldsymbol{z} \in \mathbb{R}^{d_z m}$. When $m = 1$, then MFB can be considered as a spacial case of MFH.

Mutan: Multimodal Tucker Fusion (Mutan) [Ben-Younes et al., 2017] first proposed tensor decomposition techniques to reduce the dimensionality of input visual and semantic feature vectors, and the output joint feature embedding in a VQA model. We can re-write Eq. 7.5 to obtain joint embedding vector $\boldsymbol{z}$ with tensor notation as:

$$\boldsymbol{z} = (\mathcal{W} \times_1 \boldsymbol{v}) \times_2 \boldsymbol{q}, \tag{7.12}$$

where the operator $\times_i$ defines i^{th} mode product between the learned tensor $\mathcal{W}$ and input feature vectors. Following Tucker decomposition [Tucker, 1966], the 3-way learned model tensor $\mathcal{W}$ can be decomposed into a core tensor and three factor matrices:

$$\mathcal{W} := \mathcal{T}_c \times_1 F_v \times_2 F_q \times_3 F_z \tag{7.13}$$

with the core tensor $\mathcal{T}_c \in \mathbb{R}^{d_{pv} \times d_{pq} \times d_z}$, and visual, question and joint embedding factor matrices are respectively $F_v \in \mathbb{R}^{d_v \times d_{pv}}$, $F_q \in \mathbb{R}^{d_q \times d_{pq}}$ and $F_z \in \mathbb{R}^{|\mathcal{A}| \times d_z}$. The factor matrices F_v, F_q project the input feature vectors to d_{pv} and d_{pq} dimensional space, respectively, and the core tensor $\mathcal{T}$ models the interaction between the projected feature vectors and the output joint embedding. Now, to encode the fully bilinear interaction in the joint embedding space z, we can formulate Eq. 7.12 as:

$$z = (\mathcal{T}_c \times_1 F_v^T \boldsymbol{v}) \times_2 F_q^T \boldsymbol{q}. \tag{7.14}$$

Here, the dimensions d_{pv} and d_{pq} directly contribute to the model complexity and are usually set to ~ 300 with d_z set to ~ 500. Comparing MLB (Eq. 7.7) and Mutan (Eq. 7.14), MLB can be considered as a spacial case of Mutan if $d_{pv} = d_{pq} = d_z$ and the core tensor $\mathcal{T}_c$ is set to identity. This approach is more efficient compared to MLB as the rank of the core tensor is constrained which balances the interaction between the input feature vectors to achieve a higher accuracy.

Block: In Mutan, the multimodal interaction is solely modelled by the core tensor $\mathcal{T}_c$ which captures the rich interaction between the input features but is limited by the dimensions of the output joint embedding space. This causes the VQA accuracy to saturate for a given setting of intermediate projection dimension. To overcome this bottleneck, a block-superdiagonal tensor based decomposition (Block) technique for VQA was proposed by [Ben-Younes et al., 2019]. The 3-way learned model tensor $\mathcal{W}$ is decomposed in n blocks/chunks as follows:

$$\mathcal{W} = \mathcal{T}_B \times_1 F_v \times_2 F_q \times_3 F_z, \tag{7.15}$$

where $\mathcal{T}_B \in \mathbb{R}^{(d_{pv}n) \times (d_{pq}n) \times (d_z n)}$ and $F_v = [F_v^1, F_v^2, ..., F_v^n]$ (with a similar formulation for F_q, F_z). Each of the n core tensor blocks represents bilinear interaction between chunks of input features. Dividing the core tensor and its factor matrices into blocks allows the model to capture the interaction between several chunks of input feature vectors that get mapped into the joint embedding space. The joint embedding feature

output of i^{th} block is:

$$z^i = (\mathcal{T}_b^i \times_1 (F_v^i)^T \boldsymbol{v}^i) \times_2 (F_q^i)^T \boldsymbol{q}^i, \tag{7.16}$$

where $i \in [1, n]$ and the dimensions of i^{th} core tensor and other factor matrices are reduced by a factor of n compared to the same variables in Eq. 7.15. The final output joint embedding feature vector $\boldsymbol{z}$ is computed as the concatenation of n block term joint embedding features as:

$$\boldsymbol{z} = [\boldsymbol{z}^1, \boldsymbol{z}^2, ..., \boldsymbol{z}^n] \tag{7.17}$$

where $\boldsymbol{z} \in \mathbb{R}^{d_z}$. If we set, $n = 1$ in Eq. 7.17, meaning only one core tensor is used to model the interaction between the input features, block-superdiagonal tensor based decomposition becomes the Tucker decomposition as in Eq.7.13.

7.2.3 Attention-based Meta-Architecture

Different questions about the same image would require a VQA model to attend to different spatial regions within an image. An additional attention mechanism allows the VQA models to identify relevant image regions for answering the question by learning an attention distribution. As mentioned in Sec. 7.2.1, each location of the SG and BU features represent a spatial grid location or an object proposal, respectively. This visual attention can be applied to a Spatial Grid (SG) and/or Bottom-Up (BU) image features where $G > 1$, where an attention mechanism learns to identify which grid locations or object proposals are most relevant in answering the given question. This question specific attention generally allows the model to achieve superior performance.

In the attention meta-architecture, we experiment with *co-attention*. The co-attention process consists of two steps each of which requires the model to learn a joint-embedding feature vector from visual and semantic features (see component (3) in Fig. 7.2). In the **first** step, the model learns to generate an attention distribution vector using the input visual and language features. Irrespective of the bilinear embedding module used, the model learns an attention probability distribution $\alpha \in \mathbb{R}^G$ for input visual features with $G > 1$ spatial/object locations:

$$\alpha = \text{Softmax}\big(P_\alpha \sigma(\mathcal{W}[\boldsymbol{v} \otimes \boldsymbol{q} \cdot \mathbb{1}])\big), \tag{7.18}$$

where $\mathbb{1}$ denotes the repeat (tile) operation to make the question feature $d_q \times G$ dimensional, $P_\alpha \in \mathbb{R}^{d_z \times G}$ projects the joint embedding features to G dimensions and σ is a non-linear activation function (usually *tanh* or sigmoid). It has been found ([Fukui et al., 2016; Yu et al., 2017]) that learning multiple attention distributions, commonly termed as glimpse, increases the VQA accuracy. At each glimpse t, the models learns an attention distribution α^t that results in a better probability distribution.

In the **second** stage, the attention distribution α^t is used to take a weighted sum of the input visual features in G spatial locations. The attended visual feature for

glimpse t can be formulated as:

$$\tilde{v}^t = \sum_{g=1}^{G} \alpha^t \boldsymbol{v}, \tag{7.19}$$

where $\tilde{\boldsymbol{v}}^t \in \mathbb{R}^{d_v}$. If $t > 1$, attended visual features over multiple glimpses are concatenated as $\tilde{\boldsymbol{v}} = [\boldsymbol{v}^1, \boldsymbol{v}^2, ...\boldsymbol{v}^{t_f}]$, where t_f is the last glimpse. The final attended visual feature representation undergo a second bilinear embedding with the question feature:

$$p(a|\tilde{\boldsymbol{v}}, \boldsymbol{q}; \boldsymbol{\theta}) = \text{Softmax}\big(P_{|\mathcal{A}|}\sigma(\mathcal{W}[\boldsymbol{v} \otimes \boldsymbol{q}])\big), \tag{7.20}$$

where $P_{|\mathcal{A}|} \in \mathbb{R}^{d_z \times d_{|\mathcal{A}|}}$ is a projection matrix to the candidate answer space, p is the posterior probability distribution in that space and $\boldsymbol{\theta}$ denotes the same parameter set as described in Eq. 7.1.

7.3 An Unified VQA Model

As discussed in the previous section, the VQA model is made of three main components. Different state-of-the-art models use different combinations of these meta-architectures to achieve superior performance. To experiment with different extracted features and bilinear models, we first establish a modular Unified VQA (U-VQA) architecture that supports different variations of the meta-architectures.

Visual Feature Extractor: We extract IG and SG visual features using the following pre-trained deep CNN models[1] for object detection:

- Inception Net [Szegedy et al., 2015]: Several versions of Inception net have been proposed over the years. In our experiments, we used the InceptionNet-V4 with $d_v = 1536$ and $G = 12 \times 12$. This means the IG features are 1536 dimensional and SG features are $1536 \times 12 \times 12$. Compared to other visual features extracted with pretrained object detectors, Inception features are the lowest dimensional visual features that we experiment with.

- ResNet [He et al., 2016]: Visual features extracted by ResNet are widely used in a VQA setting. In our experiment, we use the Facebook implemented version of ResNet-152[2] model, which has a slightly better performance compared to the original ResNet implementation. IG and SG features extracted with ResNet152 are 2048 and $2048 \times 14 \times 14$ dimensional, respectively.

- ResNext [Xie et al., 2017]: ResNext reported better performances than the counterpart ResNet architecture on the ImageNet and COCO detection datasets. We used ResNext101 in our experiments and the extracted IG and SG features have same dimension as ResNet features.

[1] https://github.com/Cadene/pretrained-models.pytorch
[2] https://github.com/facebookarchive/fb.resnet.torch

- SeNet [Hu et al., 2018]: In complement to spatial features, SeNet adaptively recalibrates the channel-wise features to achieve a higher accuracy om ImageNet dataset. We use the SeNet154 model in our experiments which also has the same IG and SG feature dimensions as ResNet152 and ResNext101.

- PolyNet [Zhang et al., 2017b]: We use PolyNet to extract IG and SG features which are 2048 and $2048 \times 12 \times 12$ dimensional, respectively. The d_v dimension of PolyNet features are equal to ResNet, ResNext and SeNet, but it has fewer spatial grid locations compared to the former models.

We use the BU features[Anderson et al., 2018] made available by their official online repository[3]. The BU features are extracted by using a Faster-RCNN model with a ResNet-101 backbone for the top N object proposals for each image. N can be adaptive (top 10 to 100 proposals) or fixed (top 36 proposals). For our experiments we use the BU features with $N = 36$.

Language Feature Embedding: Similar to Ben-Younes et al. [2019], we use a pretrained Skip-thought [Kiros et al., 2015] vectors and GRUs to encode the language features. The language feature embedding is set to $d_q = 2400$ for all our experiments.

Multi-modal Fusion Models: To embed the multi-modal features into a joint embedding space, we experiment with fusion models discussed in Sec. 7.2.2 and two additional baseline fusion models, namely 'Linear' and 'C-MLP'. Following are the hyper parameters settings that we use for experimenting with these fusion models:

- Linear: The intermediate dimension where the visual and semantic features are projected is set to 1000. The 1000 dimensional features are summed and projected to the candidate answer space.

- C-MLP: The visual and question features are concatenated and passed through a MLP layer with 1600 hidden dimensions. The output dimension of the MLP is set to the dimension of the candidate answer space.

- MCB: We set the joint embedding size to $16,000$ following the original implementation details reported in [Fukui et al., 2016].

- MLB: The joint embedding size d_z is set to 1200 following the original implementation details reported in [Kim et al., 2016].

- MFB: Following the notation in Sec. 7.2.2, k is set to 5 and d_z is set to 1000 following the original implementation.

- MFH: For MFH, we keep the values of k and d_z same as the values used in the MFB implementation, and cascade size is set $m = 2$ MFB blocks.

- Mutan: Following the notation described in Sec. 7.2.2, we restrict the rank $\mathcal{T}_c$ to 10 and d_z is set to 700.

[3]https://github.com/peteanderson80/bottom-up-attention

- Block: The rank of block core tensor $\mathcal{T}_B$ is set to 15, d_z is set to 1600 and the number of blocks/chunks n is set to 18 following the original implementation.

In our experiments, we use the official implementation of Mutan and Block, and PyTorch implementation of MCB from Block[4]. We re-implement MLB, MFB and MFH bilinear models in our unified VQA architecture in PyTorch[Paszke et al., 2019].

Co-Attention Mechanism: We learn an attention distribution map on the SG or BU features by using a co-attention mechanism. The learned attention probability distribution α indicates which spatial grid locations (for SG features) or object proposals (for BU features) are more important for answering the input question. For all our experiments, we use two glimpse, which means for a given image-question pair, two different α are generated. These attention distribution maps are applied on the input visual features separately and the resulting visual representation is concatenated to a vector of size $2 \times d_v$.

7.4 Datasets

We perform extensive evaluation on three VQA benchmark datasets, namely VQAv2 [Goyal et al., 2017], VQA-CPv2 [Agrawal et al., 2018] and TDIUC [Kafle and Kanan, 2017]. The first dataset we experiment on is **VQAv2** [Goyal et al., 2017]. This dataset is a refined version of the VQAv1 [Antol et al., 2015] dataset as it introduces complementary image-question pairs to mitigate the language bias present in the original dataset. The VQAv2 dataset contains over $204K$ images from the MSCOCO dataset [Lin et al., 2014] and $1.1M$ open-ended questions paired with these images (an average of 6 questions per image). Each question has 10 ground-truth answers sourced from crowd-workers for the open-ended questions. The evaluation on the VQAv2 test-set can only be done by submitting the evaluation file in their online evaluation server, and offline evaluation can be done by training the model on train split and evaluating the models performance on the validation split. For this reason, we report validation scores on Tab. 7.1 and the best test-standard scores in Tab. 7.5.

Further, we experiment on the Visual Question Answering under Changing Priors (VQA-CP) dataset. The **VQA-CPv2** dataset is re-purposed from the training and validation sets of the VQAv2 dataset. Similar Image-Question-Answer(IQA) triplets of the training and validation splits of the VQAv2 dataset are grouped together and then re-distributed into train and test sets in a way that questions within the same question type (eg., 'what color', 'how many' etc.) and similar ground-truth answers are not repeated in test and train splits. This makes it harder for any VQA model to leverage the language bias to artificially achieve a higher accuracy.

Finally, we perform experiments on the Task Directed Image Understanding Challenge (TDIUC) dataset. The **TDIUC** dataset divides the VQA paradigm into 12 different task directed question types. These include questions that require a simpler task (e.g., object presence, color attribute) and more complex tasks (e.g., counting, positional reasoning). The IQA triplets are sourced from train and validation set of

[4]https://github.com/Cadene/block.bootstrap.pytorch

VQA dataset and Visual Genome [Krishna et al., 2016] dataset, but undergo some automatic and manual annotations to generate the ground-truth.

Evaluation Metric: While experimenting on the VQA and VQA-CP dataset, we report VQA accuracy following the standard protocol [Antol et al., 2015; Goyal et al., 2017; Krishna et al., 2016]. The accuracy of a predicted answer a^* is:

$$\text{VQA Accuracy} = \min\left(\frac{\#\text{ of humans answered } a^*}{3}, 1\right) \tag{7.21}$$

which means if the predicted answer a^* is given by at least 3 human annotators out of 10, then it will be considered correct. We report overall accuracy on the dataset for all question types along with 'Yes/No', 'Number' and 'Other' question types.

When evaluating on the TDIUC dataset, we report accuracy for each of the 12 question types defined in the dataset. It allows us to further evaluate the capacity of a fusion model to answer diverse types of questions that require different reasoning capabilities. As the VQA datasets are crowd-sourced, they have an inherent bias due to a skewed question distribution. Along with the individual accuracy, we also report arithmetic and harmonic means across all per-question-type accuracy, dubbed arithmetic mean-per-type (Arithmetic MPT) and harmonic mean-per-type accuracy (Harmonic MPT). Arithmetic MPT reflects the models ability to score equally across all question categories, and Harmonic MPT measures the models' ability to have high scores across for harder (low-scoring) question-types. Further, we also report the normalized arithmetic and harmonic MPT, along with traditional overall VQA accuracy.

Answer Encoding: The VQA task is formulated as a classification problem following the benchmark practices [Antol et al., 2015; Goyal et al., 2017; Krishna et al., 2016] where a candidate answer set is created for the most frequent answers in the dataset. This is because VQA datasets have a very long tailed distribution and the least frequent answers account for a fraction of the IQA pairs in the dataset. For experimenting on VQAv2 and VQA-CPv2 datasets, we select the most frequent 3000 answers and for TDIUC we select 1460 for the candidate answer set $\mathcal{A}$.

7.5 Experiments and Results:

We perform evaluation on the VQAv2, VQA-CPv2 and TDIUC datasets in the scope of this work and group the core experiments into four main categories.

- *The Effect of Visual Features:* We vary the input visual feature meta-architecture to evaluate the effect on VQA complexity while using different level of visual feature.

- *The Effect of Fusion Meta-architecture:* We vary the fusion meta-architecture to evaluate different so-far proposed strategies and to analyze their complexity-accuracy trade-off while using simpler to complex joint embedding models.

- *The Effect of Attention Model:* We further study the effect of additional attention mechanisms on the complexity accuracy trade-off.
- *Proposed Meta-architecture:* Finally, we find the most effective meta-architecture combination and report state-of-the-art performance on VQAv2, VQA-CPv2 and TDIUC using the recommended meta-architecture combination.

	Visual Feature									
Bilinear	InceptionV4		ResNet152		ResNext101		SeNet154		PolyNet	
Model ↓	IL	SG	IL	SG	IL	SG	IL	SG	IL	SG
Linear	35.04	36.97	39.26	39.56	37.88	38.90	37.32	38.18	40.22	38.14
C-MLP	52.34	54.89	53.37	58.50	53.28	57.90	54.06	57.96	52.78	56.68
MCB	52.83	53.44	54.91	58.15	55.04	57.94	55.34	58.23	55.85	57.29
MLB	52.66	52.53	53.79	57.16	53.77	56.31	54.69	56.34	54.91	57.02
Mutan	53.35	53.97	55.60	58.94	55.67	57.21	55.41	58.11	55.97	58.75
MFB	53.88	53.55	55.47	58.31	55.45	57.63	56.16	57.51	57.69	57.93
Block	**55.08**	**55.89**	56.85	60.49	56.87	59.67	**57.36**	**59.67**	**58.12**	**60.54**
MFH	54.86	55.28	**57.07**	**60.53**	**57.06**	**59.89**	57.16	59.64	57.59	60.53

Table 7.1: *Evaluation on VQAv2 [Goyal et al., 2017] validation set with visual features extracted using different CNN models.*

We quantify the complexity of a VQA model based on the number of trainable parameters, FLOPS (floating point operations per second) and computation time (both CPU and GPU). The visual features are extracted as a pre-processing step for VQA models; thus pre-training the models on the ImageNet dataset or similar object detection dataset does not directly contribute to the complexity of a VQA model. However, the BU features require training an additional object detector on another large scale dataset (i.e., Visual Genome [Krishna et al., 2016]). Thus we offset the FLOPS and trainable parameters with the additional training cost for performing experiments with BU features, and plot VQA accuracy versus trainable parameter and FLOPS. As our goal is to determine the optimal meta-architecture configuration for highest VQA accuracy, we draw an imaginary *maximum efficiency* line on the accuracy vs. training parameter and accuracy vs. FLOPS plots, that helps us study the overall trends.

7.5.1 Varying the level of Visual Features

In Tab. 7.1 we report validation scores on the VQAv2 dataset and in Tab. 7.2 we report the test scores on the VQA-CPv2 dataset, using Image Level (IL) and Spatial Grid (SG) features across eight different fusion models. Our main insights are as follows:

Setting visual feature dimension closer to language feature embedding improves VQA performance. It can be seen from Tab. 7.1 and 7.2 that the VQA models based on InceptionV4 features perform significantly worse (about $\sim 5.0 \downarrow$) than

Bilinear Model ↓	Visual Feature									
	InceptionV4		ResNet152		ResNext101		SeNet154		PolyNet	
	IL	SG	IL	SG	IL	SG	IL	SG	IL	SG
Linear	17.61	17.77	17.58	19.7	18.09	19.93	17.97	29.11	18.6	25.11
C-MPL	27.0	29.23	27.27	31.38	27.32	30.23	28.6	32.31	26.65	29.35
MCB	27.2	28.15	28.43	30.87	27.25	29.11	30.19	31.28	30.25	31.71
MLB	26.1	27.70	24.61	31.52	26.13	30.79	27.87	32.33	27.6	31.96
Mutan	28.25	28.02	**29.27**	31.32	29.64	28.97	30.75	31.74	31.04	32.04
MFB	27.51	28.69	28.44	33.05	28.9	32.38	30.39	33.61	29.90	33.32
Block	**28.45**	29.73	29.17	34.45	29.41	33.18	31.0	35.16	30.71	35.11
MFH	28.27	**30.07**	29.1	**34.6**	**29.7**	**34.3**	**31.63**	**35.9**	**31.06**	**35.48**

Table 7.2: *Evaluation on VQA-CPv2 [Agrawal et al., 2018] test set with visual features extracted using different CNN models.*

similar models while using IL features instead of SG features (for different CNN backbones). The main reason is that for all our experiments we kept the language feature embedding at 2400, which is similar to feature dimensions $d_v = 2048$ of other feature extractors. InceptionV4 features have a significantly smaller dimension $d_v = 1536$ than the language feature embedding. While a bilinear fusion model tries to learn a joint feature embedding, the smaller visual feature dimension affects the model's ability to equally capture the visual-semantic relationships, thereby deteriorating VQA accuracy.

One can use a projection layer to make the visual features high-dimensional (ie., closer to the dimension of language feature embedding), however, it is generally not recommended. Projecting the visual features to a higher dimension through learned layers introduces a higher complexity and results in over-fitting on a specific dataset. Consequently, the pretrained model does not generalize well to held-out test sets. Another way to make to Inception visual features high dimensional is by increasing the input image size. This approach has practical limitations as we are using a pre-trained feature extraction model with a fixed architecture. In practice, one should not make the visual feature high dimensional to match the language feature embedding, rather one should modify the LSTM architecture to make the semantic feature dimension similar to the visual feature. As the language feature embedding is extracted from the last LSTM cell and is related to the hidden dimension of the cell, it is relatively easy to modify the hidden dimensions to obtain an arbitrary sized language feature embedding.

PolyNet features with smaller grid size perform surprisingly well. Expect for InceptionV4 and PolyNet, all the other feature extractors we experiment with have $14 \times 14 = 196$ grid locations. Having more grid locations allows a model to learn to identify salient image locations in a higher resolution. However, more grid locations introduce a higher complexity in the VQA models. Surprisingly, PolyNet features with $12 \times 12 = 144$ grid locations, which translates to a $\sim$ 106k reduction

in visual feature dimension compared to ResNet152 features, achieves the highest VQAv2 validation accuracy while using Block and MFH fusion models (Tab. 7.1) with SG features. Also, on a more challenging VQA-CPv2 dataset, it performs better then ResNet152 features (0.7 ↑ and 0.9 ↑ for Block and MFH models, respectively, with SG features). This means that visual features extracted by PolyNet are highly discriminative and can perform on par with higher resolution visual features in a similar VQA setting.

Resnet152 features have the largest performance boost when using SG instead of IL features. In Tab. 7.1 and 7.2, we perform experiments using both IL and SG features. While using the SG features, we employ the co-attention mechanism (see component (3) in Fig. 7.1). Naturally, using SG features instead of IL features extracted using the same visual feature extraction meta-architecture results in a significant performance boost. However, the highest improvement is achieved when ResNet152 SG features are used instead of ResNet152 IG features. The average VQA accuracy boost across all fusion models (expect for Linear) is 3.72 for ResNet152 compared to 0.65, 2.48, 2.47 and 2.26 for InceptionV4, ResNext101, SeNet154 and PolyNet, respectively, on the VQAv2 dataset (Tab. 7.1). The accuracy gain using SG features is more with ResNet152 when we experimented on the VQA-CPv2 dataset; 4.42 for ResNet152 and 1.25, 2.94, 3.41 and 3.11 for InceptionV4, ResNext101, SeNet154 and PolyNet features, respectively.

SeNet154 features perf0rm better on datasets with less language bias. The VQA-CPv2 dataset allows a more challenging evaluation benchmark for VQA models as its test split has a different schematic data distribution compared to its training counterpart. This prevents a VQA model from cheating by learning the language bias to score higher. In this challenging setting, models using SeNet154 features achieve higher accuracy compared to their performance on the VQAv2 dataset. For example, in Tab. 7.1, the MFH-SG model with ResNet152 features achieves 60.53 whereas with SeNet154 it achieves 59.64. This trend reverses when evaluated on VQA-CPv2 dataset; MFH-SG model with ResNet152 features scores significantly lower (1.3 ↓) than MFH-SG models using SeNet154 features. This trend is also true for models using SeNet154 IL features. One possible reason is that SeNet154 has an additional channel attention module that comes in handy when language bias is smaller and the model has to rely on better visual features.

Using Bottom-Up features provides a consistent accuracy gain. Instead of Spatial Grid features, most state-of-the art VQA models use bottom-up features [Anderson et al., 2018]. In this case, the whole image is represented as a collection of region-based visual features instead of visual features from a fixed grid for every image. As humans naturally tend to ask questions about the objects present in an image, localizing different regions containing objects and their parts allows a VQA model to jump-start the visual attention process and identify which object regions are more important to answer the question. Meanwhile, for a model that takes a uniform grid representation and is required to identify arbitrary image regions relevant to the question. We use the bottom-up features provided by [Anderson et al., 2018] for our experiments which uses a ResNet-101 backbone for feature extraction. We compare

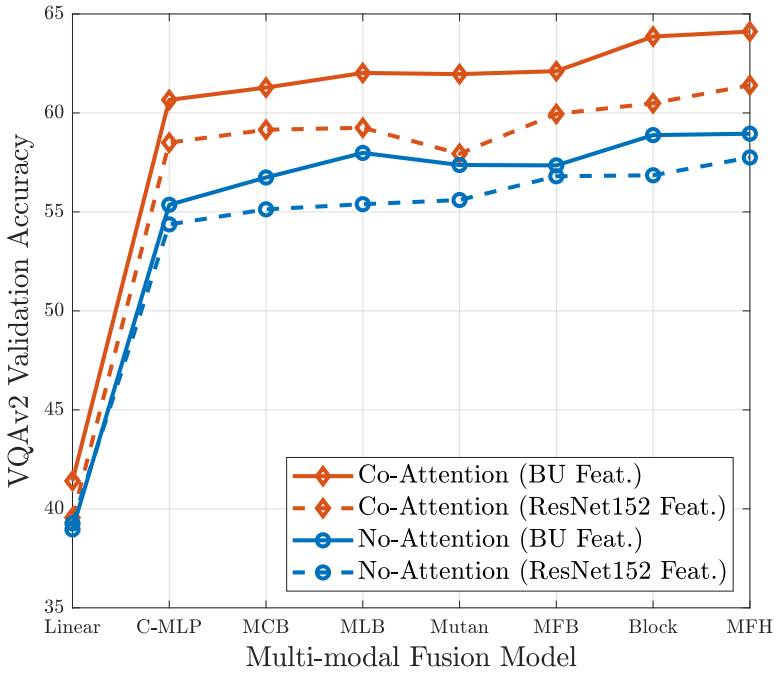


Figure 7.3: *Comparing VQAv2 validation accuracy of Co-Attention and No-Attention version of our Unified VQA model, using ResNet152 Spatial Grid (SG) and Bottom-Up (BU) features.*

similar VQA models using BU features with Resnet152 IL and SG features on VQAv2 (Tab. 7.3) and VQA-CPv2(Tab. 7.4) datasets. We generate Image Level (IL) BU features by average pooling the visual features across the number of objects-proposals to generate $d_v = 2048$ dimensional features. While using no-attention models, we use the image-level features and for co-attention model we use spatial-grid/object-level features. Form Fig. 7.3 and Fig. 7.4 we see that:

– *VQA models using BU features perform consistently better than models using ResNet152 features.* On both VQAv2 (Tab. 7.3) and VQA-CPv2 (Tab. 7.4) datasets, we see that the models using BU features (solid lines) achieve a higher accuracy than the models using ResNet152 features (dashed lines) in all cases. This is because the bottom-features undergo an additional attention step (during top-N object ROI pooling, See Fig. 7.2) compared to the conventional ResNet features. The no-attention models with BU features have less accuracy gain compared the co-attention models using BU features. This is because the original BU features have 2048-d visual feature representation for each distinct object, but when they undergo pooling operations, some spatial information pertaining to a single object and its parts is lost. However, when the co-attention models use the BU features, the models learn to generate an attention map over the collection of object proposals which results in a higher accuracy VQA accuracy gain compared to the ResNet152 features.

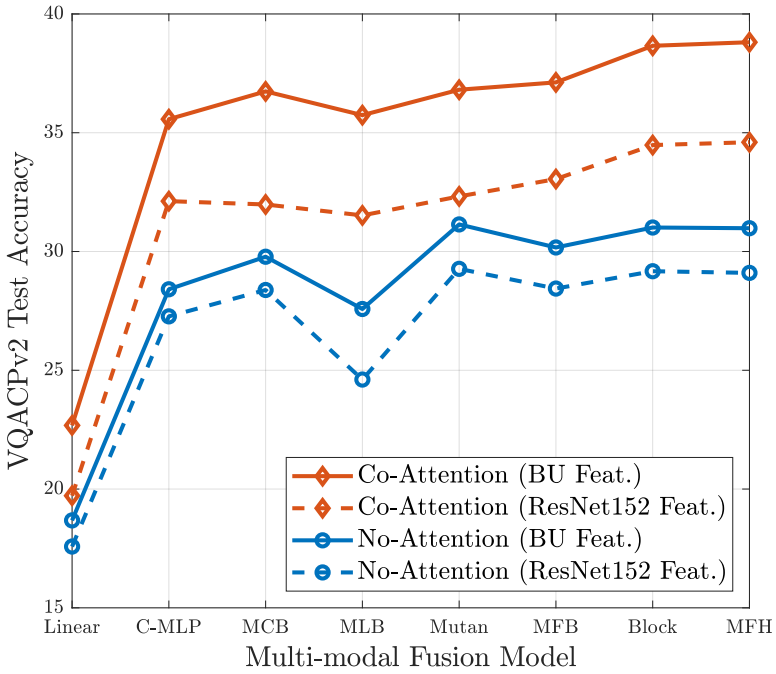


Figure 7.4: *Comparing VQA-CPv2 test accuracy of Co-Attention and No-Attention version of our Unified VQA model, using ResNet152 Spatial Grid (SG) and Bottom-Up (BU) features.*

– *BU features provide greater accuracy boost in datasets with less language bias.* Comparing Fig. 7.3 and Fig. 7.4, we can see that the accuracy gain from using BU features is greater on the VQA-CPv2 dataset compared to the VQAv2 dataset. The no-attention models on the VQAv2 dataset have an average accuracy gain of 1.3 when using BU features instead of ResNet features across all fusion models, whereas the gain is 1.8 on the VQA-CPv2 dataset (Fig. 7.3). This average gain in accuracy is even higher when using the co-attention model, 2.7 on VQAv2 and 4.1 on VQA-CPv2 dataset. This is because the BU features encode an additional attention in the form of object proposals whereas ResNet152 or other SG features only provide features uniformly distributed over the spatial grid. The VQA-CPv2 dataset compared to the VQAv2 dataset has less language bias, thus BU features provide a higher accuracy boost in a more challenging setting.

Even though using BU features instead of ResNet or other SG features improve the VQA accuracy, there is a significant training cost associated with generating BU features which is discussed in more details in the following section (Sec. 7.5.2).

VQA models are less sensitive to change in batch size. In Fig. 7.5 we report the VQA accuracy of our co-attention model using Block fusion using different CNN extracted SG features and BU features by varying the training batch size from 2 to

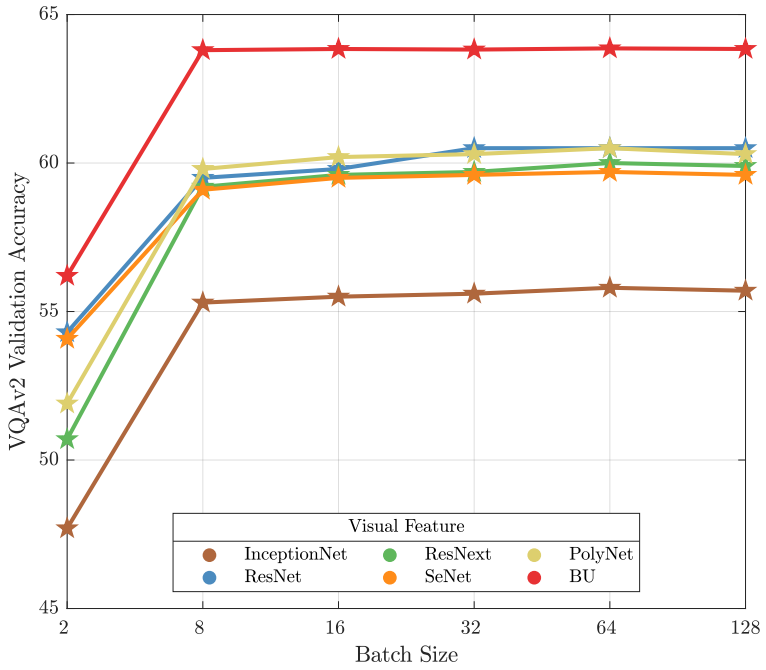


Figure 7.5: *Batch Size vs. VQA accuracy using different CNN backbones used to extract SG features employing Block fusion on VQAv2 validation set.*

128. We used a single GPU configuration to perform these experiments for a more robust evaluation and fairer comparison. We see that except for the choice of smallest batch size of 2, the VQA accuracy saturate between 8 to 128. For almost all visual feature types, we found the optimal batch size to be 64. Furthermore, the choice of batch size also depends GPU memory. One can choose a larger batch size and distribute the computational load across multiple GPUs.

7.5.2 Employing different fusion models

In our analysis, we evaluate the complexity of a VQA model in terms of number of trainable parameters, FLOPS and computation time (both CPU and GPU). The number of trainable parameters in a VQA models mostly depends on the type of fusion models, size of the input visual feature embedding and the candidate answer space. We keep the language feature embedding size and the dimension of candidate answer size fixed for our experiments. The visual feature size varies depending feature extraction meta-architecture which is predefined, and does not contribute to the calculations of trainable parameters or FLOPS except when using BU features. The main variation in the complexity and accuracy calculation comes from the bilinear fusion used in the VQA model. In this section we investigate these VQA accuracy vs. complexity relations by varying the fusion meta-architectures in the VQAv2 and VQA-CPv2 dataset. In this part of the analysis we do not include MCB fusion mechanism as its original implementation was in Caffe [Jia et al., 2014] which

	Linear	C-MLP	MCB	MLB	Mutan	MFB	Block	MFH
Scene	51.0	92.9	93.0	92.6	92.3	92.2	92.8	92.9
Sport	19.0	93.8	92.8	93.5	93.1	93.5	93.6	93.8
Color Att.	55.7	68.5	68.5	68.6	66.3	67.8	68.7	67.0
Other Att.	0.1	56.4	56.7	56.4	52.1	57.2	58.0	55.9
Activity	0.0	52.4	52.4	49.0	49.6	52.7	53.2	51.8
Position	7.28	32.2	35.4	33.5	29.4	32.8	36.1	34.7
Sub-Obj	23.9	86.1	85.4	85.8	85.8	85.9	86.3	86.1
Absurd	90.3	92.5	84.4	90.3	90.0	93.5	90.7	93.3
Utility	15.2	26.3	35.0	31.6	27.5	31.6	34.5	35.7
Presence	93.5	94.4	93.6	93.7	93.9	93.7	94.2	94.1
Counting	50.1	53.1	51.0	51.1	51.3	51.2	52.2	50.7
Sentiment	56.3	65.8	66.3	64.0	63.3	65.3	66.1	63.3
AMPT	38.5	67.9	67.9	67.5	66.2	68.1	68.9	68.3
HMPT	0.0	57.4	60.5	58.7	55.8	59.2	61.1	60.3
N-AMPT	29.8	53.8	42.5	65.3	53.6	53.4	54.8	55.6
N-HMPT	0.0	28.5	27.3	32.2	32.6	30.2	34.2	38.6
Accuracy	73.0	84.0	81.9	83.1	82.7	83.6	83.6	**84.3**

Table 7.3: *Evaluation on the testset of TDIUC [Kafle and Kanan, 2017] dataset with Spatial Grid (SG) ResNet152 features. The first 12 rows report the unnormalized accuracy for each question-type. We report Arithmetic MPT (AMPT) and Harmonic MPT (HMPT) of accuracy scores for all question types alongwith their normalized counterparts N-AMPT and N-HMPT. We also report the traditional VQA accuracy in the last row.*

was incompatible our with trainable parameters and FLOPS calculation method.

Baseline C-MLP model achieves comparable VQA accuracy in TDIUC and VQAv2 dataset. For performing evaluation on contemporary VQA datasets, we establish two simple baseline models, namely Linear Summation (Linear) and Concatenation MLP (C-MLP), for a more robust comparison with the state-of-the-art methods. Surprisingly, the simplistic C-MLP model achieves the second highest overall VQA accuracy in the TDIUC testset (Tab. 7.3) after MFH. It also performs reasonably well in VQAv2 dataset and achieves better VQA accuracy then several state-of-the-art bilinear fusion models. However, in the more challenging VQA-CPv2 dataset, C-MLP performs worse than most of the fusion models. Also, in the TDIUC dataset, the harmonic MPT of C-MLP is less than than other fusion models, because harmonic MPT is skewed towards the question type that has less accuracy. The C-MLP models learns a very high dimensional representation of the image and question distribution through MLP, thus achieves reasonably well in terms of VQA accuracy, but is less generalizable. This means that C-MLP models finds it harder to answer questions requiring superior reasoning capability (eg., object Utility, relative position), but can easily and more accurately answer question levereging language cues (eg., color attribute, scene recognition).

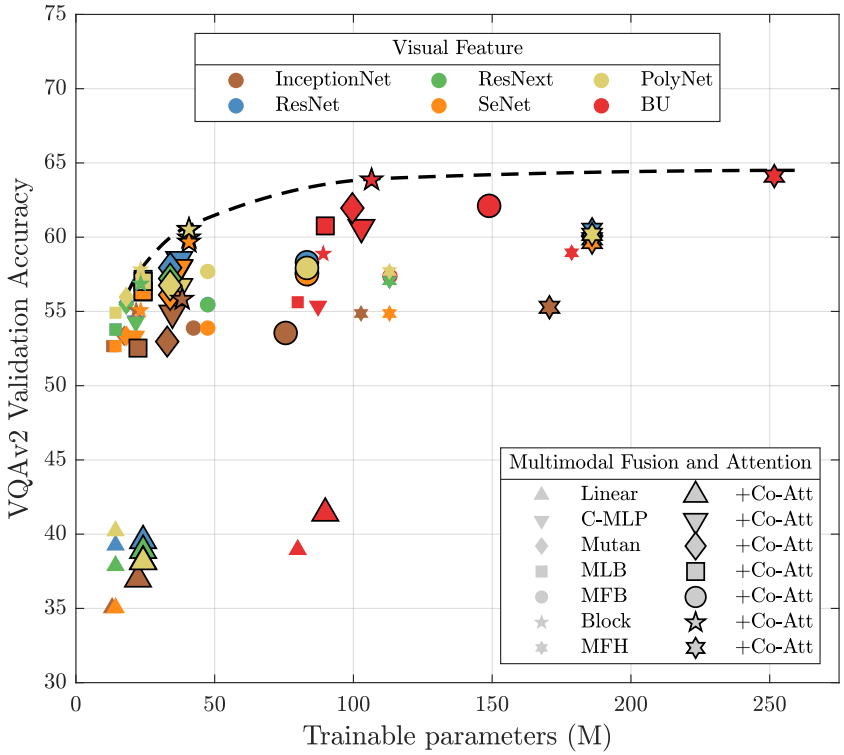


Figure 7.6: *The trade-off between VQAv2 validation accuracy vs. the number of trainable parameters.*

MFH fusion achieved highest VQA accuracy across all datasets in all settings. MFH achieves the highest VQA accuracy compared to other fusion models on the VQAv2 validation set, VQA-CPv2 testset and TDIUC dataset. MFH is also consistent when different CNN extracted IL or SG features and BU features are provided. For example, MFH achieves 0.74 ↑ higher accuracy compared to second best fusion model Block while using SeNet154 features on the VQA-CPv2 dataset (Tab. 7.2). Further, it achieved the highest normalized HMPT (N-HMPT) score among all fusion models which means that the MFH bilinear fusion generalizes well across different question types (Tab. 7.3).

7.5.2.1 Training parameters vs. VQA accuracy

In Fig. 7.6 and Fig. 7.7 we compare VQA accuracy vs. model complexity (number of trainable parameters) respectively on the VQAv2 validation and VQAv2-CP test datasets. Each point in these figures represents a VQA model that was trained on the training set and evaluated on the respective validation/test set. Models employing different visual features are color coded and different fusion strategies are represented by different shapes. The VQA models using a no-attention mechanism are represented with small-sized shapes whereas the models with co-attention mecha-

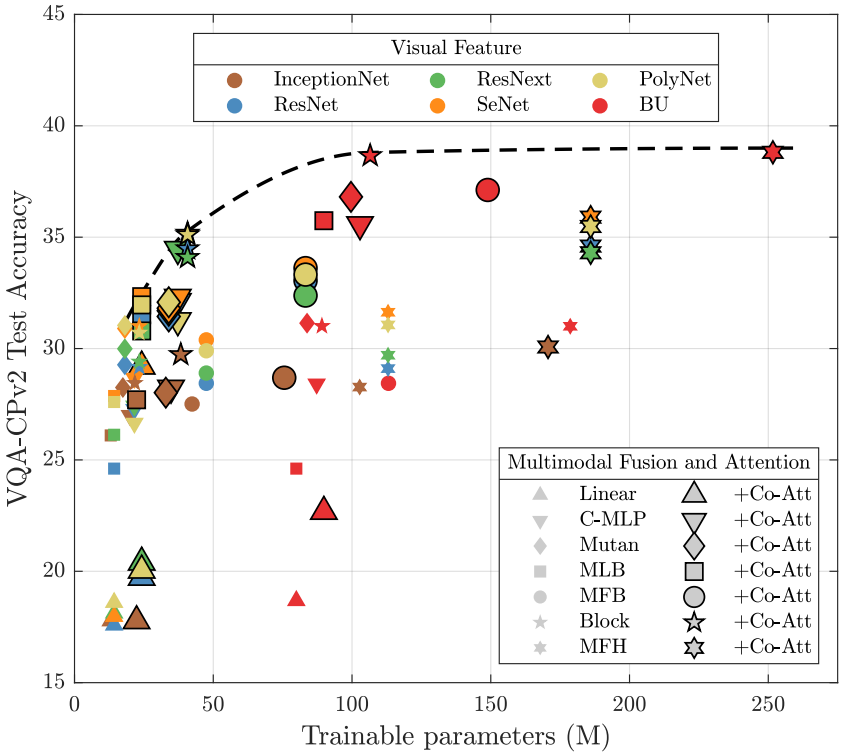


Figure 7.7: *The trade-off between VQA-CPv2 test accuracy vs. the number of trainable parameters.*

nism are represented with a larger shape size. Further, in both of the figures, we plot an imaginary *efficiency line* that infers how much better accuracy can be achieved at what additional computational cost.

Models using BU features are mostly on the maximum efficiency line at an additional complexity cost. The VQA models employing superior fusion mechanisms while using BU features achieve the highest accuracy and sit on the maximum efficiency line. Generating the BU features has an additional training cost where a Faster-RCNN [Ren et al., 2015b] model pretrained on the ImageNet dataset is again retrained on the Visual Genome dataset [Krishna et al., 2016] with a 1600 object and 400 attribute classes. To calculate the additional training cost for generating bottom-up attention, we modified the PyTorch implementation of Faster-RCNN[5] with the additional object and attribute classes, because the original implementation of BU[Anderson et al., 2018] was on a legacy version[6] of Caffe[Jia et al., 2014], incompatible with our complexity calculation method. This was done in an effort to simulate a realistic training pipeline to estimate the additional training cost. Based on our estimation, we offset the number of trainable parameters of VQA models using BU features by $65.65M$ for a more fairer comparison. Further, as discussed earlier,

[5]https://github.com/jwyang/faster-rcnn.pytorch
[6]https://github.com/peteanderson80/bottom-up-attention

one can select the N top objects and their 2048-d visual features as BU features. In our experiments we use $N = 36$, but the original BU features have up to 100 object proposals per image, making the visual feature dimensions even higher which in turn can further increase the number of trainable parameters.

VQA models employing the same bilinear fusion are clustered together. In the VQA accuracy vs. trainable parameters plot we can see the models employing the same fusion mechanism are clustered together and the MFH model has the highest number of trainable parameters. Compared to MFH with BU features, the Block bilinear fusion also performs closer to MFH both on VQAv2 and VQA-CPv2 dataset. Interestingly, we see that the Block model performs significantly worse when using visual features other than BU features compared to similar models using MFH.

Models using co-attention achieve a higher VQA accuracy with an added complexity. As illustrated in Fig. 7.1, the co-attention mechanism has a second bilinear fusion that adds to the overall complexity of the model. However, all the better performing models include the additional a co-attention mechanism, and in VQA-CPv2 (Fig.7.7) the effect of attention is even more prominent than on the VQAv2 dataset (Fig. 7.6) with the same number of additional trainable parameters.

7.5.2.2 FLOPS vs. VQA accuracy

In Fig. 7.8 and Fig. 7.9 we compare VQA accuracy vs. FLOPS (FLoating point OPerations per Second) respectively on the VQAv2 validation and VQAv2-CP test dataset. We use Thop[7] to calculate the number of FLOPS. Similar to offsetting the number of trainable parameters while using BU features, we include an offset of 687 Giga-FLOPS to generate 36 object proposals and the associated BU features. Below, we summarize the key findings.

Processing SG features requires more FLOPS compared to BU features. For our experiments, the BU features are 36×2048 dimensional and the SG features, such as from ResNet152, are 196×2048 dimensional. The visual feature dimension is directly proportional to the FLOPS. Specifically, as the visual feature dimension increases, the FLOPS count increases exponentially.

MFH model requires the highest number of FLOPS. As expected given the higher number of training parameters, the MFH bilinear fusion based VQA models require the largest number of FLOPS compared to other joint embedding models. The second highest FLOPS count is that of MFB fusion approach followed by the Block. Simplistic fusion strategies such as Linear and C-MLP require the least number of FLOPS.

7.5.2.3 Computation time vs. VQA accuracy

We use `torch.autograd.profile`[8] to report the CPU and GPU times. We use a no-attention model in this experiment because we only want to know which joint

[7] https://github.com/Lyken17/pytorch-OpCounter
[8] https://pytorch.org/docs/stable/autograd.html

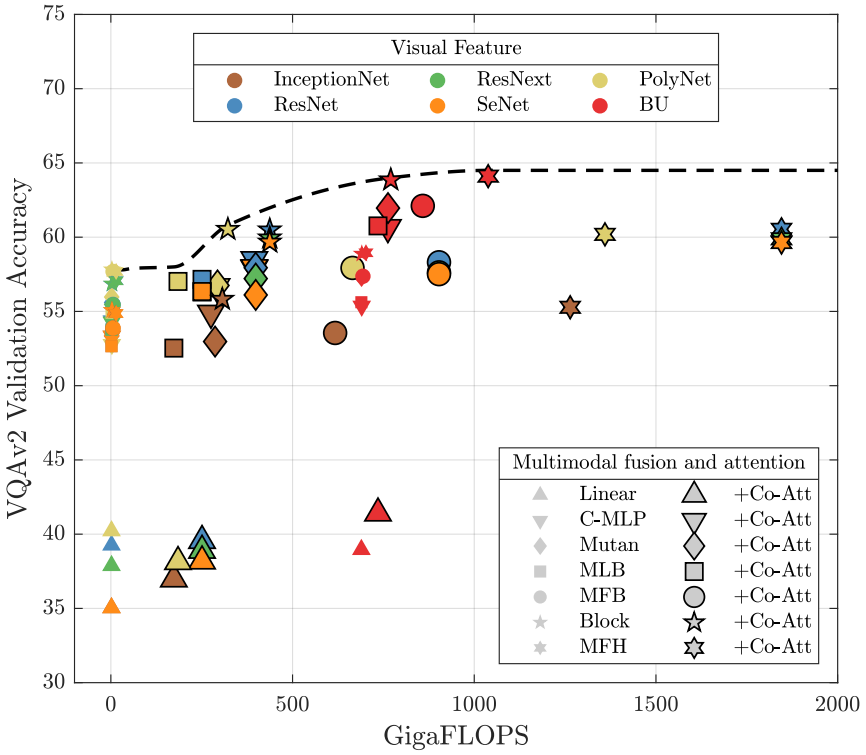


Figure 7.8: *The trade-off between VQAv2 validation accuracy vs. FLOPS.*

embedding model is faster/slower in a comparable setting and the trend we find stays applicable for the co-attention based models. We set the batch size at 64 with one Tesla P100 SXM2 16GB GPU and report the average computation time of 10 mini-batches, each containing 10 Image-Question-Answer (IQA) triplets during the training time. We perform the evaluation on the VQAv2 validation dataset using ResNet152 features ($d_v = 2048 \times 14 \times 14$) and report our findings in Fig. 7.10. The left x-axis of Fig. 7.10 represent the VQA accuracy and the right x-axis represents the computation time in micro-seconds (μS). We do not factor the time for I/O operations as it might vary arbitrarily depending on the system configuration. We use the same system configuration for all our experiments so that the result is not biased by the I/O operation.

Block fusion model takes a significantly longer time than other fusion models. The Block fusion model achieves second-best VQA accuracy but requires a significantly longer time than other fusion models. This is because the Block model decomposes the core tensor into multiple blocks/chunks which separately embed a fraction of the input feature representation in the joint embedding space, and the computational time exponentially increases when the number of blocks increase. On the other hand, even though MFH has more trainable parameters, it achieves a higher VQA

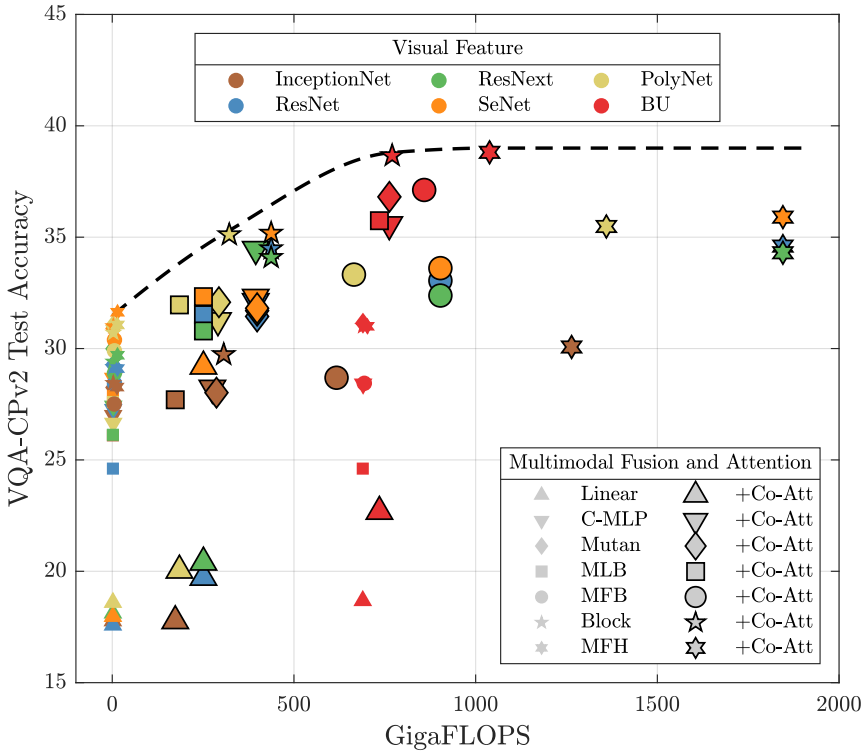


Figure 7.9: *The trade-off between VQA-CPv2 test accuracy vs. FLOPS.*

accuracy with a fraction of CPU and GPU time compared to Block model.

7.5.3 Effect of Co-attention meta-architecture

Adding Co-attention results in more VQA accuracy gain on the challenging VQAv2-CP dataset. Throughout our experiments we found that co-attention mechanism improves VQA accuracy at the cost of some additional complexity. With the co-attention mechanism, a VQA model is able to learn a question-specific attention distribution over the image and its parts, which is more important when experimenting on the VQA-CPv2 dataset. From Tab. 7.1 and Tab. 7.2, we see that across all fusion models (except for Linear) and visual features, the average accuracy gain with co-attention mechanism on the VQAv2 validation set is 2.32 and on the VQA-CPv2 test set the gain is 3.02. This suggests that a superior attention mechanism is a necessary requisite when the experimentation dataset is more challenging and requires intelligent reasoning.

Linear model with co-attention achieves 11× accuracy gain on the VQA-CPv2 dataset, compared to the VQAv2 dataset with SeNet154 features. With the Linear baseline model, we simply sum up the linear projections of the input feature and

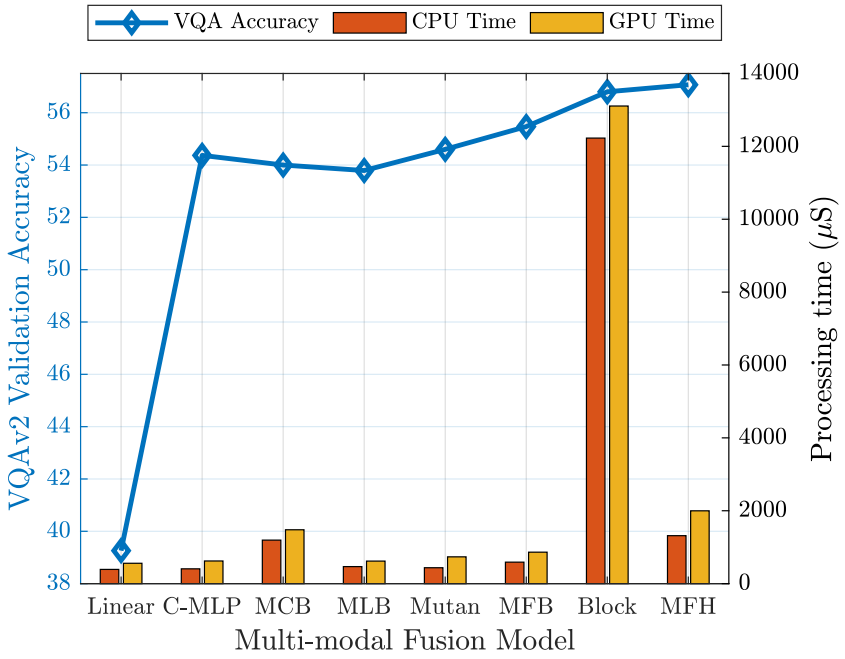


Figure 7.10: *Computation time (CPU and GPU) while employing ResNet152 image-level (IL) features with different fusion models.*

project them back to the answer embedding space. Interestingly, when the Linear model is used with co-attention, the accuracy gain is very low (0.4 across all CNN extracted features) on the VQAv2 validation dataset compared to other fusion models, even sometimes negative. For example, with the Linear model using PolyNet features, the accuracy drops by 2.08 when co-attention is used (Tab. 7.1). On the other hand, when experimenting on the VQA-CPv2 dataset, Linear models with co-attention report high VQA accuracy gain, 11.14 and 6.51 while using SeNet154 features and PolyNet features respectively, with an average accuracy gain of 4.5 (more than 11$\times$ increase) across all CNN extracted visual features (Tab. 7.2).

7.5.4 Proposed meta-architecture recommendation

We did an extensive evaluation of different meta-architectures to study the accuracy vs. complexity trade-off for VQA models. We recommend two settings based on our evaluation. *First*, a less computationally expensive setting that achieves reasonable performance with faster training and inference time. Further, we recommend a *second* setting, that achieves state-of-the-art performance on VQAv2, VQA-CPv2 and TDIUC datasets.

7.5.4.1 Low complexity setting

From our analysis, we see that using BU features yield a better VQA accuracy but have significant additional training cost. Further, incorporating BU features in an end-to-end setting can be challenging and is less generalizable. Thus, for a low complexity setting we recommend using CNN extracted SG features with co-attention, specifically the SeNet features. SeNet encodes additional channel attention, thereby the visual features are more discriminative and have the same feature dimension as popular ResNet features. For the fusion model, we recommend to use the C-MLP model because it performs very close to the state-of-the-art on three benchmark VQA datasets and is lightweight. Further, C-MLP is comparatively easy to implement and can be modified to increase or decrease the complexity of the model by simply changing the hidden dimensions of the MLP layers. This would allow a practical VQA setup flexibility with reasonable VQA performance.

7.5.4.2 High VQA accuracy setting

For a VQA model to achieve the best accuracy, we recommend using pre-processed BU features with attention and the MFH bilinear model to jointly embed visual and semantic features. Our dataset-wise results are given below.

TDIUC dataset: Without modifying our Unified VQA model (as in Fig. 7.1), using MFH with BU features and co-attention, we achieved state-of-the-art performance on the TDIUC dataset. We report the accuracy of our U-VQA model against other state-of-the-art methods in Tab. 7.4

Model	Accuracy	AMPT	HMPT	N-AMPT	N-HMPT
NMN [Andreas et al., 2016]	79.56	62.59	51.87	34.00	16.67
MCB [Fukui et al., 2016]	81.86	67.90	60.47	42.24	27.28
RAU [Noh and Han, 2016]	84.26	67.81	59.99	41.04	23.99
QAS [Shi et al., 2018]	85.03	69.11	60.08	-	-
Block [Ben-Younes et al., 2019]	85.96	71.84	65.52	58.36	39.44
Ours(U-VQA+MFH)	**86.33**	**81.54**	**65.81**	**58.80**	**41.76**

Table 7.4: *Comparison with state-of-the-art methods on TDIUC [Kafle and Kanan, 2017] testset.*

VQAv2 dataset: MCAN [Yu et al., 2019] currently achieves the state-of-the-art performance on VQAv2 dataset by employing a deep modular co-attention mechanism. Throughout our experiments, we found out that an additional attention mechanism can help VQA models achieve better accuracy. For achieving higher accuracy than MCAN on VQAv2 dataset, we adopt their implementation of deep modular co-attention[9]. They use a linear multimodal fusion operation to jointly embed attended image and question features before the classification layer. As we found MFH to be a

[9]https://github.com/MILVLG/openvqa/tree/master/openvqa/models/mcan

superior bilinear fusion model, we replace the linear fusion operation with MFH and use BU features for our experiments. Even though, MCAN is a highly engineered setup that achieves state-of-the-art performance on the saturated VQAv2 dataset, following our meta-architecture recommendations, we report overall VQA accuracy improvements of 0.13 and 0.18 respectively on VQAv2 test-dev and test-std datasets (Tab. 7.5).

	Test-dev				Test-std
Model	Overall	Y/N	Number	Other	Overall
MCB[Fukui et al., 2016]	62.27	78.46	38.28	57.80	53.36
BU [Anderson et al., 2018]	65.32	81.82	44.21	56.05	65.67
Mutan[Ben-Younes et al., 2017]†	66.01	82.88	44.54	56.50	66.38
MLB[Kim et al., 2016]†	66.27	83.58	44.92	56.34	66.62
RAF[Farazi and Khan, 2018]	67.20	84.10	44.90	57.80	67.40
Block[Ben-Younes et al., 2019]	67.58	83.60	47.33	58.51	67.92
MuRel[Cadene et al., 2019a]	68.03	84.77	49.84	57.85	68.41
Counter[Zhang et al., 2018]	68.09	83.14	51.62	58.97	68.41
MFH [Yu et al., 2018]	68.76	84.27	49.56	59.89	-
BAN[Kim et al., 2018]	69.52	85.31	50.93	60.26	-
BAN+Counter[Kim et al., 2018]	70.04	85.42	**54.04**	60.52	70.35
MCAN[Yu et al., 2019]	70.63	86.82	53.26	60.72	70.90
Ours (MCAN+MFH)	**70.76**	**87.1**	53.21	**60.77**	**71.08**

Table 7.5: *Comparison with state-of-the-art methods on VQAv2 [Agrawal et al., 2018] test-dev and test-std dataset. (†) reported from [Cadene et al., 2019a].*

Model	Overall	Y/N	Number	Other
VQA [Antol et al., 2015]	19.73	34.25	11.39	14.41
NMN [Andreas et al., 2016]	27.47	38.94	11.92	25.72
MCB [Fukui et al., 2016]	36.33	41.01	11.96	40.57
GVQA [Agrawal et al., 2018]	31.30	57.99	13.68	22.14
MuRel [Cadene et al., 2019a]	39.54	42.85	13.17	45.04
Q-Adv [Ramakrishnan et al., 2018]	41.71	64.49	15.48	35.48
RUBi [Cadene et al., 2019b]	47.11	68.65	20.28	43.18
Ours (RUBi+MFH)	**48.44**	**73.04**	**21.43**	**43.44**

Table 7.6: *Comparison with state-of-the-art methods on VQA-CPv2 [Agrawal et al., 2018] testset.*

VQA-CPv2 dataset: Similar to our approach on the VQAv2 dataset, we found that RUBi [Cadene et al., 2019b] currently achieves that state-of-the-art performance by adding an additional question only branch that reduces the language bias inherent to the dataset. This approach is particularly useful on the VQA-CPv2 dataset since

by its design the train and test splits have different semantic distribution. In their baseline architecture, they use a Block fusion model to jointly embed visual and semantic features. We replace the Block fusion model in their baseline architecture[10] with MFH and report a 1.33 accuracy gain over the current state-of-the-art (Tab. 7.6).

7.6 Conclusion

Visual question answering (VQA) is a challenging problem that is actively under investigation. A range of existing approaches exist in the literature, all developed with different ingredients, that makes it difficult to make a fair comparison between them. In this chapter, we systematically study the influence of key components commonly used within VQA models on the efficiency and final performance. We performed extensive evaluation on three benchmark VQA datasets by varying the VQA meta-architecture. Based on our extensive experiments, we provide two recommendations for meta-architecture selection. One focuses on achieving reasonable VQA accuracy with a simple and light weight architecture, while the other focuses on achieving the state-of-the-art accuracy on VQAv2, VQA-CPv2 and TDIUC datasets. Our finding suggests VQA models using visual features obtained by Squeeze-and-excitation Network (SeNet [Hu et al., 2018]) mostly outperform models using widely adopted ResNet [He et al., 2016] features, irrespective of attention and fusion mechanism. Further, we report that employing MFH fusion facilitates achieving a superior performance over its counterparts. We hope that our findings and recommendations will help researchers to find optimum design choices for VQA and other multi-modal tasks based on vision and language inputs.

[10] https://github.com/cdancette/rubi.bootstrap.pytorch

Chapter 8

Conclusion and Future Directions

The development of intelligent systems that can answer complex natural language questions about visual data can help enhance visual reasoning capability of AI agents. In this dissertation, we introduced multi-level visual attention, exemplar based learning and semantic relationship modelling to improve visual reasoning. However, for an AI-complete task such as VQA, there exist several future challenges that need to be addressed for building human-level artificial intelligence system. In this chapter, we summarize the contributions of this dissertation which bridges some of the knowledge gaps and identify future challenges in this line of research.

8.1 Summary

Drawing inspiration from human visual perception, we develop visual question answering models that can comprehend holistic understanding of the scene by modelling multi-modal interaction between visual and linguistic cues, incorporating different levels of visual and/or semantic attention, making use of transfer learning to reason about the unknown objects and concepts, and generating semantic relationship labels to infer about the subtle interactions between objects. Here we summarize the key contributions of this thesis chapter-wise:

- Chapter 3 proposes a reciprocal attention mechanism that incorporates both image and object level visual features.
- Chapter 4 proposes a question agnostic attention mechanism which leverages visual cues that are not learned with respect to the question, rather obtained from object instances.
- Chapter 5 reformulates VQA in a transfer learning framework and proposes an exemplar based approach for reasoning about the unknown objects and concepts.
- Chapter 6 instigates the use of semantic relationship features to achieve higher level visual reasoning ability for a VQA model.
- Chapter 7 serves as a guide to effectively navigate the complexity vs. accuracy trade-off for developing visual and language models.

Figure 8.1: *An example depicting future challenges in Visual Question Answering.* Example sourced from Visual Genome dataset [Krishna et al., 2016]

8.2 Challenges and Future Directions

VQA is an AI-complete task and is far from being solved by existing vision and language models. The recent approaches to solve the vision-language problem are mostly *vision heavy*. What this means is that VQA algorithms are primarily focused on extracting more discriminative visual feature representations from an image and identifying salient regions in it with respect to the questions asked. This trend is true for other vision and language tasks like image captioning [Anderson et al., 2016], visual dialogue [Das et al., 2017b], visual storytelling [Huang et al., 2016], video summarizing [Otani et al., 2016], to name a few. If we compare such *vision heavy* system with an infant learning to interact with the real word, the infant would be in its the early developmental stages, where the infant excels at vision tasks, such as identifying objects, realizing their attributes and functionality, and begins to answer questions about everyday object it sees. Recent, vision systems achieved near human accuracy in object detection, recognition, attribute classification and several other vision task on benchmark datasets. However, similar to the infant's learning, it takes more effort and time to learn and comprehend the functionality, features and usability of objects around us. To this end, in the first part of this dissertation, the contribution is mostly around the vision part of the problem, in later parts we focus more on improving semantic understanding of the image contents.

Fig. 8.1 illustrates some of the open challenges for developing visual question answering models. If a VQA models is asked 'Who is pictured?' showing the image in Fig. 8.1, a VQA model that can identify two persons in the image, might answer 'Two people'. However, the ground truth answer from the human annotators of Visual Genome dataset [Krishna et al., 2016] for this image-question pair is 'Father and Daughter'. How does a VQA model go from answering 'Two people' to 'Father and daughter' is a major challenge which vision and language models need to address. To answer the question like a human, **first,** the VQA models should solve the vision problem by detecting two person and two sets of water skis in the image, and identi-

fying attributes that one person is an adult male and another is a little girl. **Second,** the VQA model is required to gain a higher-level understating of the scene by picking up subtle visual and semantic cues e.g., the adult male is holding the handle of the ski rope with one hand and holding the young girl with the other. Using these cues the model should deduce two possible relationships, <*adult male*> - <*holding*> - <*little girl*> or <*adult male*> - <*helping*> - <*little girl*>, both of which are correct but the latter encapsulates much richer scene understanding. **Third,** with a richer understating of the visual scene, the VQA models would be able to identify the relationship between the adult male and the little girl as 'helping' rather than merely 'holding'. **Fourth,** combining all of these together the model should gain a holistic understanding of the scene and identify two most plausible answer to the question as 'Ski instructor and trainee' and 'Father and daughter', and choosing the latter. For a VQA model to predict that latter would require it to look at many similar examples of father helping his daughter doing an activity, their facial expressions and other salient visual and semantic cues. There is no way to be certain only by looking at an image that they are indeed father and daughter, but like humans, an AI agent must be able to make an educated guess. In Chapter 5 and 6, we make contributions that enable VQA models to transfer knowledge from previously seen examples and make use of high-level semantic relations. However, following are some of the future directions that would need to be explored to help VQA models achieve such enhanced human-level reasoning capabilities.

- **Long tail answer distribution:** VQA is traditionally framed as a multi-class classification problem due to the long tail nature of the answer distribution in large-scale VQA datasets. For example, only choosing 3000 most frequent answers from $1.1M$ image-question pairs in VQAv2 dataset [Krishna et al., 2016] covers $\sim$ 99% of the answers distribution. This setting works well for developing systems that can predict answers from the fixed set of candidate answers, but cannot predict answer that are not in the dataset or in the long tail. One approach is to solve this problem through meta-learning [Finn et al., 2017] or few-shot learning [Snell et al., 2017] approach. In this setup, the model can be trained with fixed set of candidate answers selected from the dataset and can have an arbitrary sized candidate answer set containing entries from the long tail during testing. The model would perform a small training on complementary image-question pairs with the novel answers sourced from external knowledge base to fine-tune learned weights to adapt to a different candidate answer set. This approach would also work for other vision and language datasets with long-tail distribution.

- **VQA evaluation metric:** The evaluation metric for VQA is based on the ground truth annotations provided by the human annotator and the measured accuracy is uniform across all question types. A predicted answer is considered accurate if 3 out of 10 ground truth annotations match the predicted answer. This evaluation approach has two major limitations. **First**, human annotators often cannot come to a consensus and answer the question correctly. For example,

Figure 8.2: *Examples of 'natural' bias in the VQA dataset.* An AI agent produces incorrect answers when subjected to image and question pairs with strong natural bias.

human annotators can provide synonymous answers given a simple question. If the predicted answer is 'sad' instead of 'unhappy' it will be marked incorrect even though they have similar meaning. **Second**, the evaluation metrics treat all questions the same irrespective of their difficulty level, but in reality, some questions are easier to answer than others. This difficulty mostly depends on the reasoning task that the model needs to perform in order to predict the correct answer. If a VQA model is asked 'What color is the water in the picture?' for the image in Fig. 8.1, the models only need to learn the color of sea-water, and can answer 'Blue'. On the other hand, a much harder question would be 'What is the relationship between the man and the little girl in the image?'. Answering this question would require much higher level reasoning. However, same reward/penalty is applied if a the answer provided by the VQA models is correct/incorrect. An adaptive evaluation metric is necessary where it will penalize the model for predicting wrong answer for easy questions and give more reward for being able to answer a hard question correctly.

- **'Natural' bias:** The questions asked about a natural image inherently have a 'natural' bias. For example, in Fig. 8.2, while answering the question 'What is the color of the banana?', a natural image dataset will have more examples of 'Yellow' bananas than 'Green' ones. This knowledge about the natural distribution of banana colours could help the model make an educated guess. This

is particularly useful when the model is operating in an open-world scenario. Meanwhile, if not careful, VQA models learn to *cheat* using this bias which make the model less generalizable across reasoning tasks. In VQA v1 dataset [Antol et al., 2015], the most common answers for the question 'How many' are 'Two' and 'Three'. A model trained on this dataset will learn to answer correctly for most of the counting questions without even learning to count. The bias in the dataset should be controlled in a way that models cannot take advantage of such bias and natural distribution of information is intact which the model can use as common-sense knowledge.

- **Life-long learning:** VQA model should be immune to *catastrophic forgetting* [McCloskey and Cohen, 1989] which would enable VQA models to accumulate and refine their reasoning skills and knowledge base over time. VQA models learn to perform several reasoning tasks which are required to answer different types of question (e.g., counting, positional reasoning, usability and so on). Often when learning a new task, deep neural network can *forget* the models learned (i.e., weights, states) for the previous tasks. For example, if a model specifically trained to answer binary questions is fine-tuned to answer counting questions, its accuracy would decrease on the former task (i.e., binary question answering). However, VQA models should be able to incorporate new reasoning skills to support life-long learning. This would allow multiple models trained on several different tasks to learn from each other and achieve better reasoning skills.

- **Interactive agent:** An interactive agent is a system that can make sense of the physical environment through perception, communicate with other agents and/or humans, and execute any instruction provided. Such interactive agents can not only *see* but also *act*. Visual question answering is the first step towards enabling intelligent agents to interact with humans or with other agents. For example, if an agent is asked 'Do I need to buy milk?', the agent first needs to understand the question and devise a plan for answering the question. The plan can be either to navigate to the fridge and checking on the inventory, or communicating to another intelligent agent that does the inventory management (i.e., smart fridge) to determine the answer to the question. Developing such interactive agents would require computer vision, NLP, internet of things and robotics to seamlessly work together and achieve near human reasoning capability.

Bibliography

Agrawal, A.; Batra, D.; Parikh, D.; and Kembhavi, A., 2018. Don't just assume; look and answer: Overcoming priors for visual question answering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 21, 22, 25, 26, 27, 61, 69, 72, 73, 104, 107, and 120)

Agrawal, A.; Kembhavi, A.; Batra, D.; and Parikh, D., 2017. C-vqa: A compositional split of the visual question answering (vqa) v1. 0 dataset. *arXiv preprint arXiv:1704.08243*, (2017). (cited on pages 25 and 61)

Anderson, P.; Fernando, B.; Johnson, M.; and Gould, S., 2016. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, 382–398. Springer. (cited on page 124)

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L., 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*. (cited on pages 25, 29, 31, 38, 39, 40, 43, 45, 53, 55, 63, 70, 74, 80, 82, 89, 97, 103, 108, 114, and 120)

Andreas, J.; Rohrbach, M.; Darrell, T.; and Klein, D., 2016. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 39–48. (cited on pages 19, 35, 54, 119, and 120)

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Lawrence Zitnick, C.; and Parikh, D., 2015. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, 2425–2433. (cited on pages 4, 15, 17, 21, 22, 24, 27, 29, 35, 37, 43, 48, 49, 50, 66, 67, 68, 70, 72, 80, 93, 94, 104, 105, 120, and 127)

Bader, B. W. and Kolda, T. G., 2007. Efficient matlab computations with sparse and factored tensors. *SIAM Journal on Scientific Computing*, 30, 1 (2007), 205–231. (cited on page 33)

Bansal, A.; Sikka, K.; Sharma, G.; Chellappa, R.; and Divakaran, A., 2018. Zero-shot object detection. (cited on page 59)

Ben-Younes, H.; Cadène, R.; Thome, N.; and Cord, M., 2017. Mutan: Multimodal tucker fusion for visual question answering. *ICCV*, (2017). (cited on pages 24, 25, 30, 32, 33, 34, 35, 36, 38, 40, 44, 46, 48, 50, 51, 52, 53, 63, 70, 71, 72, 73, 74, 80, 81, 88, 94, 100, and 120)

Ben-Younes, H.; Cadene, R.; Thome, N.; and Cord, M., 2019. Block: Bilinear superdiagonal fusion for visual question answering and visual relationship detection. In *AAAI 2019-33rd AAAI Conference on Artificial Intelligence*. (cited on pages 25, 43, 44, 46, 48, 51, 52, 53, 74, 80, 81, 88, 94, 100, 103, 119, and 120)

Borji, A. and Itti, L., 2013. State-of-the-art in visual attention modeling. *IEEE transactions on pattern analysis and machine intelligence*, 35, 1 (2013), 185–207. (cited on page 30)

Cadene, R.; Ben-Younes, H.; Cord, M.; and Thome, N., 2019a. Murel: Multimodal relational reasoning for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1989–1998. (cited on pages 80, 89, and 120)

Cadene, R.; Dancette, C.; Ben-younes, H.; Cord, M.; and Parikh, D., 2019b. Rubi: Reducing unimodal biases in visual question answering. *arXiv preprint arXiv:1906.10169*, (2019). (cited on page 120)

Charikar, M.; Chen, K.; and Farach-Colton, M., 2002. Finding frequent items in data streams. In *International Colloquium on Automata, Languages, and Programming*, 693–703. Springer. (cited on page 98)

Cho, K.; Van Merriënboer, B.; Bahdanau, D.; and Bengio, Y., 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, (2014). (cited on pages 32 and 63)

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, (2014). (cited on page 24)

Das, A.; Agrawal, H.; Zitnick, L.; Parikh, D.; and Batra, D., 2017a. Human attention in visual question answering: Do humans and deep networks look at the same regions? *Computer Vision and Image Understanding*, 163 (2017), 90–100. (cited on page 44)

Das, A.; Kottur, S.; Gupta, K.; Singh, A.; Yadav, D.; Moura, J. M.; Parikh, D.; and Batra, D., 2017b. Visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 326–335. (cited on page 124)

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 248–255. IEEE. (cited on pages 31, 32, 46, 50, and 73)

Desimone, R. and Duncan, J., 1995. Neural mechanisms of selective visual attention. *Annual review of neuroscience*, 18, 1 (1995), 193–222. (cited on page 30)

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, (2018). (cited on pages 2, 5, 80, 81, and 97)

Farazi, M.; Khan, S.; and Barnes, N., 2020a. Attention guided semantic relationship parsing for visual question answering. *arXiv preprint arXiv:2010.01725*, (2020). (cited on page 12)

Farazi, M. R. and Khan, S., 2018. Reciprocal attention fusion for visual question answering. In *The British Machine Vision Conference (BMVC)*. (cited on pages 12, 25, 29, 44, 45, 63, 70, 80, 88, and 120)

Farazi, M. R.; Khan, S. H.; and Barnes, N., 2020b. Accuracy vs. complexity: A trade-off in visual question answering models. *arXiv preprint arXiv:2001.07059*, (2020). (cited on page 12)

Farazi, M. R.; Khan, S. H.; and Barnes, N., 2020c. From known to the unknown: Transferring knowledge to answer questions about novel visual and semantic concepts. *Image and Vision Computing*, 103 (2020), 103985. (cited on pages 12 and 59)

Farazi, M. R.; Khan, S. H.; and Barnes, N., 2020d. Question-agnostic attention for visual question answering. In *Proceedings of the International Conference on Pattern Recognition*. (cited on pages 12 and 43)

Finn, C.; Abbeel, P.; and Levine, S., 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 1126–1135. JMLR. org. (cited on page 125)

Fukui, A.; Park, D. H.; Yang, D.; Rohrbach, A.; Darrell, T.; and Rohrbach, M., 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*, (2016). (cited on pages 25, 32, 34, 35, 36, 38, 40, 43, 46, 48, 50, 53, 54, 63, 70, 71, 72, 73, 81, 88, 94, 98, 101, 103, 119, and 120)

Gao, Y.; Beijbom, O.; Zhang, N.; and Darrell, T., 2016. Compact bilinear pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 317–326. (cited on page 24)

Geman, D.; Geman, S.; Hallonquist, N.; and Younes, L., 2015. Visual turing test for computer vision systems. *Proceedings of the National Academy of Sciences*, 112, 12 (2015), 3618–3623. (cited on page 3)

Geng, C.; Huang, S.-j.; and Chen, S., 2020. Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2020). (cited on page 59)

Girshick, R., 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 1440–1448. (cited on page 1)

Goyal, Y.; Khot, T.; Summers-Stay, D.; Batra, D.; and Parikh, D., 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 15, 17, 21, 22, 26, 27, 29, 36, 37, 38, 39, 40, 48, 49, 52, 64, 66, 67, 68, 73, 85, 104, 105, and 106)

Gu, J.; Joty, S.; Cai, J.; Zhao, H.; Yang, X.; and Wang, G., 2019. Unpaired image captioning via scene graph alignments. In *Proceedings of the IEEE International Conference on Computer Vision*, 10323–10332. (cited on page 17)

Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al., 2018. Recent advances in convolutional neural networks. *Pattern Recognition*, 77 (2018), 354–377. (cited on page 93)

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R., 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2961–2969. (cited on pages 1 and 50)

He, K.; Zhang, X.; Ren, S.; and Sun, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778. (cited on pages 1, 6, 24, 31, 35, 36, 46, 50, 70, 72, 80, 82, 85, 94, 95, 96, 102, and 121)

Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. *Neural computation*, 9, 8 (1997), 1735–1780. (cited on pages 24 and 97)

Hu, J.; Shen, L.; and Sun, G., 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141. (cited on pages 80, 94, 103, and 121)

Hu, R.; Rohrbach, A.; Darrell, T.; and Saenko, K., 2019. Language-conditioned graph networks for relational reasoning. In *The IEEE International Conference on Computer Vision (ICCV)*. (cited on pages 27, 80, and 89)

Huang, T.-H.; Ferraro, F.; Mostafazadeh, N.; Misra, I.; Agrawal, A.; Devlin, J.; Girshick, R.; He, X.; Kohli, P.; Batra, D.; et al., 2016. Visual storytelling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1233–1239. (cited on page 124)

Hudson, D. A. and Manning, C. D., 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6700–6709. (cited on pages 20, 21, 85, and 87)

Ilievski, I. and Feng, J., 2017. Multimodal learning and reasoning for visual question answering. In *Advances in Neural Information Processing Systems*, 551–562. (cited on pages 35 and 36)

Jabri, A.; Joulin, A.; and van der Maaten, L., 2016. Revisiting visual question answering baselines. In *European Conference on Computer Vision*, 727–739. Springer. (cited on pages 24, 29, and 94)

Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; and Darrell, T., 2014. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, (2014). (cited on pages 111 and 114)

Johnson, J.; Hariharan, B.; van der Maaten, L.; Fei-Fei, L.; Zitnick, C. L.; and Girshick, R., 2016. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. *arXiv preprint arXiv:1612.06890*, (2016). (cited on pages 19, 27, and 29)

Johnson, J.; Krishna, R.; Stark, M.; Li, L.-J.; Shamma, D.; Bernstein, M.; and Fei-Fei, L., 2015. Image retrieval using scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3668–3678. (cited on pages 17 and 26)

Judd, T.; Ehinger, K.; Durand, F.; and Torralba, A., 2009. Learning to predict where humans look. In *2009 IEEE 12th international conference on computer vision*, 2106–2113. IEEE. (cited on pages 8 and 55)

Kafle, K. and Kanan, C., 2017. An analysis of visual question answering algorithms. In *Proceedings of the IEEE International Conference on Computer Vision*, 1965–1973. (cited on pages 50, 52, 54, 104, 112, and 119)

Karpathy, A. and Fei-Fei, L., 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3128–3137. (cited on page 2)

Khan, S.; Rahmani, H.; Shah, S. A. A.; and Bennamoun, M., 2018. A guide to convolutional neural networks for computer vision. *Synthesis Lectures on Computer Vision*, 8, 1 (2018), 1–207. (cited on page 6)

Kim, J.-H.; Jun, J.; and Zhang, B.-T., 2018. Bilinear attention networks. In *Advances in Neural Information Processing Systems*, 1564–1574. (cited on pages 25, 89, 94, and 120)

Kim, J.-H.; On, K.-W.; Kim, J.; Ha, J.-W.; and Zhang, B.-T., 2016. Hadamard product for low-rank bilinear pooling. *arXiv preprint arXiv:1610.04325*, (2016). (cited on pages 34, 35, 38, 98, 103, and 120)

Kingma, D. and Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, (2014). (cited on pages 37 and 86)

Kiros, R.; Zhu, Y.; Salakhutdinov, R. R.; Zemel, R.; Urtasun, R.; Torralba, A.; and Fidler, S., 2015. Skip-thought vectors. In *Advances in neural information processing systems*, 3294–3302. (cited on pages 7, 33, 37, 46, 70, 94, 97, and 103)

Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D. A.; et al., 2016. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332*, (2016). (cited on pages 4, 17, 18, 19, 27, 29, 31, 37, 71, 83, 89, 94, 105, 106, 114, 124, and 125)

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105. (cited on page 1)

Li, G.; Duan, N.; Fang, Y.; Jiang, D.; and Zhou, M., 2019a. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. *arXiv preprint arXiv:1908.06066*, (2019). (cited on page 89)

Li, L.; Gan, Z.; Cheng, Y.; and Liu, J., 2019b. Relation-aware graph attention network for visual question answering. In *The IEEE International Conference on Computer Vision (ICCV)*. (cited on pages 27, 80, and 89)

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L., 2014. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 740–755. Springer. (cited on pages 14, 15, 17, 50, 66, 67, and 104)

Lu, C.; Krishna, R.; Bernstein, M.; and Fei-Fei, L., 2016a. Visual relationship detection with language priors. In *European Conference on Computer Vision*, 852–869. Springer. (cited on pages 27, 83, and 86)

Lu, J.; Yang, J.; Batra, D.; and Parikh, D., 2016b. Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*, 289–297. (cited on pages 24, 29, 35, 36, 40, 45, 72, 73, and 94)

Lu, P.; Li, H.; Zhang, W.; Wang, J.; and Wang, X., 2017. Co-attending free-form regions and detections with multi-modal multiplicative feature embedding for visual question answering. *arXiv preprint arXiv:1711.06794*, (2017). (cited on pages 35 and 39)

Lu, P.; Li, H.; Zhang, W.; Wang, J.; and Wang, X., 2018. Co-attending free-form regions and detections with multi-modal multiplicative feature embedding for visual question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*. (cited on page 44)

Malinowski, M. and Fritz, M., 2014. A multi-world approach to question answering about real-world scenes based on uncertain input. In *Advances in neural information processing systems*, 1682–1690. (cited on page 13)

Malinowski, M.; Rohrbach, M.; and Fritz, M., 2017. Ask your neurons: A deep learning approach to visual question answering. *International Journal of Computer Vision*, 125, 1-3 (2017), 110–135. (cited on pages 14 and 35)

McCloskey, M. and Cohen, N. J., 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, vol. 24, 109–165. Elsevier. (cited on pages 61 and 127)

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J., 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, (2013). (cited on page 1)

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J., 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119. (cited on pages 1, 7, and 80)

Nam, H.; Ha, J.-W.; and Kim, J., 2016. Dual attention networks for multimodal reasoning and matching. *arXiv preprint arXiv:1611.00471*, (2016). (cited on pages 35 and 80)

Noh, H. and Han, B., 2016. Training recurrent answering units with joint loss minimization for vqa. *arXiv preprint arXiv:1606.03647*, (2016). (cited on pages 54 and 119)

Norcliffe-Brown, W.; Vafeias, S.; and Parisot, S., 2018. Learning conditioned graph structures for interpretable visual question answering. In *Advances in Neural Information Processing Systems*, 8334–8343. (cited on page 89)

Otani, M.; Nakashima, Y.; Rahtu, E.; Heikkilä, J.; and Yokoya, N., 2016. Video summarization using deep semantic features. In *Asian Conference on Computer Vision*, 361–377. Springer. (cited on page 124)

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S., 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32* (Eds. H. Wallach; H. Larochelle; A. Beygelzimer; F. d Alché-Buc; E. Fox; and R. Garnett), 8024–8035. Curran Associates, Inc. (cited on page 104)

Patro, B. and Namboodiri, V. P., 2018. An empirical evaluation of visual question answering for novel objects. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 26 and 73)

Pennington, J.; Socher, R.; and Manning, C., 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543. (cited on pages 1, 24, 46, 80, and 97)

Rahman, S.; Khan, S.; and Barnes, N., 2018a. Polarity loss for zero-shot object detection. *arXiv preprint arXiv:1811.08982*, (2018). (cited on page 59)

Rahman, S.; Khan, S.; and Porikli, F., 2018b. Zero-shot object detection: Learning to simultaneously recognize and localize novel concepts. *Asian Conference on Computer Vision (ACCV)*, (2018). (cited on page 59)

Ramakrishnan, S.; Agrawal, A.; and Lee, S., 2018. Overcoming language priors in visual question answering with adversarial regularization. In *Advances in Neural Information Processing Systems*, 1541–1551. (cited on page 120)

Ramakrishnan, S. K.; Pal, A.; Sharma, G.; and Mittal, A., 2017. An empirical evaluation of visual question answering for novel objects. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 7312–7321. IEEE. (cited on pages 25, 61, 69, 72, and 73)

Ren, M.; Kiros, R.; and Zemel, R., 2015a. Exploring models and data for image question answering. In *Advances in Neural Information Processing Systems*, 2953–2961. (cited on page 14)

Ren, S.; He, K.; Girshick, R.; and Sun, J., 2015b. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 91–99. (cited on pages 1, 6, 25, 27, 31, 80, 82, 85, 97, and 114)

Santoro, A.; Raposo, D.; Barrett, D. G.; Malinowski, M.; Pascanu, R.; Battaglia, P.; and Lillicrap, T., 2017. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, 4967–4976. (cited on page 27)

Shapiro, S. C., 1992. *Encyclopedia of artificial intelligence second edition*. John. (cited on page 5)

Shi, Y.; Furlanello, T.; Zha, S.; and Anandkumar, A., 2018. Question type guided attention in visual question answering. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 151–166. (cited on page 119)

Shih, K. J.; Singh, S.; and Hoiem, D., 2016. Where to look: Focus regions for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4613–4621. (cited on pages 24 and 29)

Silberman, N.; Hoiem, D.; Kohli, P.; and Fergus, R., 2012. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, 746–760. Springer. (cited on page 13)

Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, (2014). (cited on pages 1, 6, 24, and 93)

Snell, J.; Swersky, K.; and Zemel, R., 2017. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, 4077–4087. (cited on page 125)

Sukhbaatar, S.; Weston, J.; Fergus, R.; et al., 2015. End-to-end memory networks. In *Advances in neural information processing systems*, 2440–2448. (cited on page 24)

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A., 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9. (cited on pages 1, 35, and 102)

Tan, H. and Bansal, M., 2019. Lxmert: Learning cross-modality encoder representations from transformers. 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)*, 5103–5114. (cited on page 89)

Teney, D.; Anderson, P.; He, X.; and van den Hengel, A., 2018. Tips and tricks for visual question answering: Learnings from the 2017 challenge. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 26, 36, 38, and 81)

Teney, D. and Hengel, A. v. d., 2016. Zero-shot visual question answering. *arXiv preprint arXiv:1611.05546*, (2016). (cited on pages 25, 26, 61, and 69)

Teney, D.; Liu, L.; and Hengel, A. v. d., 2016. Graph-structured representations for visual question answering. *arXiv preprint arXiv:1609.05600*, (2016). (cited on page 27)

Teney, D. and van den Hengel, A., 2018. Visual question answering as a meta learning task. In *The European Conference on Computer Vision (ECCV)*. (cited on pages 26, 73, and 74)

Thomee, B.; Shamma, D. A.; Friedland, G.; Elizalde, B.; Ni, K.; Poland, D.; Borth, D.; and Li, L.-J., 2016. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59, 2 (2016), 64–73. (cited on page 17)

Tucker, L. R., 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31, 3 (1966), 279–311. (cited on pages 30, 33, 63, and 100)

Turing, A. M., 1950. I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX, 236 (10 1950), 433–460. doi:10.1093/mind/LIX.236.433. https://doi.org/10.1093/mind/LIX.236.433. (cited on page 2)

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I., 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008. (cited on pages 25, 84, and 86)

Vinyals, O.; Toshev, A.; Bengio, S.; and Erhan, D., 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3156–3164. (cited on page 25)

Wang, P.; Wu, Q.; Shen, C.; Dick, A.; and Van Den Henge, A., 2017a. Explicit knowledge-based reasoning for visual question answering. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 1290–1296. AAAI Press. (cited on page 26)

Wang, P.; Wu, Q.; Shen, C.; and van den Hengel, A., 2017b. The vqa-machine: Learning how to use existing vision algorithms to answer new questions. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn*, vol. 4. (cited on page 61)

Wu, C.; Liu, J.; Wang, X.; and Dong, X., 2018. Chain of reasoning for visual question answering. In *Advances in Neural Information Processing Systems*, 275–285. (cited on page 74)

Wu, Q.; Wang, P.; Shen, C.; Dick, A.; and van den Hengel, A., 2016. Ask me anything: Free-form visual question answering based on knowledge from external sources. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4622–4630. (cited on pages 26 and 35)

Wu, Z. and Palmer, M., 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, 133–138. Association for Computational Linguistics. (cited on page 14)

Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K., 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1492–1500. (cited on pages 31, 32, 94, and 102)

Xiong, C.; Merity, S.; and Socher, R., 2016. Dynamic memory networks for visual and textual question answering. *arXiv*, 1603 (2016). (cited on pages 24 and 35)

Xu, D.; Zhu, Y.; Choy, C. B.; and Fei-Fei, L., 2017. Scene graph generation by iterative message passing. *arXiv preprint arXiv:1701.02426*, (2017). (cited on pages 17 and 27)

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y., 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, 2048–2057. (cited on pages 24, 29, and 94)

Yang, J.; Lu, J.; Lee, S.; Batra, D.; and Parikh, D., 2018. Graph r-cnn for scene graph generation. In *Proceedings of the European conference on computer vision (ECCV)*, 670–685. (cited on page 17)

Yang, Z.; He, X.; Gao, J.; Deng, L.; and Smola, A., 2016. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 21–29. (cited on pages 24, 29, 35, 43, 72, 73, and 94)

Yu, D.; Fu, J.; Mei, T.; and Rui, Y., 2017. Multi-level attention networks for visual question answering. In *Conf. on Computer Vision and Pattern Recognition*. (cited on pages 25, 35, 80, 84, 86, and 101)

Yu, Z.; Yu, J.; Cui, Y.; Tao, D.; and Tian, Q., 2019. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6281–6290. (cited on pages 25, 80, 81, 86, 87, 89, 119, and 120)

Yu, Z.; Yu, J.; Xiang, C.; Fan, J.; and Tao, D., 2018. Beyond bilinear: Generalized multimodal factorized high-order pooling for visual question answering. *IEEE Transactions on Neural Networks and Learning Systems*, (2018). (cited on pages 35, 36, 38, 50, 99, and 120)

Zhang, C.; Chao, W.-L.; and Xuan, D., 2019a. An empirical study on leveraging scene graphs for visual question answering. In *The British Machine Vision Conference (BMVC)*. (cited on pages 27, 82, and 89)

Zhang, J.; Elhoseiny, M.; Cohen, S.; Chang, W.; and Elgammal, A., 2017a. Relationship proposal networks. In *CVPR*, vol. 1, 2. (cited on page 27)

Zhang, J.; Kalantidis, Y.; Rohrbach, M.; Paluri, M.; Elgammal, A.; and Elhoseiny, M., 2019b. Large-scale visual relationship understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 9185–9194. (cited on page 82)

Zhang, P.; Goyal, Y.; Summers-Stay, D.; Batra, D.; and Parikh, D., 2016. Yin and yang: Balancing and answering binary visual questions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5014–5022. (cited on page 19)

Zhang, X.; Li, Z.; Change Loy, C.; and Lin, D., 2017b. Polynet: A pursuit of structural diversity in very deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 718–726. (cited on page 103)

Zhang, Y.; Hare, J.; and Prügel-Bennett, A., 2018. Learning to count objects in natural images for visual question answering. In *International Conference on Learning Representations*. (cited on pages 74, 89, and 120)

Zhou, B.; Tian, Y.; Sukhbaatar, S.; Szlam, A.; and Fergus, R., 2015. Simple baseline for visual question answering. *arXiv preprint arXiv:1512.02167*, (2015). (cited on pages 24 and 35)

Zhu, Y.; Groth, O.; Bernstein, M.; and Fei-Fei, L., 2016. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4995–5004. (cited on pages 24, 29, and 94)

Zhu, Y.; Lim, J. J.; and Fei-Fei, L., 2017. Knowledge acquisition for visual question answering via iterative querying. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1154–1163. (cited on page 24)