Learning Interpretable and Bias-Free Models for Visual Question Answering

Learning Interpretable and Bias-Free Models for Visual Question Answering

A THESIS PRESENTED
BY
GABRIEL J. GRAND
TO
THE DEPARTMENT OF COMPUTER SCIENCE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
BACHELOR OF ARTS
IN THE SUBJECT OF
COMPUTER SCIENCE & MIND, BRAIN, BEHAVIOR

HARVARD UNIVERSITY
CAMBRIDGE, MASSACHUSETTS

DECEMBER 2018

Thesis advisor: Professor Alexander Rush Gabriel J. Grand

Learning Interpretable and Bias-Free Models for Visual Question Answering

Abstract

Visual Question Answering (VQA) is an innovative test of artificial intelligence that challenges machines to answer human-generated questions about everyday images. Over the past several years, steadily ascending scores on VQA benchmarks have generated the impression of swift progress towards systems capable of human-level reasoning. However, closer inspection of current VQA datasets and models reveals serious methodological issues lurking behind the façade of progress. Many popular VQA datasets contain systematic language biases that enable models to cheat by answering questions "blindly" without considering visual context. Meanwhile, the predominant approach to VQA relies on black-box neural networks that render this kind of cheating hard to detect, and even harder to prevent.

In light of these issues, this work presents two sets of original research findings addressing the twin problems of interpretability and bias in VQA. We first aim to endow VQA models with the capacity to better explain their decisions by pointing to visual counterexamples. Our experiments suggest that VQA models overlook key semantic distinctions between visually-similar images, indicating an over-reliance on language biases. Motivated by this result, we introduce a technique called adversarial regularization that is designed to mitigate the effects of language bias on learning. We demonstrate that adversarial regularization makes VQA models more robust to latent biases in the data, and improves their ability to generalize to new domains. Drawing on our findings, we recommend a set of design principles for future VQA benchmarks to promote the development of interpretable and bias-free models.

Contents

For Laurie and Peter, who taught me to love learning.

Acknowledgments

This work would not have been possible without the generous support of my advisers, colleagues, friends, and family. I thank **Alexander Rush**, whose steadfast mentorship has pushed me to mature as a scientist over the past year. I would also like to thank **Aron Szanto** for many productive hours spent at the whiteboard, and **Yoon Kim** for contributing his expertise in the dark art of debugging neural networks. Aron, Yoon, and Sasha were instrumental in bringing Chapter 2 of this thesis from conception to publication in under six months. Thanks to the organizers of the **KDD 2018 Conference** for reviewing this research, and to **Clemens Mewald**, **Christine Robson**, and other members of the **Google Brain team** for making it financially and logistically possible for me to travel to London this past August to present there. I also owe a great deal to **Yonatan Belinkov**, who, soon after my return, introduced me to the idea of adversarial regularization, which forms the basis of Chapter 3 of this thesis. I thank Yonatan for entrusting me to extend his work on Natural Language Inference to the domain of VQA, and for the weekly check-ins that helped make this undertaking a success. Four years of thanks are owed to **Stuart Shieber** for helping me to integrate my twin passions for computation and cognition at Harvard, and for the many thought-provoking discussions about the Turing Test. I also thank the faculty of the **Department of Computer Science**, the **School of Engineering and Applied Sciences**, and the **Mind, Brain, and Behavior Program**. Special thanks to **Radcliffe Quad** and **Cabot Café** for keeping me cozy and caffeinated through the many late nights spent on this thesis. Finally, I would like to thank my parents, **Laurie Goodstein** and **Peter Grand**, my brother, **Ruben Grand**, and my partner in life, **Nora O'Neill**, for supporting my struggles and celebrating my successes throughout my time in college.

AI is a very hard problem so as a field we've separated out all of its pieces into separate fields (e.g. NLP, Computer Vision, Control, etc.) and we thought that we would solve all of them in isolation and then just plug them together. However, in the recent years the trends in research have convinced me that this is somewhat of a false view that will never come to fruition.

—Andrej Karpathy

Director of AI, Tesla

0

Introduction

The hardest problems in Artificial Intelligence are precisely those that ordinary people encounter every day. This phenomenon is termed "Moravec's Paradox," after the roboticist Hans Moravec, who observed that our sensorimotor skills are evolution's oldest and most refined cognitive tricks. As Moravec argued, "Encoded in the large, highly evolved sensory and motor portions of the human brain is a billion years of experience about the nature of the world and how to survive in it" (Moravec, 1988). Recognizing faces and emotions, identifying objects, interpreting actions, judg-

ing the intentions of others, using common sense, paying attention to interesting or relevant stimuli: all of these capabilities are deeply embedded in our evolutionary past, where they have remained out of scientific reach... until recently.

From an evolutionary perspective, progress in the field of AI has proceeded in a reverse-chronological march. Abstract reasoning, which Moravec terms a "new trick," was the first barrier to fall. In 1944, Popular Mechanics called the IBM Mark I a "superbrain" in reference to the machine's ability to manipulate numbers of up to 23-digits (Windsor, 1944). Fifty years later, IBM's DeepBlue bested the world chess champion, the New York Times heralded the victory as "a blow to the collective ego of humanity" (Weber, 1997). Each of these successive milestones has forced our understanding of intelligence to retreat from tasks involving high-level reasoning, and fall back towards those that involve routine, subconscious thinking.

With the rise of machine learning, however, efforts to reverse-engineer intelligence have recently begun to encroach into the territory of our everyday cognitive abilities. In computer vision (CV), object detection models recognize tennis rackets, mountains, and hot dogs (Caesar et al., 2016; Lin et al., 2014), while facial recognition models spot anger, disgust, and surprise (Fathallah et al., 2017; Bazrafkan et al., 2017). Similarly, in natural language processing (NLP), sentiment analysis models assess the attitudes of tweets and movie reviews (Agarwal & Mittal, 2016; Gautam & Yadav, 2014), while summarization models distill news articles into short snippets (Paulus et al., 2017), and even judge their factual accuracy (Oshikawa et al., 2018). Finally, in speech processing, a growing chorus of voice-powered mobile assistants offer directions, reminders, and weather updates (López et al., 2017; Hoy, 2018). Each of these systems carves out a piece of human intelligence and brings it firmly

under the control of algorithms.

Despite these impressive advances, contemporary AI systems are still confined to relatively narrow domains. Siri can offer directions, but it cannot look at a photo of a confusing street intersection in order to provide clarification. Similarly, Facebook's algorithms can recognize a friend's face, and can even tell that she is excited; however, they do not infer from context that the cause of her happiness is a recent job promotion. These examples illustrate that today's AI systems lack one of the hallmarks of human intelligence: the ability to seamlessly integrate information from multiple different sources (Lake et al., 2017). In order to make progress towards more general AI, we will need to consider tasks with significantly broader scope.

0.1 AI-Complete Problems

One approach to this endeavor is to focus on designing tasks that are "AI-complete." This term, which originated in the 1980s, is used as a stand-in for the ultimate—and perhaps unobtainable—goals of the field; a sort of holy grail of AI. The 1991 Jargon File (Raymond, 1991), a crowd-sourced glossary of the early hacker community, contained the following entry:

> **AI-complete: [MIT, Stanford, by analogy with "NP-complete"]**
>
> *adj.* Used to describe problems or subproblems in AI, to indicate that the solution presupposes a solution to the "strong AI problem" (that is, the synthesis of a human-level intelligence). A problem that is AI-complete is, in other words, just too hard.
>
> Examples of AI-complete problems are "The Vision Problem," building a system that can see as well as a human, and

> "The Natural Language Problem," building a system that can understand and speak a natural language as well as a human. These may appear to be modular, but all attempts so far (1991) to solve them have foundered on the amount of context information and "intelligence" they seem to require.

The Jargon File highlights the central importance of vision and language in the quest for AI-completeness. Conventional wisdom in the AI community has long held that these problems are each AI-complete on their own. However, with the advent of machine learning, many problems that were once perceived as intractable in both CV and NLP have been solved virtually overnight. Nevertheless, there remains a general sense in the community that the broader problem of intelligence remains unsolved. As Andrej Karpathy, a deep learning pioneer, observes (Karpathy, 2016):

> AI is a very hard problem so as a field we've separated out all of its pieces into separate fields (e.g. NLP, Computer Vision, Control, etc.) and we thought that we would solve all of them in isolation and then just plug them together. However, in the recent years the trends in research have convinced me that this is somewhat of a false view that will never come to fruition.

How should we proceed when problems that were once thought were "just too hard" suddenly become not hard enough? The answer, I believe, is to adopt a more holistic approach to AI. Rather than solve each piece of the puzzle separately, and then attempt to fit them together, our research efforts should dare to span multiple domains of intelligence. The present situation calls for audacious benchmarks that require models to achieve the kind of flexible, multimodal thinking that humans perform every day.

Within machine learning in the past several years, a small but growing body of research at the intersection of vision and language has started to bear fruit. These efforts began around 2014 with

the development of neural image captioning models capable of producing plausible descriptions of naturalistic scenes (Kiros et al., 2014; Xu et al., 2015; You et al., 2016). Since then, a small handful of other multimodal tasks have emerged, including caption-to-image generation (Mansimov et al., 2015), multimodal translation (Elliott et al., 2017), and visual question answering (Antol et al., 2015; Goyal et al., 2016; Wu et al., 2017; Gupta, 2017). While most of these other tasks have remained relatively niche, in recent years, visual question answering (VQA) has exploded in popularity. Within the field, VQA is regularly held up as a new standard for AI-completeness, with many researchers labeling it a "Visual Turing Test" (Malinowski & Fritz, 2014; Kafle & Kanan, 2017; Chao et al., 2017). In just a few pages, we will delve deeply into the contemporary state of VQA research, exploring the triumphs and missteps of this exciting research frontier. However, we will first take a brief historical detour to elucidate why, in the first place, the field of AI needs a new Turing Test.

0.2 The Turing Test & Jeopardy: Question Answering as an Intelligence Test

The tradition of question answering as a general test of intelligence traces back to the origins of the field of AI. In his seminal 1950 work, Alan Turing introduced the "imitation game," in which a computer attempts to convince a human interrogator that it is also a human, as a mechanism for studying machine intelligence. Though Turing placed few constraints on the content of this conversation, the example dialogues he provided all revolve around question-answering (Machinery, 1950):

```
Q: Please write me a sonnet on the subject of the Fourth
Bridge.
A: Count me out on this one. I could never write poetry.
Q: Add 34957 to 70764.
```

```
A: (Pause about 30 seconds and then give as answer) 105621.
Q: Do you play chess?
A: Yes.
Q: I have K at my K1, and no other pieces. You have only
K at K6 and R at R1. It is your move. What do you play?
A: (After a pause of 15 seconds) R-R8 mate.
```

The question of whether successful performance on the Turing Test constitutes proof of intelligence has been the subject of protracted debate in the AI community (Shieber, 2004). Scholars have raised various objections to the Turing Test and its incarnations in sponsored competitions like the Loebner Prize (Gunderson, 1964; Apter, 1970; Moor, 1976; Block, 1981; Shieber, 1994, 2004, 2006). Indeed, the literature on this subject is so rich that further discussion of the Turing Test here risks venturing into a Pandora's box.

Nevertheless, for our purposes, two shortcomings of the Turing Test are worth highlighting. First, as the above dialogue reveals, Turing's intuitions about the kinds of questions that an ideal interlocuter should ask fell prey to Moravec's Paradox. Adding numbers and playing chess are both examples of high-level symbolic reasoning tasks that are easily solved algorithmically. Meanwhile, while sonnet-writing has not yet been convincingly "solved," most ordinary people would also be hard-pressed to perform this task.[1] Prescient as he was about many aspects of modern computing, Turing would likely be surprised to see that so many of the remaining unsolved problems in AI center around everyday, as opposed to intellectual, thinking.

It is also instructive to consider a second deficiency in the structure of the Turing Test; namely,

[1]Indeed, in Turing's example, the subject passes on this question—perhaps a clever computer might similarly elect to hide its knowledge of Shakespeare in order to avoid rousing suspicion.

its reliance on verbal behavior as the sole window into intelligence. Turing's conception of the imitation game was likely shaped by technological circumstances. In Turing's time, human-computer interaction consisted of feeding input into the machine via tape or punch cards, and receiving a printed readout. In this context, the idea of using images as inputs to a computer would likely have seemed far-fetched. Indeed, when Turing published "On Computing Machinery and Intelligence," the first digital camera, which ushered in the era of digital photography, was still nearly forty years away (Lloyd & Sasson, 1978). Thus, it is not surprising that Turing based his imitation game on a modality that was consistent with the technology of his time. Nevertheless, the legacy of the Turing Test has proved enduring, and its reliance on linguistic interactions has in turn constrained the modern imagination with respect to the design of tests of machine intelligence.

The shortcomings of the Turing Test outlined above suggest two desiderata for a general test of machine intelligence.

Criteria for a new Turing Test:

1. The test should involve a broad cross-section of everyday reasoning. Accordingly, it should not hinge on behaviors typically associated with high levels of intellect (e.g., performing complex computations, playing chess, etc.).
2. The test should require integrating multimodal information from perceptual, linguistic, and other sources in the service of this reasoning.

One might contend that the lack of these prerequisites is a particular property of the Turing Test and its historical context. However, these same two misconceptions about what constitutes a "hard problem" for AI have continued to drive research long after Turing's time. When IBM Watson de-

feated *Jeopardy!* champions Ken Jennings and Brad Rutter in 2011, the New York Times heralded this development as "a big step toward a world in which intelligent machines will understand and respond to humans" (Markoff, 2011). Though *Jeopardy!* is by no means a trivial problem, it falls on the wrong side of both criteria outlined above. First, *Jeopardy!* calls for specific, arcane knowledge; like sonnet-writing, the average person is not likely to perform particularly well. Second, *Jeopardy!* is, for all intents and purposes, a unimodal task; Watson reads in clues as text, and compares them to a large stored database of information, also represented as text (Ferrucci et al., 2010). As IBM's Director of Research John Kelly III observed following Watson's *Jeopardy!* victory (Kelly III & Hamm, 2013),

> The *Jeopardy!* challenge was relatively limited in scope. It was bound by the rules of the game and the fact that all the information Watson required could be expressed as words on a page. In the future, Watson will take on more open-ended problems. It will ultimately be able to interpret images, numbers, voices, and sensory information.

It is no coincidence that Kelly's vision for the future of the Watson program pointed to open-ended, multimodal problems as the future waypoints of AI. Unfortunately, the *Jeopardy!* victory represented somewhat of a high water mark that IBM has since been unable to overcome. Indeed, over half a decade later, we are still struggling with the basic questions that govern this new research domain. These questions include: What constitutes a good multimodal task? Which modalities should be included in the task? And what role, if any, should question answering play in the design of such a task?

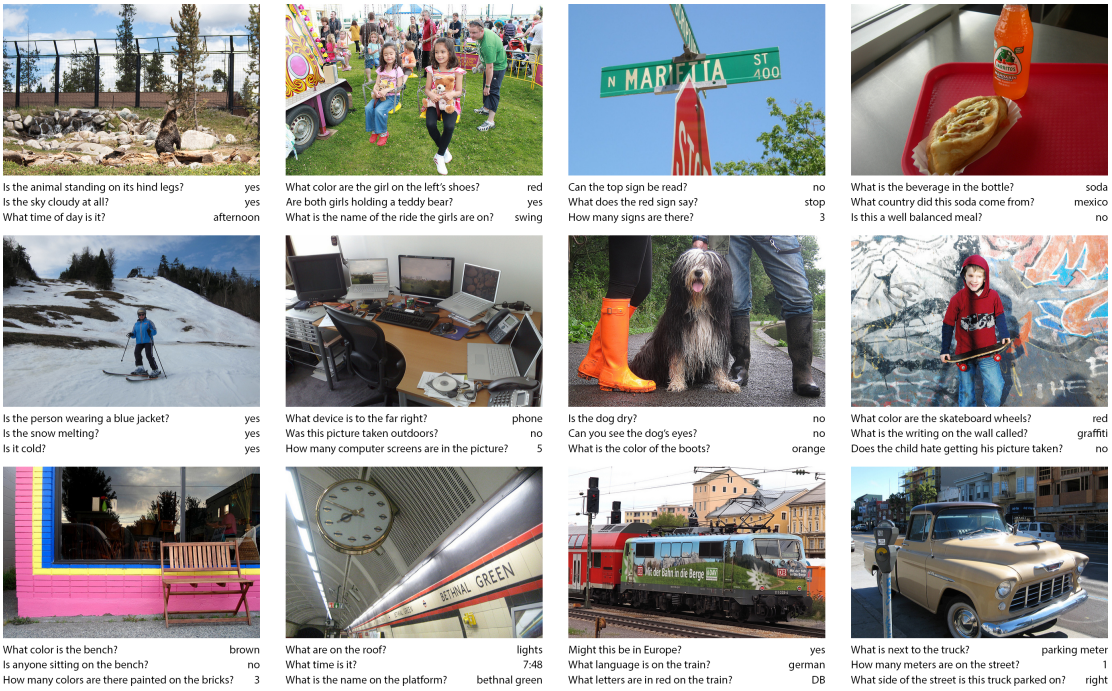


Figure 1: Examples from the VQA Challenge v1 dataset (Antol et al., 2015).

0.3 Introduction to Visual Question Answering

Visual Question Answering (VQA) offers one approach to operationalizing the vision of a holistic test of machine intelligence. Simply put, VQA challenges models to answer natural language questions about images. VQA satisfies both of the desiderata of a machine intelligence test introduced above. The VQA datasets used in this work represent a diverse range of everyday scenarios. The data collection methods employed for both questions and images make extensive use of internet crowdsourcing to elicit a variety of commonplace contexts (Lin et al., 2014; Antol et al., 2015). VQA is also deeply multimodal: successful performance is intended to require skillful integration of visual and linguistic information. In addition to this information, many VQA questions also require incorpo-

rating broad-based world knowledge; i.e., "common sense." Finally, in keeping with longstanding traditions in AI since Alan Turing, VQA leverages question-answering as a vehicle for examining intelligence. Thus, in principle, VQA is well-positioned to serve as a catalyst for many exciting avenues of AI research.

Yet, like many endeavors that hold much promise in the abstract, the current state of VQA research is plagued by several issues. These problems are endemic not just to today's popular VQA datasets, but also to the kinds of models that researchers apply to them. In particular, they arise from the application of powerful black-box machine learning methods to large, crowdsourced datasets. In this work, I attempt to delineate, investigate, and ultimately, propose solutions to what I see as two of the main dilemmas of contemporary VQA research. I call these "The Problem of Interpretability" and "The Problem of Bias."

0.3.1 The Problem of Interpretability

Issues of interpretability are not specific to VQA; indeed, they affect the majority of today's machine learning systems. Here, we adopt the definition of interpretability from Doshi-Velez & Kim (2017) as "the ability to explain or to present in understandable terms to a human." Under this criterion, a wide class of present-day machine learning models are not interpretable. In particular, in deep neural networks, reasoning is distributed across millions of parameters. Consequently, it is notoriously difficult for humans—who are generally accustomed to causal forms of reasoning—to understand the outputs of deep learning models (Chakraborty et al., 2017). As ML systems increasingly assume responsibility in many areas of society, from criminal justice to transportation, there is a growing

need for these systems to be interpretable by humans. Indeed, under the European General Data Protection Regulation, which went into effect in 2018, individuals in the European Union are entitled to a "right to explanation" from any algorithm that makes decisions based on personal data (Parliament & of the European Union, 2016). As evidenced by the recent DARPA Explainable AI initiative (Gunning, 2017), interpretability has become one of the hot topics in AI, and serves as a major catalyst for ongoing research.

In the realm of VQA, the need for interpretability is motivated by both pragmatic and more philosophical considerations. For many years, researchers have touted the potential role of VQA models in assisting the visually-impaired (Lasecki et al., 2014; Antol et al., 2015; Gurari et al., 2018). In order to gain the trust of their users, these systems must be able to account for their decisions (Goyal et al., 2016). Beyond the potential applications for VQA, however, there is a more fundamental argument for interpretability: understanding the underlying process by which VQA models arrive at their decisions is an important means of ensuring that these models actually learn the kinds of knowledge that we would like them to learn. As Doshi-Velez & Kim (2017) note, interpretability is often a prerequisite for confirming other important desiderata of ML systems. In particular, interpretability is useful for ascertaining whether a system is robust to biases that may exist in its training data. In VQA, researchers have begun to discover that this is not the case, which brings us to our second dilemma.

0.3.2 The Problem of Bias

Like interpretability, the Problem of Bias affects many domains in AI. In machine learning, two semi-distinct notions of bias are at play: *social bias* and *statistical bias*. In their new textbook on ML fairness, Barocas et al. (2018) define social biases as "demographic disparities in algorithmic systems that are objectionable for societal reasons." The effects of social bias in ML range from the acute to the understated; from systematic racial discrimination in criminal sentencing algorithms (Angwin et al., 2016; Chouldechova, 2017; Berk et al., 2017) to misrecognition of minority and non-male faces in vision algorithms (Buolamwini, 2017). These forms of social bias are dangerous because they have the potential to perpetuate existing societal injustices. Nevertheless, in many cases, models that exhibit social biases may actually be well-fit to the data (Barocas et al., 2018). In contrast, in statistics, a biased model is one that consistently over- or under- estimates the true value of its target in comparison to the empirical distribution (Larson & Larson, 1969). Thus, it is possible to identify and correct statistical bias purely by improving a model's fit to the existing data. On the other hand, diagnosing and addressing social bias often requires examining the data itself to look for trends that clash with social norms and values.

Recently, the term "bias" has begun to appear in the VQA literature in a way that highlights the complex and ambiguous relationship between notions of social and statistical bias. The core issue is that VQA datasets tend to contain superficial regularities that allow models to "cheat" by memorizing relationships between question and answer words. For instance, in one popular VQA dataset, for questions of the form, "What sport is...?", the correct answer is "tennis" 41% of the time (Zhang

et al., 2016). In this example, the source of the bias is the underlying Common Objects in Context (COCO) image dataset (Lin et al., 2014), which happens to contain a large number of tennis images. Another source of bias occurs in the question-generation process, which involves human subjects whose behavior may be unintentionally influenced by subtle psychological factors. One such factor is the phenomenon of visual priming, which selects for questions with affirmative answers. For instance, in the same dataset, for questions beginning with "Do you see a...?" the correct answer is "yes" 87% of the time (Zhang et al., 2016). A growing body of work demonstrates how, by exploiting these language biases, models can disregard the image and still achieve high performance on VQA (Agrawal et al., 2016; Zhang et al., 2016; Jabri et al., 2016; Goyal et al., 2016; Chao et al., 2017; Johnson et al., 2017; Agrawal et al., 2018; Thomason et al., 2018).

On the surface, the Problem of Bias does not match the kinds of AI bias that we are accustomed to hearing about in the media. The existence of a reliable mapping from question words to answer words in VQA data is a general phenomenon, and does not appear to encode prejudices against any particular group. However, a closer examination reveals that the Problem of Bias in VQA is an instance of social, not statistical, bias. Consider that the contents of the images in a VQA dataset may encapsulate subtle social biases. For instance, while the over-representation of "tennis" images in the COCO dataset may not necessarily call to mind a specific gender bias, the labeled object "tennis racket" is much more likely to appear in images containing men. In fact, as Zhao et al. (2017) reveal, the majority of the objects in COCO reflect gender biases (e.g., "knife," "fork," and "spoon" co-occur more frequently with women, while "snowboard" and "boat" appear more frequently with men). Similarly, researchers in NLP have noted that word embeddings derived from text corpora

contain many recognizable social biases (Bolukbasi et al., 2016; Caliskan et al., 2017; Grand et al., 2018a). Consequently, a VQA model that learns to predict answers "blindly" on language priors is more likely to make socially-biased decisions than a VQA model that also gives consideration to the specific visual context.

The social nature of the biases in VQA makes a solution significantly harder to come by. If the issue were one of statistical bias, then we could simply apply one of many well-established bias correction methods from statistical inference. However, in VQA, the issue is not that we lack models that fit the data well—indeed, it is precisely the opposite: existing models *overfit* to the available information in the training data. Given the well-known trade-off between bias and variance in statistics (Goodfellow et al., 2016), an ideal solution to the Problem of Bias actually involves *increasing* the bias of the model in order to reduce the variance in performance from one dataset to another. However, in order to successfully perform this trade-off, we will need some way of distinguishing the kinds of patterns that we would like to learn from those that we wish to avoid generalizing from one context to another. This is exactly the same problem that researchers face when trying to reduce learned social biases in ML models. As Barocas et al. (2018) describe,

> Some patterns in the training data (smoking is associated with cancer) represent knowledge that we wish to mine using machine learning, while other patterns (girls like pink and boys like blue) represent stereotypes that we might wish to avoid learning. But learning algorithms have no general way to distinguish between these two types of patterns, because they are the result of social norms and moral judgments. Absent specific intervention, machine learning will extract stereotypes, including incorrect and harmful ones, in the same way that it extracts knowledge.

In this light, the Problem of Bias in VQA reduces to the problem of preventing a model from learn-

ing stereotypes and other harmful social biases. As a result, research in this area has the potential to yield new methods that are generally applicable to a variety of scenarios in which we seek to mitigate social biases in ML models.

0.3.3 The Road Ahead

This work examines the current state of VQA research through the dual lenses of interpretability and bias. It presents findings from two sets of original research that address these respective issues.

Chapter 1 provides a background on contemporary methods in VQA aimed at helping the reader quickly get up-to-speed on the relevant datasets, models, and techniques.

Chapter 2 examines the Problem of Interpretability by seeking to endow VQA models with the capacity to better explain their decisions. Specifically, we reformulate VQA as a counterexample prediction task that challenges models to identify pairs of similar images that produce *different* answers to a particular question. Although our methods improve the state-of-the-art on this task, we find that the representations learned by a top-performing VQA architecture do not appear to capture key semantic distinctions between visually-similar images. This result, which is indicates that VQA models lack visual grounding, provides motivation for Chapter 3.

Chapter 3 applies a recently-developed adversarial regularization method from the NLP literature to address the Problem of Bias in VQA. We demonstrate that this method successfully reduces dependence on language priors, and boosts performance on unseen domains with different priors. Moreover, we show that the benefits of adversarial regularization are proportional to the amount

of bias in the training data. These results offer promising support for adversarial regularization as a general purpose method for building ML models that are robust to learned biases.

Chapter 4 draws on our findings to put forward a set of desiderata for future VQA benchmarks that promote development of interpretable and bias-free models. It concludes with a presentation of several new datasets that promise to bring VQA into the real world, and highlights the need for continued attention to the issues of interpretability and bias in these efforts.

Algorithms force us to look into a mirror on society as it is.

—Sandra Wachter

Professor, Oxford Internet Institute

1

Visual Question Answering: Datasets and Models

Contemporary research interest in VQA for everday images began with the release of DAQUAR, the DAtaset for QUestion Answering on Real-world images (Malinowski & Fritz, 2014). Since then, at least five other naturalistic VQA benchmarks have been proposed. These include COCO-VQA (Ren et al., 2015a), FM-IQA (Gao et al., 2015), VisualGenome (Krishna et al., 2017),

Visual7w (Zhu et al., 2016), and the VQA Challenge datasets (Antol et al., 2015; Goyal et al., 2016). With the exception of DAQUAR, all of the datasets use include images from the Common Objects in Context (COCO) dataset (Lin et al., 2014), which contains 330K images sourced from Flickr.

1.1 VQA v1

The eponymous VQA Challenge dataset (hereafter, just "VQA") was first introduced in Antol et al. (2015) as more free-form, open-ended VQA benchmark. Previous datasets placed restrictions on the kinds of questions authored by human annotators (e.g., Visual7w, VisualGenome), or relied on image captioning models to generate questions (e.g., COCO-VQA). In contrast, the crowdsourcing method employed by Antol et al. (2015) was designed generate a more diverse range of question types requiring both visual reasoning and common knowledge. However, owing in part to the lack of constraints on question generation, the original VQA dataset contains several conspicuous biases. As discussed in Section 0.3.2, for questions beginning with the phrase, "What sport is...", the correct answer is "tennis" 41% of the time. Additionally, question generation was impacted by a visual priming bias, which selected for questions with affirmative answers. For instance, for questions beginning with "Do you see a...," the correct answer is "yes" 87% of the time (Zhang et al., 2016). Models that exploit these biases can achieve high accuracy on VQA without understanding the content of the accompanying images (Agrawal et al., 2016; Zhang et al., 2016; Jabri et al., 2016; Goyal et al., 2016; Chao et al., 2017; Johnson et al., 2017; Agrawal et al., 2018; Thomason et al., 2018).

Figure 1.1: Examples from VQA v2. To increase the heterogeneity of answers, the dataset includes pairs of complementary images that produce different answers to the same question. (Figure from Goyal et al., 2016)

1.2 VQA v2

In an effort to balance the VQA dataset, Goyal et al. (2016) introduced VQA v2, which is built on pairs of visually-similar images that result in different answers to the same question. Specifically, for each image in the original dataset, Goyal et al. (2016) determined the 24 nearest neighbor images using convolutional image features derived from VGGNet (Simonyan & Zisserman, 2014a). For each image/question/answer pair in the original VQA dataset, crowd workers were asked to select a complementary image that produced a *different* answer to the same question. The most commonly selected complementary image was then paired with the original question and new answer. These data were included as new examples in VQA v2, resulting in a dataset that is roughly double the size of the original. The inclusion of complementary pairs data in VQA v2 also makes it possible to more

directly examine how models represent distinctions between visually-similar images. We explore this idea further in Chapter 2.

By several metrics, VQA v2 succeeds at reducing biases in the answer distribution. Goyal et al. 2016 note that the entropy of the answer distributions averaged across various question types (and weighted by frequency) increases by 56% from VQA v1 to VQA v2. Indeed, in Figure 1.2, we can see that answers over many question types are more evenly distributed in VQA v2. Additionally, questions with binary (yes/no) answers exhibit an answer distribution that is closer to 50/50. Goyal et al. (2016) found that across the board, state-of-the-art VQA architectures for VQA v1 perform worse on VQA v2. This finding provides strong evidence for the theory that these models succeed primarily by exploiting answer biases. Finally, VQA models designed to ignore answer biases demonstrate less of a performance discrepancy between VQA v1 and VQA v2, suggesting that VQA v2 contains fewer biases for models to exploit (Agrawal et al., 2018; Ramakrishnan et al., 2018).

While VQA v2 represents progress towards the goal of reducing latent biases, several sources of bias still remain. Although the introduction of complementary pairs results in a near 50/50 balance for yes/no questions, most questions in the VQA datasets are not binary. For questions with many possible answers (e.g., "What type...?"; "What sport...?"; "What brand...?"), VQA v2 method shifts some probability mass into the tail, but a handful of top answer choices continue to dominate (see Figure 1.2).

The persistence of answer-class biases in VQA v2 means that models that exploit these biases can continue to enjoy dominant performance on this task. Indeed, even after the organizers of the annual VQA Challenge switched to VQA v2 in 2017, many of the high-performing architectures

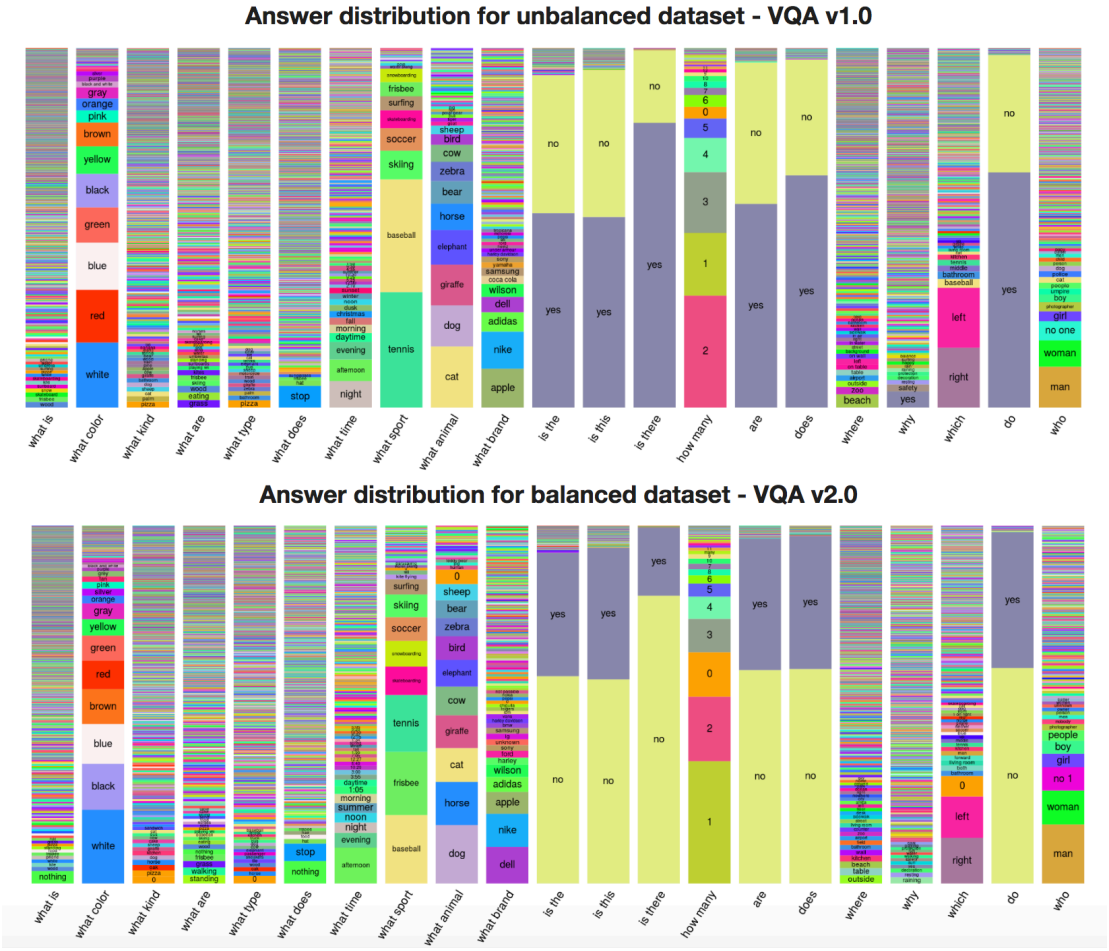


Figure 1.2: Comparison of answer distributions for different question types in VQA v1 and v2. The inclusion of complementary pairs in VQA v2 results in visibly less skewed distributions. This effect is most noticeable for binary questions (e.g., "is the", "is this", "is there", "are", "does"). (Figure from Goyal et al., 2016)

from the previous year continued to top the leaderboard.[1] Broadly speaking, the switch from VQA v1 to VQA v2 did not prompt competitors to alter their methods or models to address the issue of bias. In a paper titled "Tips and Tricks for Visual Question Answering: Learnings from the 2017 Challenge," the 2017 contest winners detailed the many methods they employed to attain top performance (Teney et al., 2017a). These included employing gated tanh activation functions, increasing the minibatch size, using pretrained output embeddings, and obtaining image features from a more powerful convolutional neural network. Crucially, the vast majority of these improvements were orthogonal to the changes in the dataset composition. The only methodological change directly motivated by VQA v2 was the inclusion of the complementary pairs data in the same minibatch. While the authors argued that this "smart shuffling" procedure stabilized training, in an ablation study, they found that it actually led to a slight *reduction* in overall score (Teney et al. 2017a, Table 1). Thus, in theory, VQA v2 succeeded in bringing attention to the issue of bias in VQA. However, in practice, the introduction of the dataset failed to spur corresponding solutions to these issues within the VQA research community.

1.3 VQA-CP

The Problem of Bias in VQA poses a philosophical dilemma. On the one hand, from the perspective of designing a naturalistic question answering task, the existence of a few very likely answers is not inherently a problem. Indeed, certain answers are simply *a priori* unlikely. For instance, it would be suboptimal for a question answering system grounded in world knowledge to believe that the sport

[1] VQA Challenge 2017 leaderboard at http://visualqa.org/roe_2017.html.

of broomball is just as common as baseball. In this light, it is neither realistic nor desirable for a real-world VQA dataset to contain uniform distributions over answer classes. On the other hand, the fact that models can attain top scores on VQA benchmarks by simply learning language biases defeats the purpose of *visual* question answering. Moreover, the resulting inflation in accuracy scores on VQA threatens to give false impressions about the capabilities of current VQA systems.

Concerns over these issues within the community led to the introduction of a new variant of the existing VQA datasets, based on the following premise: What if, instead of reducing the bias in the answer distribution, we simply *varied* them so as to discourage overfitting? This idea is the basis for Visual Question Answering under Changing Priors (VQA-CP; Agrawal et al. 2018). In VQA-CP, for each question type, the prior distribution over answers is designed to differ significantly between the train and test splits. For instance, the most frequent sport in the train split is "tennis," while in the test split it is "skiing." Figure 1.3 shows how the answer distribution changes between train and test splits in VQA-CP. Note that VQA-CP is constructed by redoing the train/test splits for the existing VQA v1 and v2 data; not by collecting additional data. For this reason, there are two versions of VQA-CP (i.e., VQA-CP v1 and VQA-CP v2). Moreover, because they lack the complementary pairs data, VQA v1 and VQA-CP v1 both contain more language biases than their v2 counterparts (Agrawal et al., 2018). In Chapter 3, we will show that regularization methods designed to counter language biases show larger gains on the more biased VQA-CP v1 dataset.

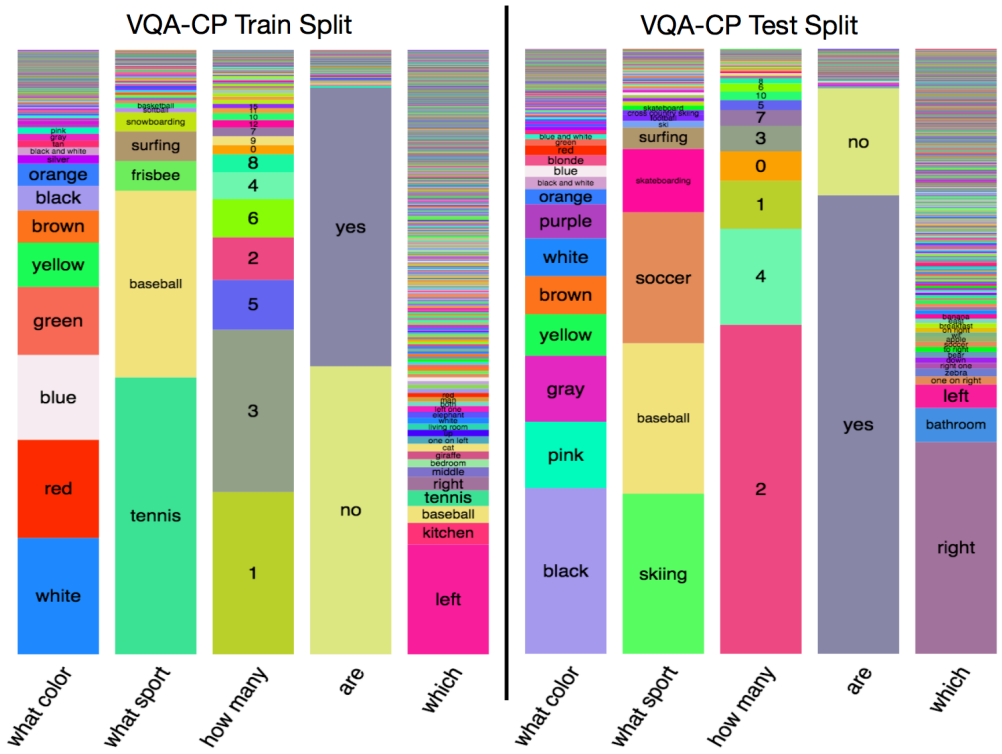


Figure 1.3: Distribution of answers per question type for a random sample of 60K examples from VQA-CP v1. The distribution of answers varies significantly between the train and test splits. (Figure from Agrawal et al., 2018)

1.4 Problem Formalization

Before proceeding further, it is useful to formalize the VQA problem and lay out the terminology that will be used throughout this work. A complete glossary of notation is given in the Appendix.

The principal objects in VQA are images, questions, and answers. We can refer to a VQA dataset as a set of examples $D = \{d_1, \ldots, d_N\}$, where each example is a tuple $d_i = (I_i, Q_i, \mathcal{A}_i)$. As is standard practice in computer vision, images $I \in (h, w, c)$ are represented as tensors of fixed height,

width, and RGB color channels. Similarly, as is standard practice in natural language processing, questions are represented as sequences of tokens $Q = \{q_1, \ldots, q_M\}$, where each $q_m \in \{0, 1\}^{|\mathcal{V}|}$ is a one-hot vector representing the index of the token in a fixed-size vocabulary $\mathcal{V}$. In VQA, it is common to produce vectorized feature representations of images and questions using existing vision and language models. These representations are usually produced either early in the VQA pipeline, or as a separate preprocessing step entirely. Thus, when we refer to images and questions, we are usually dealing with the vectorized versions, which we denote with lowercase v and q.

1.5 VQA Architectures: A Framework

Many current VQA models follow the same abstract architectural pattern. In this section, we cover the common components of these models.

1.5.1 Image Encoding

In order to reason about the contents of an image, it is necessary to first extract some higher-level feature representation from the raw pixel data. Convolutional neural networks (CNNs), which are widely used in contemporary computer vision, provide a powerful and modular solution for feature extraction. While CNNs are typically trained on image classification or object recognition tasks, the feature representations they learn are transferable out-of-box to a wide range of domains. For some applications, finetuning the weights of a pretrained CNN has been shown to improve transfer (Yosinski et al., 2014). However, in VQA, finetuning is not a common practice; since the images in the VQA datasets come from COCO (Lin et al., 2014), a widely-used computer vision benchmark, it

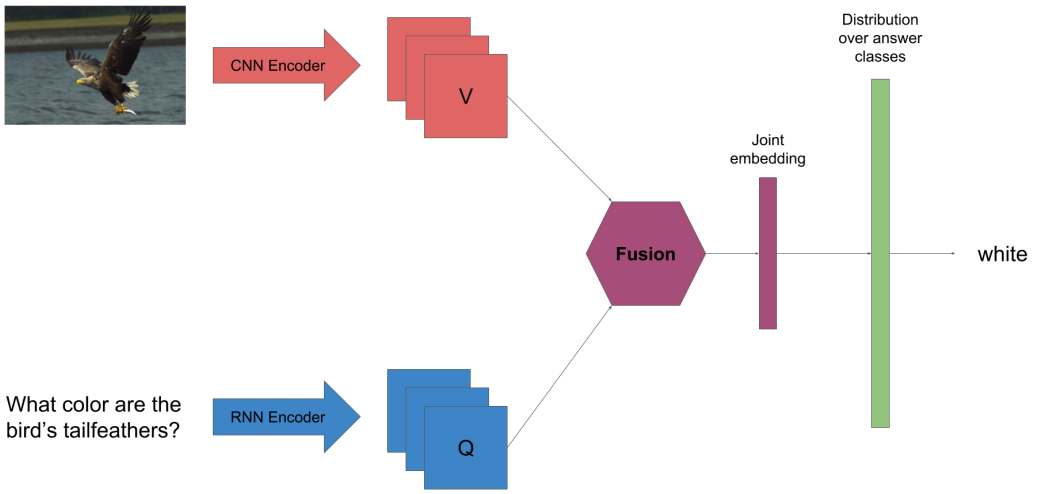


Figure 1.4: Schematic diagram of a common VQA architectural pattern. Images are encoded by a CNN, while questions are encoded by an RNN. These representations are combined via a multimodal fusion scheme into a joint embedding. Finally, a fully-connected network predicts a distribution over answer classes.

is easy to obtain CNNs that have already been pretrained on this data.

There are many popular off-the-shelf CNN architectures that are commonly applied to VQA, including (from oldest to newest): VGG (Simonyan & Zisserman, 2014b), Inception (Szegedy et al., 2015), ResNet (He et al., 2015b), Faster R-CNN (Ren et al., 2015b), and Mask R-CNN (He et al., 2017). Over the past several years, CNNs have become both more powerful and more efficient. Moreover, in addition to producing feature representations, models like Faster R-CNN and Mask R-CNN output object bounding boxes, which can be used to obtain performance benefits on VQA. Consequently, the choice of CNN can have a significant impact on the performance of a VQA model. For this reason, one should note the underlying CNN architecture when comparing

performance across different VQA architectures. In this thesis, direct performance comparisons are generally made between variants of the same model that, by definition, use the same CNN.

1.5.2 Question Encoding

As with images, it is necessary to extract the contents of the question into a machine-readable format that encodes its semantics. Unlike an image, however, a question is sequential in nature, and its meaning depends on the relative ordering of its component words. In natural language processing, recurrent neural networks (RNNs) have emerged as the method of choice for dealing with linguistic data. In particular, a class of RNNs called long short-term memory networks (LSTMs; Hochreiter & Schmidhuber 1997), which are adept at learning long-term dependencies in sequences, is now standard-issue for many NLP tasks, including language modeling (Peters et al., 2018), machine translation (Sutskever et al., 2014; Bahdanau et al., 2014), and summarization (Nallapati et al., 2016; Kryściński et al., 2018).

Unlike CNNs, of which there are many different architectural variations, RNN implementations for sequence modeling are fairly standardized. Only a few iterations on these models have been proposed. Most notably, the gated recurrent unit (GRU; Chung et al. 2014) has emerged as a more efficient alternative to the LSTM cell. Nevertheless, performance performance between LSTMs and GRUs is comparable across many tasks (Chung et al., 2014). For the purposes of VQA, therefore, the choice of RNN for question encoding is less relevant to performance than the choice of CNN for image encoding.

1.5.3 Multimodal Fusion

In order to perform VQA optimally, a model must integrate both visual and linguistic data. Given that the image and question features are typically encoded using separate pipelines, the integration of these representations is a key methodological concern in VQA. Abstractly, we can represent multimodal fusion as a function $z = f(I, Q)$ mapping the image and question to a multimodal representation z. Early approaches combined the vectors I and Q via an elementwise sum $f(I, Q) = I + Q$ or product $f(I, Q) = I \odot Q$, or alternatively, by simple concatenation; i.e., $f(I, Q) = [I, Q]$ (Zhou et al., 2015; Lu et al., 2015; Kim et al., 2016a; Antol et al., 2015). While these naive approaches have been shown to work surprisingly well, subsequent work has explored more expressive fusion schemes through parametrized bilinear interaction; i.e., $f(I, Q) = W[I \otimes Q]$, where $\otimes$ denotes the outer product (Fukui et al., 2016; Kim et al., 2016b; Ben-younes et al., 2017; Duke & Taylor, 2018). Finally, a separate line of research has used multi-step attention mechanisms as a means for fusing image and question representations (Yang et al., 2016; Lu et al., 2016; Anderson et al., 2018). In many cases, these novel fusion mechanisms have been shown to result in significant performance gains. Thus, the design of mechanisms for multimodal fusion is an active area of ongoing research.

1.5.4 Answer Prediction

Given a multimodal representation z, the end goal of any VQA model is to predict answers as outputs. The final module of many VQA models is a classifier network consisting of one or more fully-connected layers, which maps z to a distribution $P(\mathcal{A}|I, Q)$ over possible answers in a fixed answer

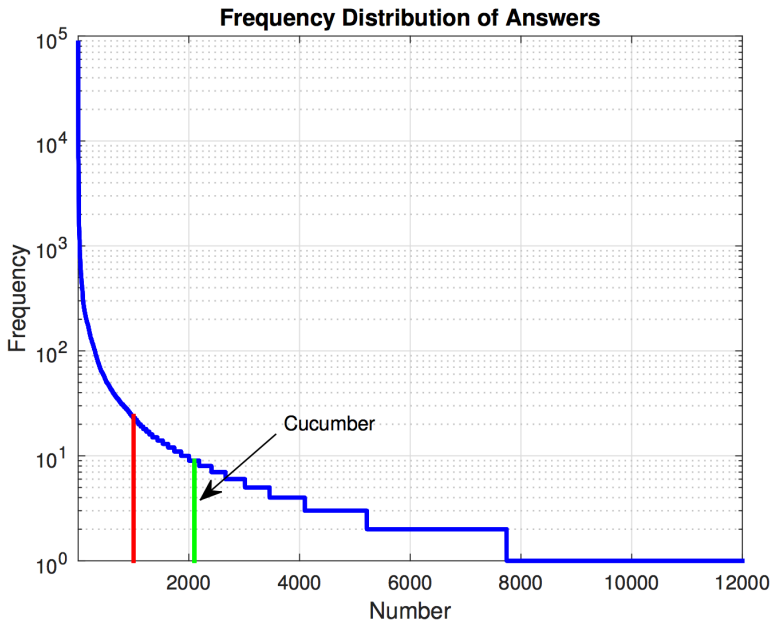


Figure 1.5: Frequency distribution over answers $\mathcal{A}$ in VQA v1. Models typically truncate $\mathcal{A}$ to the 1000 (red line) or more most common answers. This approach ignores the majority of answers, such as "cucumber," that fall in the long tail. (Figure from Ma et al., 2017)

vocabulary $\mathcal{A}$. Because answers in VQA are open-ended and may consist of multiple words, the most natural approach to this task would treat it as a sequence generation problem akin to image captioning (Xu et al., 2015; You et al., 2016). However, as discussed, the distribution over answers has a very long tail, meaning that many answers will only occur a handful of times in the dataset. Consequently, the vast majority of competitive VQA models truncate $\mathcal{A}$ to the top few thousand answers, and treat VQA as a large multiclass classification problem (see Wu et al. 2017 for a breakdown of different VQA approaches, including classification vs. generation).

By formulating VQA as a classification problem, these models sacrifice accuracy on the long tail of answers in exchange for higher performance on the most common answers. Given the over-

whelming prevalence of this approach, we treat this design decision as a fact of the current state of the VQA research space. Ultimately, however, the fact that the existing VQA datasets promote this kind of hack could be considered a serious shortcoming. A potential remedy to this problem is discussed in Chapter 4.

1.6 Selected VQA Models

The experiments presented in this thesis were performed by extending two existing VQA model implementations. The experiments presented in Chapter 2 were based on a model from a paper titled Multimodal Tucker Fusion for Visual Question Answering (MUTAN, Ben-younes et al. 2017). Meanwhile, the experiments presented in Chapter 3 were based on a more recent implementation of a model from Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering (Bottom-Up/Top-Down; Anderson et al. 2018). In the next two sections, we cover each of these models in more depth.

The switch from MUTAN to Bottom-Up/Top-Down was motivated by performance and engineering considerations that reflect the fast pace of research progress in VQA. In March 2018, when the experiments for Chapter 2 were performed, MUTAN's reported 58.2% overall accuracy on VQA v2 was on par with the state-of-the-art (Ben-younes et al., 2017). However, at the Conference on Computer Vision and Pattern Recognition (CVPR) in June 2018, it was announced that a team at Facebook AI Research had won the VQA 2018 Challenge (Jiang et al., 2018b) with 70.01% single-model accuracy. The following month, the group released Pythia, a PyTorch-based VQA framework

based on the Bottom-Up/Top-Down model (Jiang et al., 2018a). In addition to achieving significantly higher performance, the Pythia code is also more efficient. (We found that Pythia takes less than 2 hours to train to peak accuracy on a Tesla V100 GPU, compared with approx. 8 hours for MUTAN). Since rapid testing cycles are crucial for maintaining research velocity, we decided to switch from MUTAN to Pythia for the experiments in Chapter 3. Thus, while the findings from Chapters 2 and 3 are intended to be considered in a mutual context as part of this broader narrative of this thesis, any comparison of absolute accuracy numbers between these two chapters should be taken in light of the difference in baseline models.

In addition to achieving different levels of performance, MUTAN and Bottom-Up/Top-Down differ architecturally. While both models adhere to the general framework outlined in Section 1.5, they differ in the choice of fusion mechanism. In particular, MUTAN uses a parametrized bilinear interaction for multimodal fusion, while Bottom-Up/Top-Down relies on attention to perform the fusion. Indeed, these two models happen to be representative of the two schools of thought on fusion mechanisms presented in Section 1.5.3. In the discussion of these two models in the following sections, special focus is given to the role of the fusion mechanism as a defining aspect of the respective VQA architectures.

1.6.1 MUTAN

MUTAN (Ben-younes et al., 2017) uses a ResNet-152 CNN to encode images and a GRU-based SkipThoughts RNN (Kiros et al., 2015) to encode questions. The core of MUTAN is a multimodal fusion scheme based on the Tucker decomposition (Tucker, 1966). The goal of this setup is to model

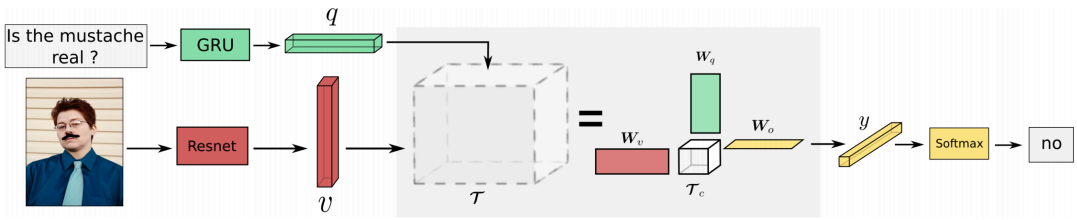


Figure 1.6: MUTAN model architecture. The bilinear interaction T is factorized via Tucker decomposition into a smaller core tensor T_c and three matrices, dramatically reducing the number of parameters. (Figure from Ben-younes et al., 2017)

a bilinear interaction between the image and question features:

$$z = T \times_1 Q \times_2 I \tag{1.1}$$

Here, $T \in \mathcal{R}^{|Q| \times |I| \times |A|}$ is a 3D tensor, and $\times_i$ is the i-mode tensor product, which involves slicing along the ith dimension of T. This bilinear approach is powerful because, unlike elementwise multiplication, it allows a multiplicative interaction between all elements of both Q and I (Fukui et al., 2016). However, it suffers from significant dimensionality issues, since the size of T grows exponentially with the size of Q and I. In practice, Ben-younes et al. (2017) use $|Q| = 2400$, $|I| = 2048$, and $|A| = 2000$. To use the full T in this scenario would introduce some 9.83 billion parameters into the model. This number of parameters is many orders of magnitude larger than today's biggest neural networks.[2] For this reason, it is impractical to explicitly model a full bilinear interaction between Q and I in the network.

To overcome this obstacle, various fusion mechanisms have been proposed that approximate a

[2] ResNet-152, for instance, contains on the order of 60 million parameters (He et al., 2015a; Chandrasekhar et al., 2017)

bilinear interaction, but with a significantly reduced number of parameters. Fukui et al. (2016) introduced a popular Multimodal Compact Bilinear (MCB) pooling model, which randomly projects Q and I into a higher-dimensional embedding space, and then convolves both vectors in the Fourier space. Similarly, Kim et al. (2016b) proposed a Multimodal Low-rank Bilinear (MLB) pooling model, which constrains the tensor T to be of low rank in order to limit the number of free parameters. However, both MCB and MLB introduce constraints that limit the expressiveness of the bilinear interaction. In MUTAN, Ben-younes et al. generalize these approaches by leveraging the Tucker decomposition (Tucker, 1966). The Tucker decomposition factorizes the full bilinear interaction into a product of a small core tensor $T_c \in \mathbb{R}^{t_q \times t_v \times t_o}$ with three factor matrices $\boldsymbol{W}_q \in \mathbb{R}^{d_q \times t_q}$, $\boldsymbol{W}_v \in \mathbb{R}^{d_v \times t_v}$, $\boldsymbol{W}_o \in \mathbb{R}^{|\mathcal{A}| \times t_o}$:

$$T = ((T_c \times_1 (Q^T \boldsymbol{W_q})) \times_2 (I^T \boldsymbol{W_v})) \times_3 \boldsymbol{W_o} \tag{1.2}$$

Notably, the Tucker composition fully captures the expressivity of the bilinear interaction, while the other approaches discussed do not. In this way, MUTAN is able to surpass the performance of MCB and MLB, while limiting the number of trainable parameters to a reasonable 4.9 million.

1.6.2 Bottom-Up / Top-Down Attention

Visual attention is one of the most successful ideas to emerge from computer vision in the past decade. Attention mechanisms, which allow a network to direct additional processing resources to salient image regions, have been successfully employed in both image captioning (Xu et al., 2015; Lu et al., 2017) and VQA (Lu et al., 2016; Xu & Saenko, 2016; Yang et al., 2016). The concept of vi-

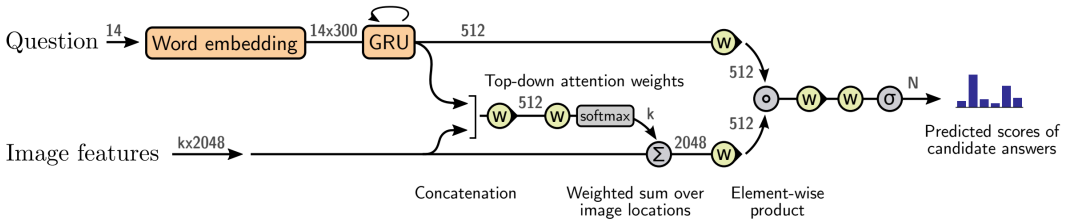


Figure 1.7: Bottom-Up / Top-Down Attention model architecture. The model uses attention to combine visual and linguistic information into a joint embedding. (Figure from Anderson et al., 2018)

sual attention is loosely-inspired by human biology. As Anderson et al. (2018) note, neuroscientists have long hypothesized the existence of two distinct attentional mechanisms. While "top-down" attention is directed volitionally towards task-relevant stimuli, "bottom-up" attention is driven by unexpected, novel, or salient stimuli in an automatic manner (Buschman & Miller, 2007; Corbetta & Shulman, 2002).

In a typical VQA model, the image encoding module consists of feed-forward CNN, which provides a high-level feature representation of the image. These networks operate in a bottom-up manner, learning to focus on image regions that are generally relevant across many contexts. In their Bottom-Up / Top-Down Attention model, Anderson et al. (2018) introduce an additional top-down attention mechanism, which uses task-specific context to further direct visual attention. Figure 1.7 outlines the architecture of the Bottom-Up / Top-Down network. First, Faster R-CNN (Ren et al., 2015b) is used to compute a set of variable-sized bounding boxes corresponding to recognized objects in the image. Feature vectors corresponding to these regions of interest are extracted with ResNet-101 (He et al., 2015a). A mean-pooled representation of these bottom-up image features is fed as auxiliary input into an LSTM, which encodes the question in light of the overall content

of the image. However, at each timestep, the question encoder also produces a top-down attention distribution over the image features. Finally, the question and attention-weighted image features are combined via element-wise product. In summary: bottom-up image features into the question encoder, which in turn predicts a top-down attention distribution over these features. In this way, the Bottom-Up / Top-Down Attention model uses attention as a vehicle for combining visual and linguistic information into a multimodal representation.

The key to artificial intelligence has always been the

representation.

—Jeff Hawkins

Neuroscientist and Founder, Numenta

Der Teufel steckt im Detail. (The devil is in the details.)

—German proverb, c. 1800s

2

On the Flip Side: Identifying Counterexamples in VQA

The ability to distinguish between perceptually-similar stimuli is a hallmark of intelligence. In the face of similar-looking objects, children as young as three years make valid inferences based on discrete categorical properties (Markman & Hutchinson, 1984; Gelman & Markman, 1987). Thus, it is naturally desirable that a computational model for visual reasoning should be able to make such

distinctions.

As discussed in the introduction, VQA is often framed as a general test of visual-semantic reasoning (Antol et al., 2015; Kafle & Kanan, 2017; Chao et al., 2017). However, researchers have recently begun to question whether existing state-of-the-art models actually learn to make meaningful semantic distinctions between visually-similar images (Goyal et al., 2016; Agrawal et al., 2018; Johnson et al., 2017; Thomason et al., 2018). In this chapter, we explore a reformulation of the VQA task that more directly evaluates a model's capacity to reason about the underlying concepts encoded in images. Under the standard VQA task, given the question "What color is the fire hydrant?" and an image of a street scene, a model might answer "red." Under the alternative task, the model must produce a counterexample; e.g., an image of a fire hydrant that is not red. Successful performance on the visual counterexample prediction task (abbreviated VQA-CX) requires reasoning about how subtle visual differences between images affect the high-level semantics of a scene.

The VQA-CX task was originally proposed in Goyal et al. (2016) as a useful explanation modality for VQA models. However, despite its applicability as a powerful tool for model introspection, this idea has remained largely under-explored by the research community. To our knowledge, this work represents the first follow-up attempt to operationalize the VQA-CX paradigm originally proposed by Goyal et al. (2016).

We introduce two plug-and-play approaches for evaluating the performance of existing, pretrained VQA models on VQA-CX. The first method is an unsupervised model that requires no training and works out-of-box with a pretrained VQA model. The second method is a supervised

°Chapter 2 is adapted from Grand et al. (2018b).

neural model that can be used with or without a pretrained VQA model. The unsupervised model outperforms the baselines proposed in Goyal et al. (2016). Meanwhile, the supervised model outperforms all existing unsupervised and supervised methods for counterexample prediction.

Crucially, while we use a state-of-the-art VQA model to facilitate counterexample prediction, we find that our methods perform almost as well without receiving any information from this model. In other words, the multimodal representation learned by the VQA model contributes only marginally (approximately 2%) to performance on VQA-CX. These results challenge the assumption that successful performance on VQA is indicative of more general visual-semantic reasoning abilities.

2.1 Approach

We treat VQA-CX as a supervised learning problem, which can be formalized as follows. For each image, question, and answer (I, Q, A) in the original VQA task, the model is presented with the the $K = 24$ nearest neighbor images $I_{\text{NN}} = \{I'_1, \ldots, I'_K\}$ of the original image. The model assigns scores $S = S(I'_1), \ldots, S(I'_K)$ to each candidate counterexample. The crowd-selected counterexample $I^* \in I_{\text{NN}}$ serves as ground truth. For notational clarity, we distinguish between raw images I and convolutional image features v. Additionally, we use prime notation (I', A') to denote candidate counterexamples, asterisk notation to denote the ground truth counterexample (I^*, A^*), and no superscript when referring to the original example (I, A). We do not use any superscripts for Q, since the question is the same in all cases.

Figure 2.1: The goal of VQA-CX is to identify a counterexample (green border) that results in a different answer to the question from a set of 24 visually-similar images.

Both of our VQA-CX models use an existing VQA model as a submodule. While there exist many diverse models for VQA (Wu et al., 2017), we mostly treat the VQA model as a black box that can be expressed as a function of its inputs. We make only two assumptions about the architecture. First, we assume the model outputs a distribution $P(\mathcal{A}|I, Q)$ over a discrete number of answer classes.[1] Second, we assume the model internally combines its inputs into some multimodal representation z, which we can access. (Note that this second assumption, which violates the black box

[1]As discussed in Section 1.5.4, while most models treat VQA as a classification task, some adopt a generative approach (e.g., Wu et al. (2016); Zhu et al. (2015); Wang et al. (2017)), which is not compatible with this assumption.

principle, is only used optionally in the NeuralCX model.) We therefore treat a VQA model as a function $\text{VQA}(I, Q) = P(\mathcal{A}|I, Q), z$.

In order to establish a basis for comparison with Goyal et al. (2016), we began by reproducing their baselines, described in the following section. We then developed two architectures for VQA-CX. Both models can be used in conjunction with any VQA model that meets the above two criteria. The first architecture, which we call the Embedding Model, compares the semantic similarity between candidate answers in an embedding space, weighing different answers by $P(\mathcal{A}|I, Q)$. Since the Embedding Model relies solely on a pretrained VQA model and a pretrained answer embedding, it is fully unsupervised and requires no training. The second architecture is a straightforward multilayer perceptron that takes as input features related to I, I', Q, and A, including the outputs of a VQA model, and returns a score $S(I')$. This NeuralCX model is trained in a pairwise fashion using standard supervised learning methods.

2.2 Models

2.2.1 Prior Work

To our knowledge, the only previous work on VQA-CX was carried out by the authors of the VQA v2 dataset. Goyal et al. (2016) present a two-headed model that simultaneously answers questions and predicts counterexamples. The model consists of three components:

Shared base: Produces a joint embedding of the question and image via pointwise multiplication.

$$z = \text{CNN}(I) \odot \text{LSTM}(Q)$$

During a single inference step, a total of $K + 1$ images (the original image and its KNNs) are passed through this component.

Answering head: Predicts a probability distribution over answer classes.

$$P(\mathcal{A}|Z) = \sigma(W_{out}z + b_{out})$$

Only the z corresponding to the original image is used in the answering head.

Explaining head: Predicts counterexample scores for each of K nearest neighbor images.

$$S(I'_i) = (W_{zd}z_i + b_{zd}) \cdot (W_{ad}\mathcal{A} + b_{ad})$$

This component can be seen as computing vector alignment between a candidate counterexample and the ground truth answer. To allow for the dot product computation, z_i and $\mathcal{A}$ are both projected into a common embedding space of dimensionality d. Note that in the final layer of the network, all K scores $S = S(I'_1), \ldots, S(I'_K)$ are passed through a $K \times K$ fully-connected layer. Presumably, this layer is intended to allow the model to learn the distribution over the rank of I^* within I_{NN}. However, as we note in Section 2.5, this layer functions as a bottleneck that limits the expressivity of the model outputs.

The two-headed model is trained on a joint loss that combines supervision signals from both heads.

$$L(S) = -\log P(\mathcal{A}|I, Q) + \lambda \sum_{I'_i \neq I^*} \max(0, \mathcal{M} - (S(I^* - I'_i)))$$

The answer loss is simply the cross entropy loss induced by the ground truth answer $\mathcal{A} \in \mathcal{A}$. Meanwhile, the explanation loss is a pairwise hinge ranking loss (Chopra et al., 2005), which encourages the model to assign the ground-truth counterexample I^* a higher score than the other candidates.

2.2.2 Baselines

In addition to their counterexample model, Goyal et al. (2016) introduce three key baselines for VQA-CX:

- **Random Baseline:** Rank I_{NN} randomly.
- **Distance Baseline:** Rank I_{NN} by L2 distance from I. Closer images are assigned higher scores.
- **Hard Negative Mining:** For each $I'_i \in I_{\text{NN}}$, determine the probability of the original answer $P(\mathcal{A})_i = \text{VQA}(I'_i, Q)$ using a pretrained VQA model. Rank the I'_i according to *negative* probability $-P(\mathcal{A})_i$. In other words, choose counterexamples for which the VQA model assigns low probability to the original answer.

2.2.3 Unsupervised Embedding Model

Successful performance on VQA-CX requires reasoning about a complex semantic relationship between answers. While the counterexample answer $\mathcal{A}^*$ is distinct from the original answer $\mathcal{A}$, the two are often close neighbors in semantic space. For example, for the question-answer pair ($Q =$

"What animal is in the tree?"; $\mathcal{A}$ ="cat"), the counterexample answer is more likely to be "dog" than "meatball," even though the semantic distance between "cat" and "meatball" is greater. Ideally, a VQA-CX model should capture this nuanced relationship between complementary pairs of answers.

The Embedding Model balances the goal of identifying a semantically-similar counterexample answer with the requirement that the answer not be identical to the original. The model uses answer-class predictions $P(\mathcal{A}|I', Q)$ from a pretrained VQA model, and answer embeddings $W_{\mathcal{A}}$ from a pretrained Skip-Thoughts model (Kiros et al., 2015) to assign a score to each nearest neighbor image:

$$S(I'_i) = \lambda \sum_{\substack{a \in \mathcal{A}; \\ a \neq A}} \text{cossim}\,(a, A)\, P(a|I'_i, Q) - (1 - \lambda) \log P(A|I'_i, Q) \tag{2.1}$$

The term to the left of the subtraction encourages the model to select counterexamples that produce answers similar to the original. Meanwhile, the term to the right discourages the model from selecting the exact same answer as the original. The λ hyperparameter, chosen empirically, determines the relative weight of these terms.

2.2.4 Supervised NeuralCX Model

NeuralCX is a fully-connected network that takes as input 10 features derived from I, I', Q, and A. Some of these features, such as v, q, and a, are feature representations of the original image, question, and answer. Others, such as z and $P(\mathcal{A}')$, are computed by a VQA model. Table 2.1 summarizes the input features.

All features are concatenated into a single input vector and passed through a series of hidden

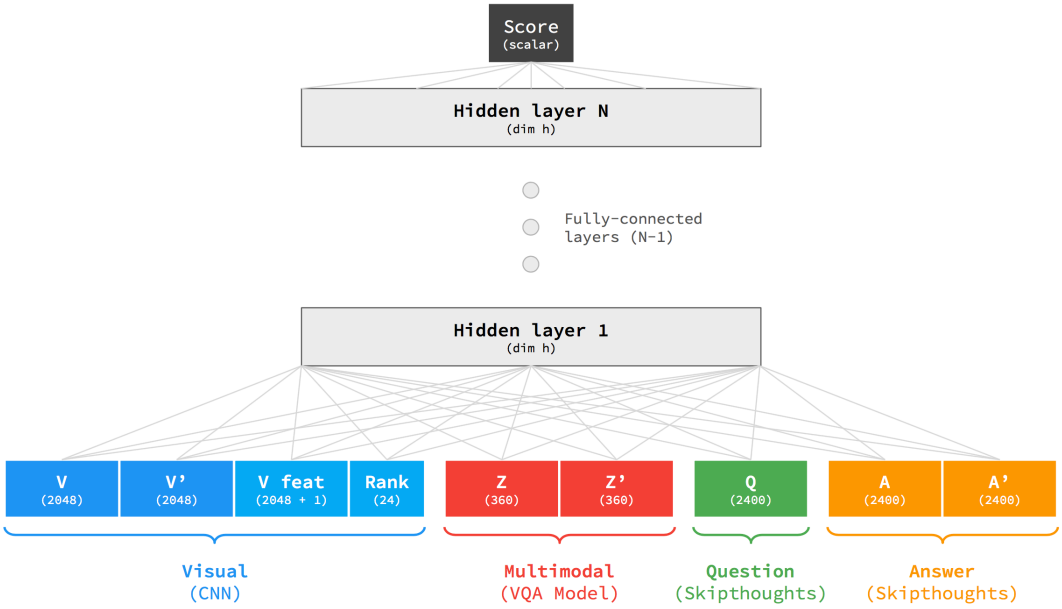


Figure 2.2: Diagram of NeuralCX architecture. The model is a fully-connected neural network that takes visual, question, answer, and multimodal features as input and produces a score indicating the relevance of I' as a counterexample.

layers, where the size h and number N of layers are hyperparameters. All layers share the same h and use ReLU activation. The output of the last hidden layer is projected to an unnormalized scalar score $S(I')$. Figure 2.2 depicts the NeuralCX architecture.

A single training iteration for NeuralCX consists of K forward passes of the network to produce a score for each candidate $I'_i \in I_{\text{NN}}$. We compute the cross-entropy loss for the ground truth I^*, and optimize the parameters of the network via backpropagation.

Feature	Definition	Size	Origin
v	CNN(I)	2048	CNN
v'	CNN(I')	2048	CNN
v_M	$v \odot v'$	2048	Computed
v_D	$\|\|v' - v\|\|$	1	Computed
Rank	onehot(i)	24	Computed
q	LSTM(Q)	2400	LSTM
a	$W_{\mathcal{A}}\mathcal{A}$	2400	$\mathcal{A}_{\text{EMB}}$
a'	$(W_{\mathcal{A}})^T P(\mathcal{A}')$	2400	$\mathcal{A}_{\text{EMB}}$, VQA
z	VQA(I, Q)	360	VQA
z'	VQA(I', Q)	360	VQA

Table 2.1: Full set of features input to NeuralCX model.

2.3 Methods

Our VQA-CX dataset consists of 211k training examples and 118k test examples, of which 10k were reserved as a validation set. A single example in our dataset consists of the original VQA v2 example $(I, Q, \mathcal{A})$, and the 24 nearest neighbor images I_{NN}. One of these neighbors is guaranteed to contain the ground truth counterexample $I^* \in I_{\text{NN}}$ and its corresponding answer $\mathcal{A}^*$.

Our train and test data are, by necessity, proper subsets of the VQA v2 training and validation datasets, respectively. To construct our trainset, we first identified the examples for which the image I had a corresponding I^*. Approximately 22% of the images in VQA v2 do not have a labeled complement.[2] Next, we filtered out examples for which I^* did not appear in I_{NN}. Since we used the nearest neighbors data provided by Goyal et al. (2016), I^* should theoretically always appear in I_{NN}.

[2] These images correspond to instances in which crowd workers indicated that it was not possible to select a counterexample from among I_{NN} (Goyal et al., 2016).

However, because the KNN relation is not symmetric (i.e., $I^1 \in I^2_{\text{NN}} \nRightarrow I^2 \in I^1_{\text{NN}}$), we found that in certain cases, $I^* \notin I_{\text{NN}}$. After filtering, we were left with $211,626/433,757$ training examples and $118,499/214,354$ validation examples. Note that while Goyal et al. (2016) also collected labeled counterexamples for the VQA v2 test split, this data is not public. As a result, we did not make use of the VQA v2 test set, instead testing on the VQA v2 validation set.

We implemented our models and experiments in Pytorch (Paszke et al., 2017).[3] For all experiments involving VQA models, we used the MUTAN model presented in Section 1.6.1 (Ben-younes et al., 2017). We pretrained MUTAN separately on VQA v2 for 100 epochs with early stopping to a peak test accuracy of 47.70. Unfortunately, because we needed to train the model on only the VQA v2 training set (and not the validation set), this accuracy is considerably lower than the 58.16 single-model accuracy obtained by Ben-younes et al. (2017). Additionally, since the VQA-CX task requires us to load all 24 V_{NN} features into memory simultaneously, we opted use a no-attention variant of MUTAN that is more space-efficient, but lower-performing. We used a pretrained ResNet-152 model (He et al., 2015a) to precompute visual features for all images, and a pretrained Skip-Thoughts model (Kiros et al., 2015) to compute question and answer embeddings. We also utilized framework code from the vqa.pytorch Github repository.[4]

For all experiments with the NeuralCX model, we trained for a maximum of 20 epochs with early stopping. We optimized the model parameters with standard stochastic gradient descent methods, using the Pytorch library implementation of Adam (Kingma & Ba, 2014) with learning rate 0.0001

[3] Code for this project is available at https://github.com/gabegrand/VQA-Counterexamples.
[4] https://github.com/Cadene/vqa.pytorch

and batch size 64. We also employed dropout regularization ($p = 0.25$) between hidden layers (Srivastava et al., 2014).

We experimented with different numbers of hidden layers $N = 1, 2, 3$ and hidden units $h = 256, 512, 1024$, but found that larger architectures resulted in substantial training time increases with negligible performance gains. We therefore used a moderate-sized architecture of $N = 2$, $h = 512$ for all reported results. This model takes about 35 minutes to train to peak performance on a single Tesla K80 GPU.

We evaluate the performance of our models and baselines with recall@k, which measures the percentage of the ground truth counterexamples that the model ranks in the top k out of the 24 candidate counterexamples. Results on the test set for the NeuralCX Model, Embedding Model, and baseline models are reported in Table 2.2. To better understand the relative importance of the different inputs to the NeuralCX model, we selectively ablated different features by replacing them with noise vectors drawn randomly from a uniform distribution. We chose to randomize inputs, rather than remove them entirely, so as to keep the model architecture constant across experiments. In each ablation experiment, the model was fully retrained. Results from these experiments are reported in Table 2.3.

2.4 Results

We began by reimplementing the baselines presented by Goyal et al. (2016) and comparing our results with theirs. As expected, the Random Baseline performed approximately at chance (recall@5

CX Model	VQA Model	Our Results Recall@1	Our Results Recall@5	Goyal et al. Recall@5
Random Baseline	-	4.20	20.85	20.79
Hard Negative Mining	untrained	4.06	20.73	-
Hard Negative Mining	pretrained	4.34	22.06	21.65
Embedding Model	untrained	4.20	21.02	-
Embedding Model	pretrained	7.77	30.26	-
Distance Baseline	-	11.51	44.48	42.84
Two-headed CX	trainable	-	-	43.39
NeuralCX	untrained	16.30	52.48	-
NeuralCX	pretrained	18.27	54.87	-
NeuralCX	**trainable**	**18.47**	**55.14**	-

Table 2.2: Results of VQA-CX models and baselines. Where applicable, we compare our results with those reported in Goyal et al. (2016). The midline separates unsupervised models (above), which were evaluated without training on VQA-CX, with those that were trained on VQA-CX (below). We also distinguish how the underlying VQA model was trained: "untrained" denotes that the VQA model parameters were randomly-initialized and immutable; "pretrained" denotes parameters that were learned on the VQA task and then made immutable; and "trainable" denotes parameters that were first learned on VQA, and then fine-tuned on VQA-CX.

$\approx \frac{5}{24}$ or 0.2083). Our Distance Baseline was comparable with, but slightly higher than, the result reported by Goyal et al. This discrepancy indicates that it is possible that the distribution over the rank of the ground-truth counterexample is more skewed in our dataset than in the one used by Goyal et al. Notably, in both cases, the strategy of ranking counterexample images based on distance in feature space is more than two times better than chance, and serves as a strong baseline.

As in Goyal et al. (2016), we found Hard Negative Mining to be a relatively under-performing approach. Since we used a different VQA model from Goyal et al., our results on this baseline are not directly comparable. Nevertheless, in both cases, Hard Negative Mining performed only marginally above chance (+1.21% points in our case, and +0.86 points in theirs). To isolate the impact of the

VQA model, we computed the Hard Negative Mining baseline using an untrained (randomly initialized) VQA model. After this change, the performance dropped to random.

The Embedding Model performed between Hard Negative Mining and the Distance Baseline. Interestingly, the value of λ that maximized performance was 1.0, meaning that integrating the overt probability of A under the VQA model only hurt accuracy. We observed a smooth increase in performance as we varied λ between 0 and 1. Clearly, there is some signal in the relative position of the candidate answer embeddings around the ground truth answer, but not enough to improve on the information captured in the visual feature distance.

The NeuralCX model significantly outperformed both the Distance Baseline and the two-headed model from Goyal et al. (2016). To quantify the impact of the VQA model on the performance of NeuralCX, we tested three conditions for the underlying VQA model: untrained, pretrained, and trainable. In the untrained condition, we initialized NeuralCX with an untrained VQA model. In the pretrained condition, we initialized NeuralCX with a pretrained VQA model, which was frozen during VQA-CX training. In the trainable condition, we allowed gradients generated by the loss layer of NeuralCX to backpropagate through the VQA model. We found that fine-tuning the VQA model in this manner produced small gains over the pretrained model. Meanwhile, with an untrained VQA model, the recall@5 of NeuralCX was only 2.39% points lower than with a trained model.

In the NeuralCX ablation experiments (Table 2.3), we found that visual features were crucial to strong performance. Without any visual features, recall fell below the Distance Baseline. Both v and the rank embedding appear to be especially important to the task. Intriguingly, these features also

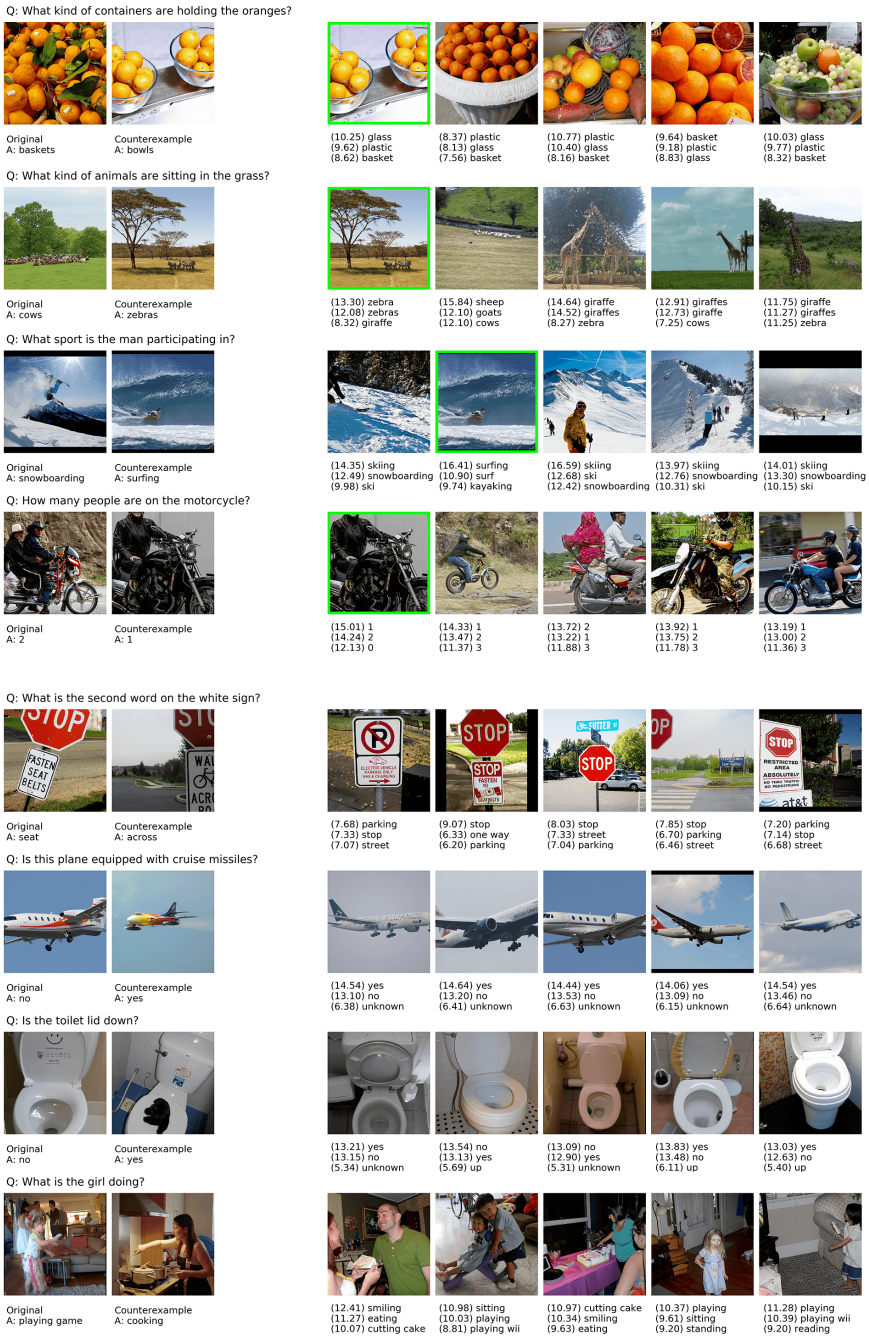


Figure 2.3: Qualitative results on the counterexample prediction task. Left: The original image and ground truth counterexample from VQA v2, along with the question and ground truth answers. Right: the top 5 counterexamples selected by NeuralCX, with the top 3 answers (as scored by a VQA model) underneath. In the top 4 rows, NeuralCX correctly identifies the correct counterexample (green outline), while in the bottom 4 rows, it fails. See Section 2.5 for a discussion of common failure modes.

Performance		Ablated Features						
R@5	R@1	v	v_M	v_D	Rank	q	a	z
43.05	12.33	✖	✖	✖	✖			
44.48	11.42	✖						
44.48	11.51		✖	✖	✖			
44.48	11.52	✖	✖	✖		✖	✖	✖
44.55	13.17				✖			
47.09	13.29					✖	✖	✖
52.18	16.48						✖	
54.87	18.27					✖		
54.87	18.27							✖
54.87	18.27							

Table 2.3: Selective ablation of NeuralCX inputs. Features that are marked ✖ are replaced with noise. Ablations are sorted from top to bottom in order of disruptiveness, with the bottom row showing results from an unablated model. The different features are defined in Table 2.1.

appear to be interdependent; ablating either v or the rank embedding was almost as disruptive as ablating both. Meanwhile, we found that ablating the non-visual features produced a much smaller impact. While ablating a resulted in a small performance drop, ablating q and z did not affect performance at all.

2.5 Discussion

Our results highlight both promises and challenges associated with VQA. On the one hand, the fact that NeuralCX outperforms the methods from Goyal et al. (2016) demonstrates that the data contain enough signal to support supervised learning. This result is especially important in light of the pronounced skew in the distribution over the rank of I^* in the dataset, which makes approaches based merely on image distance unreasonably dominant. Given that the supervised neural model

from Goyal et al. (2016) barely surpasses the Distance Baseline, it seems likely that this model overfits to the I^* rank distribution. Indeed, the $K \times K$ fully-connected layer of this model severely limits the information that can pass through to the output. Due to this bottleneck, it is unlikely that this network learns anything other than the optimal activation biases of the K output units.

In contrast, we observed that NeuralCX effectively leverages both visual and semantic information. When provided with only visual features, the recall@5 for NeuralCX was 7.78% points lower than when the model was provided with both visual and semantic features (Table 2.3). In particular, the answer embedding provides information about the semantic similarity between A' and A, which we hypothesize allows the model to select counterexamples that are semantically distinct from the original example. The strong performance of the Embedding Model—which bases its predictions solely on answer similarity, and does not model image distance—also supports this hypothesis. Thus, while visual similarity remains a crucial feature for VQA-CX, our findings demonstrate that in order to achieve peak performance on this task, a model must also leverage semantic information.

While our results indicate that the answer embeddings encode task-relevant information, the same cannot be said for the multimodal embeddings z produced by the VQA model. In our ablation experiments, we found that replacing z and z' with noise did not affect the performance of NeuralCX. Since z is, by definition, a joint embedding of q and v, it is possible that z encodes redundant information. However, if this were the case, we would expect z to help the model in cases where visual features are not available. Instead, we see a significant drop in accuracy when we ablate the visual features but leave z, indicating that z does not support the recovery of visual features.

Our experiments with untrained VQA models suggest that the representations learned by the

VQA model do not contain useful information for identifying counterexamples. Replacing the pretrained VQA model with an untrained version results in a decrease of only 2.39% recall@5. (Based on our ablation experiments, this performance hit is not due to the loss of z, but rather, the loss of the distribution over the counterexample answer $P(\mathcal{A}')$, which is used to weight the embedding representation of $\mathcal{A}'$). One could argue that it is unfair to expect the VQA model to provide useful information for VQA-CX, since it was not trained on this task. However, when we co-train the VQA model with NeuralCX, we find only a small performance improvement compared to the pretrained model. This result holds regardless of whether the VQA model is initialized from pretrained weights when trained on VQA-CX.

This transfer failure raises questions about the extent to which models that perform well on the VQA dataset actually learn semantic distinctions between visually-similar images. In our qualitative analysis, we found that while the VQA model often produces the correct answer, it also assigns high probability to semantically-opposite answers. For instance, when answering "yes," the model's other top guesses are almost always "no" and "unsure." Similarly, counting questions, the VQA model often hedges by guessing a range of numbers; e.g., "1, 2, 3" (see Figure A.3). While this strategy may be optimal for the VQA task, it suggests that the VQA model is effectively memorizing what types of answers are likely to result from questions. In other words, it is unclear from these results whether the VQA model can actually distinguish between the correct answer and other answers with opposite meanings.

While our results expose issues with existing approaches to VQA, it is important to consider two external failure modes that also affect performance on VQA-CX. First, in some cases, NeuralCX

fails to fully utilize information from the VQA model. Even when the VQA model correctly identifies a particular I' as producing the same answer as the original, NeuralCX still sometimes chooses I' as the counterexample. In other cases, NeuralCX incorrectly assigns high scores to images for which $A' \approx A$; e.g., an image of children "playing" was selected as a counterexample to an image of children "playing game." These failures indicate that NeuralCX does not optimally leverage the semantic information provided by the VQA model.

The second failure mode arises from issues with the data itself. While the complementary pairs data in VQA v2 makes it possible to formalize counterexample prediction as its own machine learning task, several idiosyncrasies in the data make VQA-CX a partially ill-posed problem.

- There may be multiple images in I_{NN} that could plausibly serve as counterexamples. This is particularly evident for questions that involve counting (e.g., for $Q =$ "How many windows does the house have?" the majority of images in I_{NN} are likely to contain a different number of windows than the original image.) In many cases, our models identified valid counterexamples that were scored as incorrect, since only a single $I^* \in I_{\text{NN}}$ is labeled as the ground truth.
- For approximately 9% of the examples, the counterexample answer A^* is the same as A. This irregularity is due to the fact that the tasks of identifying counterexamples and assigning answer labels were assigned to different groups of crowd workers (Goyal et al., 2016). In addition to potential inter-group disagreement, the later group had no way of knowing the intentions of the former. This discontinuity resulted in a subset of degenerate ground truth counterexamples.
- The distribution over the rank of I^* within I_{NN} is not uniform; there is a strong bias towards closer nearest neighbors. In the training set, I^* falls within the top 5 nearest neighbors roughly 44% of the time.
- Certain questions require common knowledge that VQA models are unlikely to possess (e.g., "Is this a common animal to ride in the US?"; "Does this vehicle require gasoline?").

- Other questions require specialized visual reasoning skills that, while within reach for current machine learning methods, are unlikely to be learned by general VQA architectures (e.g., "What is the second word on the sign?" or "What time is on the clock?")
- Finally, a small portion of the questions in VQA v2 simply do not admit to the counterexample task. For instance, given the question, "Do zebras have horses as ancestors?" it is impossible to select an image, zebra or otherwise, that reverses biological fact.

While these idiosyncrasies in the data complicate the task of counterexample prediction, we nevertheless view work on VQA-CX as crucial to the broader goals of representation learning. As leaderboard-based competitions like the VQA Challenge continue to steer research efforts towards singular objectives, auxiliary tasks like VQA-CX present a useful opportunities to sanity-check our progress. In this case, our results suggest that the representations learned by current VQA models may not capture key semantic distinctions between visually-similar images. These findings, which coincide with a growing both of work that shows that VQA models over-rely on superficial patterns in the data, call for further efforts to improve visual grounding in VQA models.

We must learn actually not to have enemies, but only

confused adversaries who are ourselves in disguise.

—Alice Walker

American Poet, Pulitzer Prize Winner

3

Adversarial Regularization: Reducing Language Bias in VQA

In light of the evidence presented so far, learned bias stands as a significant obstacle to future research progress in VQA. Chapter 1 established that language priors are a pervasive and inevitable component of any naturalistic VQA dataset. Meanwhile, Chapter 2 demonstrated that the multimodal representations learned by high-performing VQA models do not necessarily encode key se-

mantic distinctions between visually-similar images. Thus, while VQA v2 is a step towards "making the V in VQA matter" (Goyal et al., 2016), the inclusion of negative examples in the data is not sufficient to wean VQA models off their dependence on language priors. For all intents and purposes, contemporary VQA models remain "hooked on phonics."

While we may not be able to rid the VQA datasets of their omnipresent language priors, perhaps we can inoculate our models against the toxic influence of these biases. In this chapter, we explore one possible rehabilitation method, which we call "adversarial regularization." Section 3.1 begins by developing an understanding of regularization and adversarial methods in statistical machine learning. Combining these concepts, Section 3.2 introduces our approach to adversarial regularization. Sections 3.3 and 3.4 present our experiments and results, which demonstrate that adversarial regularization successfully reduces dependence on language biases in VQA models. Finally, in Section 3.5, we conclude with a discussion of the key questions raised by our findings.

3.1 Background

3.1.1 Regularization in Statistical Machine Learning

One of the central goals of machine learning is to produce models that will generalize well to unseen data. Regularization is an important tool for achieving this aim. Goodfellow et al. (2016) define regularization as "any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error." In other words, regularization methods are designed to achieve lower test error, but may come at the expense of increased training error. This trade-off is

a fundamental aspect of regularization.

A broad class of regularization methods are in practice today. Many of these involve imposing constraints on a model's behavior through the addition of an extra loss term.

$$\tilde{L}(\Theta, X, y) = L(\Theta, X, y) + \lambda R(\Psi) \tag{3.1}$$

Here, λ is a hyperparameter that controls the weight of the regularization penalty $R(\Psi)$ relative to the loss term $L(\Theta, X, y)$. Meanwhile, Ψ can stand in for many different metrics that we may wish to regulate. For instance, a common form of regularization sets $\Psi = |\Theta|$, imposing a penalty on the norm of a model's parameters so as to discourage the model from learning large parameter values. Another approach is to set $\Psi = h$, so as to enforce sparsity constraints on the internal representations h learned by the model. Finally, a prominent regularization technique called "dropout" enforces sparsity at the level of a model's layer-wise activation functions $\Psi = f(h)$, randomly zeroing the outputs of a certain fraction of the model's hidden units (Srivastava et al., 2014). While these techniques each affect different components of the model, they all serve to reduce over-fitting by encouraging the model to be more robust to variation in the data.

3.1.2 Adversarial Methods

In the previous section, we demonstrated how the addition of an extra loss term $\lambda R(\Psi)$ captures many common forms of regularization in machine learning. In the methods presented above, Ψ stands in for some intrinsic component of the model that we wish to regulate. However, it is also possible for $R(\Psi)$ to reflect metrics associated with a *different* model.

The concept of Generative Adversarial Networks (GANs), first introduced in Goodfellow et al. (2014), can be viewed as a regularization framework in which two models mutually regulate one another's behavior. In its classical formulation, adversarial learning involves a two-player minimax game between two opponent networks G and D, which are co-trained to optimize the following loss function (Goodfellow et al., 2014):

$$\min_G \max_D L(G, D) = E_{x \sim p_{\text{data}}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{3.2}$$

In the original context, G is a generator, whose goal is to produce synthetic samples that reflect the data distribution. Meanwhile, D is a discriminator, whose goal is to ascertain whether a particular example came from the original data, or from G. Informally, the discriminator can be seen as a regularizer for the generator: when D is able to accurately distinguish the samples produced by G, the loss value of Equation 3.2 is high for G. However, as Goodfellow et al. note, GANs are not, strictly speaking, a form of regularization, since the competition between the networks is the sole training criterion. Nevertheless, this style of adversarial approach, in which successful discrimination by an adversary is used as a negative signal for a main network, is highly applicable to other, non-generative tasks in machine learning.

3.2 Adversarial Regularization

3.2.1 Domain-Adversarial Training

We come now to the proper intersection between regularization and adversarial methods. In their paper, "Domain-Adversarial Training of Neural Networks" (2016), Ganin et al. lay out a framework for leveraging adversarial methods to assist in domain adaptation. The goal of domain adaptation is to improve the performance of a model when trained and tested on data that come from distributions with different characteristics. Theoretical work on domain adaptation suggests that, in order for a representation to transfer successfully from a source domain $\mathcal{D}_S$ to a target domain $\mathcal{D}_T$, it must not depend on features that are particular to either domain (Ben-David et al., 2007, 2010).

To operationalize this idea, Ganin et al. introduce Domain Adversarial Neural Networks (DANNs), which consist of three components (see Figure 3.1). The feature extractor network G_f produces feature representations of examples x. The label predictor network G_y takes these representations as input, and predicts the class label y associated with x. Finally, the feature representations are also fed to the domain classifier network G_d, which attempts to infer whether x came from the source or target domain.

During training, the main network $G_y(G_f(x; \Theta_f); \Theta_y)$ is trained on labeled examples from the source domain $x \in \mathcal{D}_S$ in the usual manner, using backpropagation via stochastic gradient descent. Simultaneously, the domain classifier network $G_d(G_f(x; \Theta_f); \Theta_d)$ is trained on both labeled examples $x \in \mathcal{D}_S$ and unlabeled examples $x \in \mathcal{D}_T$. The parameters of the domain classifier Θ_d are up-

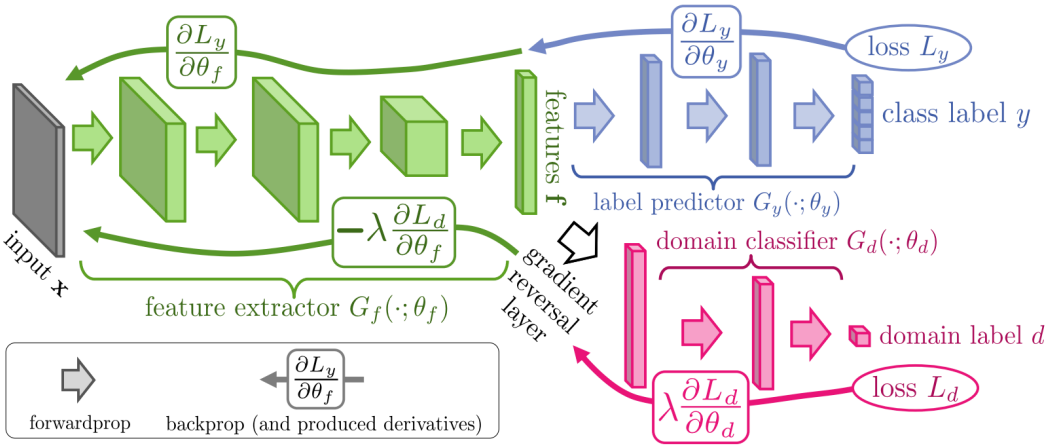


Figure 3.1: Architecture of a Domain Adversarial Neural Network. Inputs are processed by a shared feature extractor (green). The label predictor (blue) performs the main inference task, while the domain classifier (pink) attempts to infer whether the sample came from the source or target domain. A gradient reversal layer is used to negate updates from the domain classifier during backpropagation. (Figure from Ganin et al., 2016)

dated via gradient descent to minimize the network's cross entropy loss on the binary classification task. However, before backpropagation reaches G_f, the gradients from G_d pass through a gradient reversal layer GRL_λ (Ganin et al., 2016):

$$\text{GRL}_\lambda(x) = x \tag{3.3}$$

$$\frac{\partial \text{GRL}_\lambda}{\partial x} = -\lambda_{\text{GRL}} I \tag{3.4}$$

During the forward pass, the GRL acts as the identity function, leaving the input untransformed. Meanwhile, during the backward pass, the GRL negates the gradients and scales them by a hyper-parameter λ_{GRL}, which controls the magnitude of the gradients passed to G_f. In practice, the GRL

can be written in a few lines of code, and implementations exist already in PyTorch[1] and Caffe.[2] Reversing the gradients passed from G_d to G_f encourages the feature extractor network to learn more robust representations that do not encode distinctions between the source and target domains. Indeed, Ganin et al. (2016) found that DANNs improved domain adaptation performance on image classification and document sentiment analysis tasks.

While domain-adversarial training is an effective method for adapting machine learning models from one domain to another, it requires training on data from the target domain. Even though the target domain data need not be labeled, this requirement is still problematic for both practical and philosophical reasons. From a practical perspective, in many cases, examples from the target domain may be in short supply, or else we may not have knowledge of the target domain at all. Indeed, data scarcity is a major issue in many applied machine learning domains, such as medicine and materials science (Hutchinson et al., 2017; Shaikhina & Khovanova, 2017). Moreover, even when we do have access to plentiful examples from both source and target domains, we may deliberately wish to avoid training on the target domain. For instance, while we could hypothetically perform domain-adversarial training on the VQA-CP train and test splits, there is a sense in which this approach defeats the purpose of the VQA-CP benchmark. In VQA, we are not be interested in adapting the model to the particular answer-class biases contained in the VQA-CP test split, so much as improving transfer to any such target domain with different language biases from the source. In other words, the purpose of the VQA-CP test split is to assess transfer performance on an *unseen*

[1]GRL in Pytorch: https://discuss.pytorch.org/t/solved-reverse-gradients-in-backward-pass/3589
[2]GRL in Caffe: https://github.com/ddtm/caffe/tree/grl

domain with novel biases. Thus, in the interest of learning maximally-transferable models, it is desirable that our regularization scheme should not require access to training data from the target domain.

3.2.2 Adversarial Training in Natural Language Inference

While VQA reveals a number of drawbacks associated with domain-adversarial regularization, these issues are not isolated to VQA. Recently, a number of follow-up ideas on adversarial regularization have emerged from the NLP literature in the context of a task called Natural Language Inference (NLI). The goal of NLI is to determine whether a given premise (e.g., "A young family enjoys feeling ocean waves lap at their feet") supports a particular hypothesis (e.g., "A family is at the beach").

As in VQA, a growing body of work points to the existence of pervasive "biases" or "annotation artifacts" in benchmark NLI datasets (Gururangan et al., 2018; Poliak et al., 2018; Tsuchiya, 2018; Belinkov et al., 2019). These biases make it possible for models to infer the answer to the entailment problem from the hypothesis alone. Much like in the VQA setting, these "hypothesis-only" models appear to pick up on keywords in the input that reliably predict the answer. While many of these keywords are clearly associated with the corresponding answer class, for others, the link is much less obvious. In their analysis of the SNLI (Bowman et al., 2015) dataset, Poliak et al. (2018) found that universal negation words like "no," "nobody," "alone," and "empty" were strongly associated with contradictory hypotheses. However, the words "sleeping," "sleeps," and "asleep" were also found to be highly predictive of contradiction. By way of explanation, the authors noted that, since many SNLI premises deal with activities (e.g., "A woman is riding a bicycle"), an easy way for "lazy" crowd-

workers to construct a contradictory premise is to negate the agency of the subject (e.g., "A woman is sleeping") (Poliak et al., 2018). This example reveals how idiosyncrasies in the data collection process can give way to peculiar biases in ML benchmarks.

The literature on NLI offers useful methodological insights on how to improve model robustness when training on biased datasets. Belinkov et al. (2019) introduce a variant of the domain-adversarial technique that does not require training on examples from the target domain. In place of the domain classifier from Ganin et al. (2016), the authors use an "adversarial classifier" G_{H}, which attempts to infer entailment from the hypothesis alone. Successful performance by G_{H} is used as a regularization signal for the main network G_{NLI}. Specifically, G_{H} and G_{NLI} share a hypothesis encoder module f_H, and the two networks are trained in tandem. Gradients from G_{H} are backpropagated through a gradient reversal layer before entering f_H. This method discourages f_H from encoding biases or other artifacts that facilitate successful hypothesis-only NLI performance. Belinkov et al. (2019) demonstrate that adversarial training improves transfer performance on 9 out of 12 NLI datasets.

3.2.3 Application to VQA

While their study is limited to NLI, Belinkov et al. (2019) note that their method is broadly applicable to problems that require understanding the relationship between multiple different sources of input data. In this work, we apply this adversarial training scheme to VQA. While VQA incorporates images in addition to text, the structure of the task is analogous to NLI; here, the image corresponds to the premise, and the question corresponds to the hypothesis. At the time we began this

research, our work was, to our knowledge, the only existing attempt to employ adversarial regularization methods in a VQA setting. However, in October 2018, Ramakrishnan et al. (2018) published a paper demonstrating the successful application of adversarial training to VQA. These developments, which demonstrate the fast pace of VQA research, provide a useful point of comparison to our work.

3.3 Methods

3.3.1 Approach

We operationalize the adversarial training method from Belinkov et al. (2019) for bias removal in the VQA setting. We consider a base VQA model with the following four component modules, corresponding to the framework from Section 1.5:

- $f_v(I)$ Image feature extractor network
- $f_q(Q)$ Question feature extractor network
- $f_z(v, q)$ Multimodal fusion module
- $g_{\text{VQA}}(z)$ Answer classifier

Composing the four components, we obtain the following general expression for the base VQA model. This model is trained to minimize the cross entropy loss with respect to the ground truth answer a_{gt}.

$$P_{\text{VQA}}(\mathcal{A}|I, Q) = g_{\text{VQA}}(f_z(f_v(I), f_q(Q))) \tag{3.5}$$

$$L_{\text{VQA}} = -\frac{1}{|D|} \sum_{d \in D} \sum_{a \in \mathcal{A}} a_{gt}^{(d)} \log P_{\text{VQA}}(a|I, Q) \tag{3.6}$$

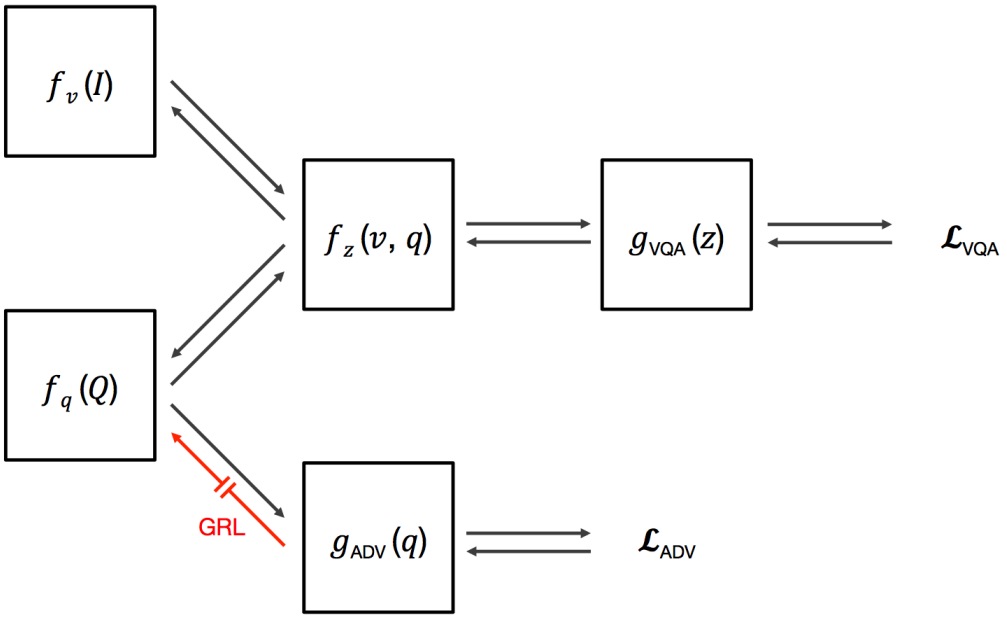


Figure 3.2: Schematic diagram of adversarial VQA architecture. Right and left arrows represent forward and backward propagation of gradients through the network, respectively. The red arrow indicates the gradient reversal layer.

We also introduce a question-only adversarial classifier $g_{\text{ADV}}(q)$, which attempts to infer the correct answer from only the question information. The adversary shares the same question feature extractor f_q as the base VQA model. However, f_q and $g_{\text{ADV}}(q)$ are separated by a gradient reversal layer (Eq. 3.3 and 3.4). As above, this model is trained to minimize cross entropy loss with respect to the ground truth answer.

$$P_{\text{ADV}}(\mathcal{A}|Q) = g_{\text{ADV}}(\text{GRL}_\lambda(f_q(Q))) \tag{3.7}$$

$$L_{\text{ADV}} = -\frac{1}{|D|}\sum_{d\in D}\sum_{a\in\mathcal{A}} a_{gt}^{(d)} \log P_{\text{ADV}}(a|Q) \tag{3.8}$$

We can now express the adversarial relationship between the main model and the adversary in a manner analogous to Eq. 3.2:

$$\min_{\text{VQA}} \max_{\text{ADV}} L = L_{\text{VQA}} - \lambda_{\text{ADV}} L_{\text{ADV}} \tag{3.9}$$

Here, the regularization coefficient $\lambda_{\text{ADV}} \geq 0$ controls the trade-off between performance on VQA and robustness to language bias. Additionally, the hyperparameter $\lambda_{\text{GRL}} \geq 0$ (from the gradient reversal layer in Eq. 3.7) controls the scaling factor of the gradient reversal. These two hyperparameters, which are the focus of our tuning experiments, perform related, but different functions. Setting either or both to zero disables the regularization, since f_q receives no gradients from the adversary. This combination is equivalent to the baseline VQA model. Meanwhile, setting $\lambda_{\text{ADV}} > 0, \lambda_{\text{GRL}} > 0$ enables adversarial regularization. This setting is the main focus of our experiments.

3.3.2 Implementation

Our experimental setup is implemented as an extension to the Bottom-Up / Top-Down model described in Section 1.6.2.[3] Unless otherwise noted, we use the default hyperparameters from Pythia. The adversarial classifier g_{ADV} is implemented as a two-layer fully-connected network with 512 hidden units and ReLU activation.[4] Both the adversary and the base VQA model are randomly initial-

[3] Code for this project is available at https://github.com/gabegrand/adversarial-vqa. Note that we base our work on the recent Pythia implementation of the Bottom-Up / Top-Down model from Facebook AI Research (Jiang et al., 2018a), as opposed to an earlier implementation that is popular on Github.

[4] As in Chapter 2, we experimented with different numbers of hidden layers $N = 1, 2, 3$ and hidden units $h = 256, 512, 1024, 2048$ in the adversarial classifier. We found the details of the adversary architecture to have little impact on performance, with the exception that adversaries with $N > 1$ hidden layers were more effective regularizers than one-layer adversaries.

ized with a fixed seed at the start of training. We co-train the networks for 16k iterations with two separate PyTorch Adamax optimizers with batch size 512 and learning rate 0.001. Note that Jiang et al. (2018b) use a handcrafted learning rate schedule, which consists of a 1k linear warm-up phase, after which point the learning rate is dropped by a factor of 0.1 at iterations 5k, 7k, 9k, and 11k. To minimize the possibility of gradient scaling mismatch between the base model and the adversary, we keep the learning rate fixed throughout training. While this modification causes the performance of the baseline VQA model to drop 1.1% points, it greatly improves stability and convergence during adversarial training.

The main goal of our experiments was to study the effects of adversarial regularization on our model's ability to generalize to new domains with different language priors. In our primary experiments, we train on the train split of VQA-CP v1 / v2, and evaluate generalization performance on the respective test split. By construction, the train and test splits have radically different language priors, so performance on VQA-CP test is a good measure of robustness to bias. As discussed in Section 3.1, regularization methods offer a trade-off between train and test accuracy. To assess this trade-off, we report the performance of our models on both splits of VQA-CP. Moreover, to understand how these models perform in their original context, we also retrain these models with the same hyperparameter settings on the regular VQA v1 and v2 datasets.

3.3.3 A Note on Validation in VQA-CP

One of the main methodological hurdles we encountered in working with VQA-CP relates to the issue of validation. In most machine learning settings, a small portion of the data is held out as a val-

idation set, which allows for an unbiased evaluation of the model's performance without running the model on the test set. Unfortunately, VQA-CP does not provide a validation set. While Agrawal et al. (2018) do not offer rationale for this decision, follow-up work co-authored by Agrawal notes, "VQA-CP does not have a validation set and generating such a split is complicated by the need for it to contain priors different from both the training and test sets in order to be an accurate estimate of generalization under changing priors – an ill-defined notion for binary questions" (Ramakrishnan et al., 2018). The authors explain that, in place of early stopping, they train their models "until convergence" on the training set.

Ramakrishnan et al. are correct in noting that the nonstandard structure of VQA-CP makes validation tricky. However, we take issue with the idea of training until convergence as an acceptable replacement for early stopping, for the reason that this practice is essentially guaranteed to result in overfitting for the baseline VQA model. Indeed, one of our first findings was that the baseline model quickly and severely overfits to the VQA-CP train set (see the top right plot in Figure 3.3 in the following section). This behavior is not surprising, since we expect that an unregularized model will overfit to the strong language priors in VQA-CP. However, it is important to note that, regardless of the biases inherent in the data, neural networks that are trained for a long time will tend to overfit to the specific examples contained in the training set. Indeed, as Figure 3.3 shows, train loss continues to decrease throughout the duration of training, reaching 90%+ accuracy, long after the test performance peaks. Therefore, if we wait until convergence, we are all but certain to end up with a baseline model that is unnecessarily overfit. This finding is problematic, since it suggests that training until convergence may artificially lower the performance of our baseline.

In order to address this issue, we must have some mechanism for distinguishing between overfitting to language priors in the training data, and overfitting to the training data more generally. Our solution is to randomly sample 10% of the examples in the VQA-CP train split to form a new VQA-CP validation split. In all our experiments on VQA-CP, we train models on the remaining 90% of the examples, and use the val split for early stopping. Performance on the val split will stop improving when the model begins to overfit to the specific training data. However, because the val split shares the same language biases as the train split, the val split cannot be used to assess how well the model will transfer to the test split. In this way, our methods correct for the issues associated with training until convergence, while preserving our ability to remain agnostic to the distribution of priors in the test set.

While dividing the trainset into train and val splits allows us to perform early stopping, some validation-related challenges still remain. In addition to early stopping, it is standard practice to use validation performance for model selection; i.e., choosing the best hyperparameter combination to run on the testset. However, since the new valset contains different biases from the testset, validation performance does not forecast how the model will perform on the testset. In the case of early stopping, we are content for the valset to be "blind" to test performance, since we care only about overfitting in the context of the training domain. In contrast, for model selection, we want to be able to identify regularization coefficients that facilitate good transfer performance. However, since regularization tends to reduce training accuracy, choosing models that perform well on the valset will tend to select for models that are underregularized.

We do not currently have a satisfying resolution to the issue of model selection on VQA-CP. As

Ramakrishnan et al. observe, a true validation set for VQA-CP would need to contain priors that are different from those in both train and test. However, the methods used by Agrawal et al. (2018) to construct the VQA-CP splits are fairly involved, and we do not have access to the underlying code. Moreover, even with the code, as Ramakrishnan et al. note, for questions with binary answers, it is unclear how we might achieve the notion of "changing priors" across three different splits. In the absence of a true validation set, one possible interim solution that we considered be to create a second valset derived from VQA-CP test. However, this approach compromises our ability to remain agnostic to the testset. Therefore, instead of introducing additional methodological complexity, we elect to bite the bullet; as in the work of Ramakrishnan et al. (2018), we perform model selection based on results on VQA-CP test. However, rather than selectively present our best-performing model, we report results across a broad range of hyperparameters. We hope that in the future, recognition of these challenges will prompt the introduction of a proper validation set for VQA-CP.

3.3.4 Experiments

We perform all of our experiments in parallel on VQA-CP v1 and VQA-CP v2. In adversarial regularization, the main experimental challenge is to identify a combination of λ_{ADV} and λ_{GRL} that achieves good regularization performance. As discussed in Section 3.3.1, the interaction of these two hyperparameters controls the strength of the regularization. Because of the novelty of this method, we have little prior insight as to what values to assign to these hyperparameters. Therefore, we perform a series of grid searches in order to hone in on the optimal values. We exhaustively report the test performance of all hyperparameter combinations on our grid search so as to maximize the trans-

parency of our results.

To improve gradient stability during the early stages of training, we experiment with a novel scheduling regime for the gradient reversal layer. GRL scheduling has two components: delay and warmup. During the first μ iterations of training, we set $\lambda_{\text{GRL}} = 0$, which allows the question encoder to receive clean gradients from the VQA model. Next, we have a warmup phase for w iterations, in which we increase λ_{GRL} linearly from 0 to some constant c. The following equation summarizes our GRL scheduling implementation, where t refers to the current training iteration:

$$\lambda_{\text{GRL}}(t) = \begin{cases} 0 & t \leq \mu \\ \frac{t-\mu}{w-\mu} & \mu \leq t \leq \mu + w \\ c & t > \mu + w \end{cases} \tag{3.10}$$

GRL scheduling introduces two new hyperparameters μ and w. We performed separate grid searches for VQA-CP v1 and v2 to identify optimal values for each dataset. We also report the results of these experiments in the following section.

3.4 Results

Adversarial regularization significantly reduces overfitting on VQA-CP. Figure 3.3 highlights the impact of adversarial regularization on model performance. Models that are not regularized exhibit characteristics of severe overfitting on both VQA-CP v1 val and test. Note that overfitting begins much earlier on the test set (around 2000 iterations) as the model begins to over-rely on language priors. In contrast, overfitting on the val set appears later, around 3500 iterations. This pattern holds for both VQA-CP v1 and v2.

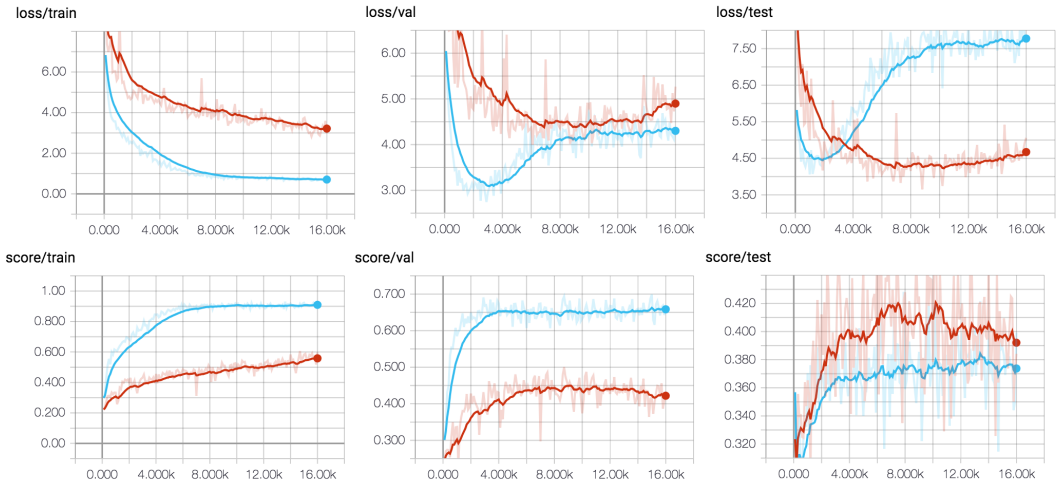


Figure 3.3: Performance comparison of [R]egularized (red) and [B]aseline (blue) models on VQA-CP v1 train, val, and test splits.[5] The baseline model exhibits severe overfitting on both the val and test splits. In contrast, in the regularized model, overfitting appears later in training, and is much less pronounced. Regularization bolsters the model's robustness to language biases, resulting in improved performance on the test set. However, these gains come at the cost of significantly reduced performance on the training domain.

[5] Note that, while Figure 3.3 shows loss on the set throughout the course of training, we assume this metric would be unavailable in a real-world setting. Therefore, we base early stopping only on the validation loss only.

Models (V1)	λ_{ADV}	λ_{GRL}	VQA-CP v1 (test)	VQA-CP v1 (val)	VQA v1 (val)
Baseline	0	0	37.87	**65.79**	**62.68**
+ AdvReg	0.1	0.01	**45.69**	46.94	46.34
Models (V2)	λ_{ADV}	λ_{GRL}	**VQA-CP v2 (test)**	**VQA-CP v2 (val)**	**VQA v2 (val)**
Baseline	0	0	38.80	**67.76**	**63.27**
+ AdvReg	0.005	1	36.33	50.63	48.78
+ GRL Sch	0.005	1	**42.33**	56.90	51.92

Table 3.1: Performance comparison of baseline and adversarially-trained models on VQA(-CP) v1 and v2 datasets using the best-performing hyperparameters. Adversarial regularization markedly increases performance on VQA-CP test, indicating improved generalization to out-of-domain examples. However, these gains come at the cost of substantially reduced performance on in-domain data on the VQA-CP and VQA validation sets.

Adversarial regularization improves test performance on VQA-CP v1. In general, adversarial regularization works well out-of-box on VQA-CP v1. Many of the hyperparameter combinations we tested (Figure 3.4) outperform the baseline on VQA-CP v1 test, with the strongest one improving on the baseline by 7.82% points (Table 3.1). The key to successful regularization appears to be balancing λ_{ADV} and λ_{GRL}. As Figure 3.4 reveals, large values of λ_{ADV} perform better with small values of λ_{GRL}, and vice-versa. However, when λ_{ADV} is too small, adversarial regularization fails to yield any performance improvements; none of the models we tested with $\lambda_{ADV} = 0.001$ outperformed the baseline. On the other hand, when λ_{ADV} is too large, training becomes unstable; for $\lambda_{ADV} > 1$, we observed that many training runs failed to converge due to exploding gradient values.

Out-of-box adversarial regularization fails to improve test performance on VQA-CP v2. As Figure 3.4 shows, none of the hyperparameter combinations we tested outperformed the baseline on VQA-CP v2 test. Section 3.5 features a discussion of this discrepancy and its potential causes. One

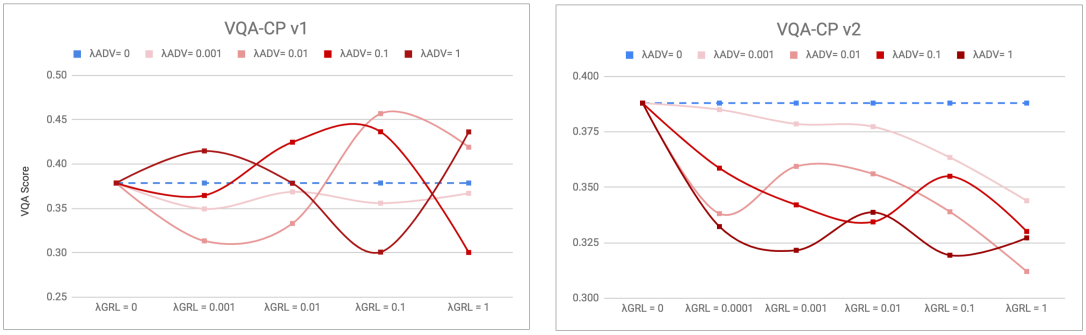


Figure 3.4: Hyperparameter tuning curves on VQA-CP v1 and v2 test. Each line represents a different setting of λ_{ADV}; lighter red indicates less regularization, while darker red indicates more. λ_{GRL} is varied along the x-axis. The blue dashed line shows the performance of the baseline model. On VQA-CP v1, many hyperparameter combinations outperform the baseline; these successful combinations tend to balance high values of λ_{ADV} with low values of λ_{GRL}, or vice versa. In contrast, on VQA-CP v2, none of the hyperparameter combinations tested outperform the baseline. For values of $\lambda_{\text{GRL}} > 1$, training diverges due to exploding gradients.

possible relates to the substantial amount of noise that the adversary inserts into the gradient updates for the question encoder. This phenomenon, which is readily observable by recording gradient norms throughout training, is most evident in the first 2000-4000 iterations, and is especially pronounced for VQA-CP v2 (see Figure 3.5). We hypothesized that this instability interferes with the early stages of optimization, causing training to converge to a suboptimal area of the global parameter space. In order to test this hypothesis, we ran a series of experiments with the GRL schedule described in Section 3.3.4, which yielded the following findings.

With GRL scheduling, adversarial regularization outperforms the VQA-CP v2 baseline. Figure 3.6 shows the results from the various schedules that were tested. For VQA-CP v2, several of these schedules improve performance over the baseline. In the highest-performing of these schedules, regularization is delayed until $\mu = 2000$ iterations, and slowly warms up for the following $w = 4000$ steps. This schedule results in a 6.00% point performance increase compared to using

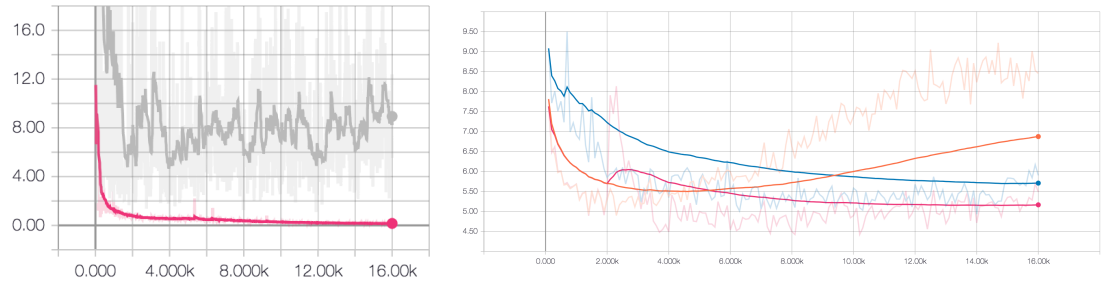


Figure 3.5: Left: Norm of gradient updates to the question encoder during adversarial training (gray) compared to baseline (magenta). Adversarial training introduces significant noise into the updates. Right: Comparison of loss plots illustrating how GRL scheduling helps to overcome gradient instability during training on VQA-CP v2. The baseline model (orange) begins to overfit around iteration 4000. Adversarial regularization (blue) helps to reduce overfitting, but the regularized model never recovers from the higher loss values early in training. When training under a GRL schedule (magenta), learning proceeds uninterrupted for the first μ iterations. When regularization is first enabled at $\mu = 2000$, there is a small bump in the loss curve. Nevertheless, the loss quickly recovers, and converges to a lower value than the baseline.

the same regularization coefficients without GRL scheduling, and a 3.53% point improvement over the baseline (see Table 3.1). Figure 3.5 helps to illustrate how GRL scheduling allows the model to converge to lower loss values during training.

GRL scheduling does not yield improvements on VQA-CP v1. Despite improving performance on VQA-CP v2, when applied to VQA-CP v1, we did not find any gains from GRL scheduling. Note that the baseline model begins to overfit roughly twice as quickly on VQA-CP v1 than on VQA-CP v2 (compare Figure 3.3 with 3.5); accordingly, we use accelerated GRL schedules for VQA-CP v1. Figure 3.6 shows the results of running adversarial training with various GRL schedules on VQA-CP v1. While five of the runs outperformed the baseline, three of these were with no start delay. Moreover, all of the runs with GRL scheduling performed worse than a model with the same regularization coefficients with static λ_{GRL}. Finally, many of the runs on VQA-CP v1, and especially those with fewer warm-up iterations, diverged due to exploding gradients.

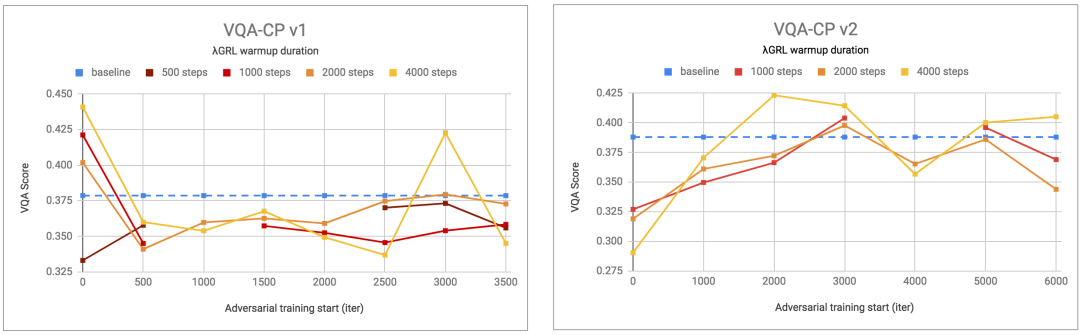


Figure 3.6: Effects of gradient reversal layer (GRL) schedules on performance on VQA-CP v1 and v2 test. Regularization is delayed until the start iteration μ, indicated on the x-axis, after which point λ_{GRL} is increased linearly from 0 to a constant value c over the course of w steps. Each line represents a different value of the warmup duration w. The blue dashed line shows the performance of the baseline model. Missing points indicate instances where training diverged due to exploding gradients.

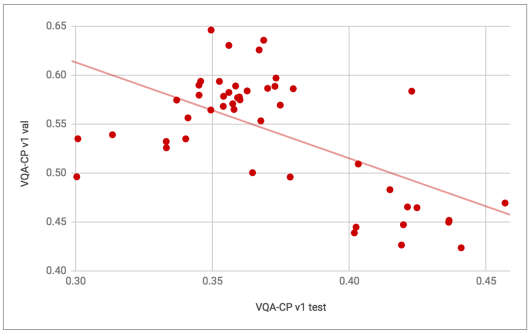


Figure 3.7: Trade-off between in-domain and out-of-domain performance on VQA-CP v1. Improvements on the test set were moderately correlated ($r^2 = 0.355, p = 0.014$) with reduced performance on the validation set.

Adversarial regularization consistently diminishes performance on the original VQA v1 and v2 datasets. Across all hyperparameter settings tested on VQA-CP v1, increases in test performance were moderately correlated ($r^2 = 0.355, p = 0.014$) with decreases in validation performance (see Figure 3.7). In general, score on the validation split of VQA-CP was a good predictor of performance on the original VQA datasets. Table 3.1 shows the results from the best-performing reg-

ularized models on VQA-CP test. Compared to the baseline, these models experienced significant reductions in performance when trained on the original VQA datasets (−16.34% points on VQA v1 and −11.35% points on VQA v2). Thus, while the gains due to regularization on VQA-CP test were more significant on v1 as compared to v2, the losses were also greater on the original versions of these datasets.

Adversarial regularization visibly reduces the presence of language priors in the posterior distribution. Figure 3.8 compares outputs of the baseline and regularized models for various question types. In all of these cases, the baseline model posterior shares significant overlap with the training prior. In contrast, the regularized model is able to correctly predict answers that have low prior probability in the training set. This effect is most apparent in questions with binary (i.e., yes/no) answers. In VQA-CP v1 train, the answer to more than 90% of the questions that begin with "Is there a...?" is "no," while the opposite is true on VQA-CP v1 test. Consequently, the baseline model almost always predicts "no" for questions in the test set of this type. In contrast, the regularized model answers roughly 71% of these questions with "yes." The regularized model also demonstrates the ability to make inferences that contradict real-world language priors; i.e., the regularized model correctly identifies the double-decker bus in Figure 3.8 as pink, while the baseline model thinks the bus is red.[6]

[6]In this case, red also happens to be the dominant color in the training prior.

Figure 3.8: Visualization of dataset priors and model scores for different question types on VQA-CP v1 test. In each example, the leftmost two bars show the prior distribution over answers for the given question type (in bold) for the train and test sets.7The rightmost two bars show the scores assigned to different answers by the baseline and regularized models for a particular example of the given type. The baseline model frequently assigns high probability to incorrect answers that are prominent in the training distribution. In contrast, the regularized model is able to make correct inferences in cases where the ground truth answer has low prior probability.

While qualitative inspection of model outputs finds many success cases for adversarial regularization, it also reveals a handful of idiosyncratic side-effects. Figure A.3 in the Appendix highlights the common failure modes of regularized models. Unsurprisingly, the regularized model struggles on questions that rely heavily on language priors (e.g., "What color is mustard?"). However, in many cases, the regularized model also ignores linguistic cues that constrain the expected form of the answer. For instance, when presented with image of horses on a beach, and the question "What animal is pictured?" the model's top answer was "beach." Salient visual features, such as the bright colors of a parrot's feathers, also appear to distract the model from the correct answer. Finally, in cases where the answer is difficult to deduce, the baseline tends to fall back on the language priors from the training set; meanwhile, the regularized model instead predicts visual features from the current image that may or not be relevant to the question at hand.

3.5 Discussion

Our findings provide clear support for adversarial regularization as a general method for reducing dependence on language priors in VQA models. Regularization leads to marked improvements on VQA-CP test for both v1 (+7.82% points) and v2 (+3.53% points), indicating improved generalization to environments with novel priors. On both these datasets, regularized models exhibit significantly less overfitting, as seen in the validation and test loss curves during training. Our use of

[7] In Figure 3.8, we show the VQA-CP test priors for different question types to facilitate comparison with the train priors. However, the reader should take care not to construe the test priors as "target" distributions for the models, since the baseline and regularized scores reflect model predictions *for the specific examples in the figure*. Moreover, since the models are not trained on the test domain, they have no means of learning the test priors.

separate validation sets on VQA-CP helps to establish that this phenomenon is specifically related to reduced overfitting to language priors. Furthermore, an inspection of the model outputs confirms that compared to the baseline model, regularized models rely much less on the strong language priors present in questions like "What sport is...?" and "Is there a...?" Additionally, their answers demonstrate more grounding in the image and its salient visual features. Taken as a whole, these results show clear promise for adversarial regularization as a method for improving the robustness of VQA models to latent biases in the data.

Though we find adversarial regularization to be effective in combating language bias, one key concern is that the pendulum may have swung too far: there are both qualitative and quantitative signs that our models may actually be over-regularized. Qualitatively, we observe that regularized models often ignore linguistic cues in the question, and are heavily swayed by salient visual features (see Figure A.3). These findings suggest an under-utilization of useful language priors as well as specific information in the question. Meanwhile, quantitatively, the performance of our regularized models suffers dramatically on in-domain examples, as seen through the substantial drops in score on the original VQA datasets.

In response to these results, it is natural to ask whether impaired in-domain performance is a necessary evil of adversarial regularization. Our aggregate correlation analysis on VQA-CP v1 (Figure 3.7) suggests that the trade-off between validation and test performance is pervasive across all hyperparameter settings tested in this work. This phenomenon makes intuitive sense, given that, by design, adversarial regularization censors information that is useful for in-domain inferences. Based on this evidence, we feel it is reasonable to expect that adversarial regularization will necessarily di-

minish performance on the original VQA datasets.

Despite these findings, it bears noting that in their work, which uses similar methods, Ramakrishnan et al. (2018) reported that their adversarially regularized VQA model only minimally underperformed their baseline model on VQA v2 (see Table 3.2 for a comparison their results with ours). In our comparison study, we attempted to replicate the methods of this study as closely as possible, despite lacking access to their code.[8] This included using the same hyperparameter settings and learning rate schedule. However, while both studies use the Bottom-Up / Top-Down VQA architecture from Anderson et al. (2018), we use the more modern Pythia implementation (Jiang et al., 2018a) in place of an earlier implementation that is popular on Github.[9] An unfortunate reality of ML research is that implementation details can sometimes have outsized impact on the effectiveness of a particular method. Therefore, is entirely possible that the difference in our findings is due to some small methodological discrepancy. Nevertheless, in order for adversarial regularization to gain broader adoption in the ML community, it is necessary that this technique not hinge on specific implementation details. Consequently, we feel our findings showing the steep costs of regularization on in-domain examples offers an important counter-narrative to the results presented by Ramakrishnan et al. (2018).

Setting aside these considerations, there is one interesting difference between our approach and that of Ramakrishnan et al. that may at least partially explain the deviation in our findings. In addi-

[8] At the time of writing, the code from Ramakrishnan et al. was not publicly available. We attempt to recapitulate the same hyperparameters and methods used in the study for comparison purposes. However, without looking at the code we cannot be sure that we are not missing some key implementation detail. Nevertheless, the authors indicated to us that they were planning to release this code in the future, at which point a replication of their exact methods will be possible.

[9] https://github.com/hengyuan-hu/bottom-up-attention-vqa

	Ours		Ramakrishnan et al.	
	VQA-CP v2 (test)	VQA v2 (val)	VQA-CP v2 (test)	VQA v2 (val)
Baseline	38.80	63.27	39.74	63.48
+AdvReg	36.33	48.78	40.08	60.53
+DoE	–	–	41.17	**62.75**
+GRL Sch	**42.33**	51.92	–	–

Table 3.2: Comparison of our results with those of Ramakrishnan et al. (2018) on VQA(-CP) v2.[10] Our best model outperforms that of Ramakrishnan et al. on VQA-CP v2 test by 1.16% points, despite starting at a lower baseline due our holding out 10% of the training examples. However, while Ramakrishnan et al. report only a minimal reduction (−0.73% points) in performance on VQA v2 due to regularization, we observe a substantially greater reduction (−11.35% points). Ramakrishnan et al. introduce an additional difference of entropies regularizer (denoted DoE) that improves both in-domain and out-of-domain accuracy.

tion to their adversarial regularizer, Ramakrishnan et al. introduce a second regularizer that considers the difference of entropies (DoE) between the output distributions of the main and adversarial models. Specifically, the DoE regularizer minimizes the entropy of the main VQA model's posterior distribution, while maximizing the entropy of the adversarial classifier's posterior distribution. Intuitively, this regularization technique seeks to ensure that the image, which only the main model has access to, significantly increases certainty about the answer. Indeed, Ramakrishnan et al. find that the addition of the DoE regularizer increases out-of-domain performance on VQA-CP v2 test by 1.09% points. Moreover, they also find that DoE increases in-domain performance on VQA v2 val by 2.22% points. In their discussion, the authors suggest that DoE helps the main model to retain discriminative information in the question encoding. In fact, they show that with DoE, λ_{ADV} can be increased to higher values without losing performance. These results suggest that DoE regularization or a similar technique may be useful for countering the negative effects of adversarial

[10] In Table 3.2, we compare on VQA(-CP) v2, since Ramakrishnan et al. do not report results for their Bottom-Up / Top-Down model on VQA(-CP) v1.

regularization.

While entropy difference regularization is an intriguing concept that merits further exploration, the need for a second regularizer to counter the original regularization technique is something of an oxymoron. Moreover, part of the appeal of adversarial regularization is its simplicity; thus, the introduction of yet another regularizer creates methodological complications that may hinder adoption of this technique. Our work offers an alternative solution in the form of GRL scheduling. GRL scheduling is a simple technique akin to other widely-employed forms of incremental scheduling in ML, such as learning rate decay and curriculum learning. Moreover, there is ample precedent in the literature that, when training on an auxiliary objective, it is necessary to gradually introduce this objective over the course of training. For instance, Ranzato et al. (2015) demonstrated that training sequence models on reward signals derived from non-differentiable test metrics (e.g., BLEU, ROUGE) using reinforcement learning improved performance over traditional cross entropy based training. However, they found that incremental integration of this secondary signal over the course of training was crucial to achieving performance gains; without scheduling, model performance did not improve beyond random chance (Ranzato et al., 2015). Since the underlying ideas are well-grounded in existing work—and will be instantly familiar to anyone with an ML background—GRL scheduling offers a practical contribution to the adversarial training literature.

In our case, given the gradient instability that adversarial training introduces, we found GRL scheduling to be a useful tool for improving the convergence properties of this technique. Indeed, we found that GRL scheduling was crucial for improving on the baseline score on VQA-CP v2 test. This result suggests that GRL scheduling counters the most acute negative effects of adversarial

regularization by allowing the main model to learn unencumbered early in training. Furthermore, because VQA v2 contains measurably less language bias than VQA v1, it is possible that adversarial regularization only becomes helpful later in training, when the model begins to overfit to these biases. Nevertheless, GRL scheduling is not the solution to all of the issues associated with adversarial training. In particular, even with GRL scheduling, we found that adversarial regularization significantly diminished performance on VQA v2. More research is necessary to form more principled theories of when and how GRL scheduling will be helpful. An interesting area for further exploration would be to develop an automated method of adjusting λ_{GRL} during training. This research idea, which is similar to the adaptive learning rate techniques widely in use in ML today (Zeiler, 2012; Kingma & Ba, 2014), has the potential to lead to a more theoretically-grounded understanding of adversarial training methods.

Regardless of whether we use GRL scheduling, difference of entropy, or some other technique, it is clear that the tendency of adversarial training to over-regularize is a significant drawback of this method. In particular, our qualitative inspection of common failure modes (see Figure A.3) reveals that adversarially-regularized models tend to ignore important linguistic cues in the question. These findings suggest the need to develop more targeted regularization techniques. Our current approach regularizes the entire question representation v. One possible improvement would be to use an attention-weighting scheme to apply different amounts of regularization to different words in the question. In this way, regularization could be focused only on the first few words of the question (e.g., "Is there a...") that encode answer-distribution biases, while preserving other important linguistic information, such as common knowledge.

The poor interpretability/explainability of deep neural networks and other state-of-the-art AI techniques is primarily due to cultural factors, rather than technical limitations. We can and should fix this by refocusing our research priorities as a community.

—Arvind Narayanan

Professor of Computer Science, Princeton University

4

Conclusion

The current moment in AI research is one of collective reckoning. The wildly-successful marriage of black box inference models and big data has allowed us to broach problems that seemed intractable just a decade ago. However, this pairing has given rise to a unique set of headaches that researchers are only now beginning to confront. The brief history of VQA research perfectly encapsulates the dawning of this realization. The early VQA literature from 2014-2016 was characterized by a certain gung-ho optimism: simply apply existing methods from computer vision and natural

language processing, combine, and "voilà!" Over the following years, however, researchers seeking to interpret the behavior of these models began to uncover the extent to which they internalize spurious patterns in the data. In this way, the twin problems of interpretability and bias have forced VQA researchers to appreciate the limitations of our current end-to-end learning approaches.

Though VQA research currently faces many hurdles, perhaps the greatest obstacle is not technical, but cultural. The majority of VQA research is aimed at improving accuracy on big-ticket benchmarks, chief among these being the VQA Challenge datasets discussed in this work. While competitive benchmarks offer a convenient way for researchers to compare performance metrics, they also cause VQA research as a whole to "overfit" to these particular datasets. As ML bias researcher Arvind Narayanan puts it:

> ML has an "accuracy fetish" — comparing algorithms and models based on a single performance number on a benchmark dataset, such as ImageNet. This has certainly had benefits and led to rapid progress in some areas. But when the whole field is engaged in one-dimensional, competitive pursuits, it becomes culturally hard to work on mitigating bias. And if the benchmark datasets we use themselves encode our historic biases, only biased models can "win".[1]

As Narayanan argues, these cultural obstacles can be overcome by "refocusing our research priorities as a community" to promote work that addresses issues of interpretability and bias in ML. Indeed, since 2016, a growing group of researchers have begun to tackle these questions in VQA. These efforts, which dovetail with the work presented in this thesis, offer exciting glimpses of a future generation of models capable of offering both accurate and transparent answers to multimodal questions.

[1]Narayanan, Arvind (@random_walker), *Twitter*, Dec. 21, 2017. https://twitter.com/random_walker/status/943922485594075136

4.1 Towards Interpretable VQA

One of the principal objectives of interpretable ML research is to create models capable of explaining their internal reasoning processes. This goal requires translating a model's internal representations into output that can be understood by a human. Pointing to counterexamples is one way for a model to provide clues about its reasoning. Ultimately, however, we would like for models to be able to divulge their entire reasoning process from start to finish. Unfortunately, the relative simplicity of many naturalistic VQA queries allows models to make educated guesses without taking into account the physical and conceptual relationships between objects in the image. For this reason, VQA models can often get by with pattern-matching between words in the question and visual features in the image. The absence of structured reasoning in current VQA models presents a significant obstacle to the goal of providing human-interpretable explanations.

One method of cracking down on this kind of behavior is to constrain the VQA task to require more advanced forms of compositional reasoning. Synthetic VQA datasets like SHAPES (Andreas et al., 2016) and CLEVR (Johnson et al., 2017) replace natural scenes with procedurally-rendered images of geometric objects with different spatial orientations and physical properties. By using a functional-style programming language to generate both questions and images, this approach allows for complex, hierarchical questions; e.g, "There is a sphere with the same size as the metal cube; is it made of the same material as the small red sphere?" (Johnson et al., 2017) These benchmarks have given rise to new class of VQA architectures called Neural Module Networks (NMNs) (Andreas et al., 2016; Hu et al., 2017; Johnson et al., 2017; Mascharka et al., 2018). NMNs consist of a set of

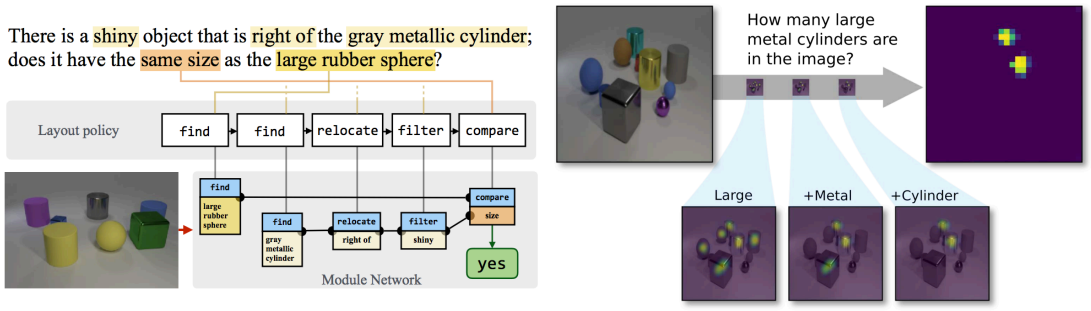


Figure 4.1: Examples of neural module networks, which segment reasoning into a sequence of human-interpretable operations. Left: End-to-end Module Networks generate instance-specific network structures composed of learnable modules. (Figure from Hu et al. 2017). Right: Transparency by Design networks utilize a set of visual reasoning primitives that produce human-interpretable attention maps, offering useful insight into the model's reasoning process. (Figure from Mascharka et al. 2018).

modules that learn to perform discrete visual reasoning tasks (e.g., `find`, `relocate`, `and`, `or`, `filter`, `count`, etc.). During inference, a generator network composes a sequence of modules on-the-fly to answer the question (see Figure 4.2). In this way, NMNs provide significantly more insight into their internal reasoning processes than black-box neural models. Moreover, NMNs have been successfully extended from synthetic datasets to naturalistic VQA, though they do not perform as well as black-box models on the latter (Andreas et al., 2016; Chandu et al., 2018).

Another exciting approach to interpretable VQA leverages graph-based representations to facilitate novel forms of inference. Much of this work utilizes Graph Convolutional Networks (GCNs; Kipf & Welling 2016), which have recently emerged as a popular method of knowledge mining for graph-structured data. Since the early days of VQA, an interesting sub-literature has focused on the use of relational databases to augment VQA models with external knowledge (Wang et al., 2015; Wu et al., 2016; Teney et al., 2017b; Lu et al., 2018). Recently, Narasimhan et al. (2018) introduced a VQA model that utilizes GCNs to support its inferences with human-interpretable relational facts mined

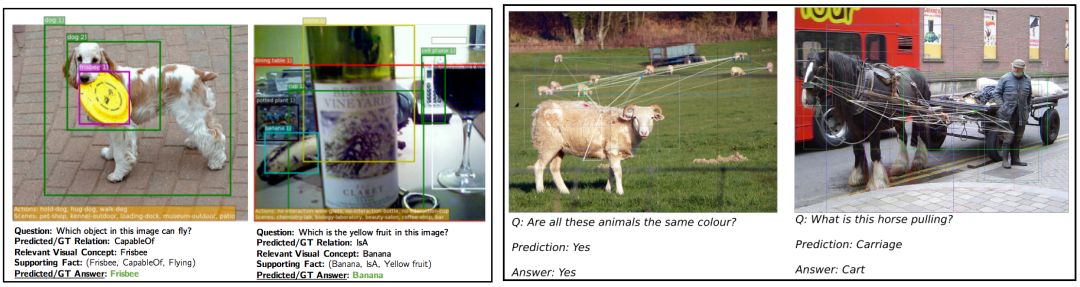


Figure 4.2: Graph Convolutional Networks (GCNs) facilitate novel forms of interpretable inference in VQA. Left: GCNs allow models to support their inferences with relational data stored as knowledge graphs. (Figure from Narasimhan et al. 2018) Right: GCNs learn complex, query-dependent relationships between detected objects in the image. (Figure from Norcliffe-Brown et al. 2018).

from a knowledge graph. Taking this idea a step further, Norcliffe-Brown et al. (2018) noted that objects in the image also implicitly define a relational graph. Based on this insight, they proposed an approach that uses GCNs to learn complex, query-dependent relationships between the objects in images, without the aid of an external database. As both of these approaches demonstrate, graph-based methods offer interesting and creative ways to construct more interpretable VQA models.

4.2 Towards Bias-Free VQA

With respect to the Problem of Bias, there is much cause for optimism. Growing awareness of bias-related issues in the ML community has spawned a broad range of solutions. As the work presented in Chapter 3 demonstrates, adversarial regularization is an effective technique for mitigating learned bias in VQA models, with many promising avenues for further research. While our work puts forward one method for dealing with bias in models, it will ultimately fall on the VQA community to address these issues at the dataset level. In particular, in light of our finding that adversarial regularization reduces in-domain accuracy, it is not likely that researchers will sacrifice precious points to

reduce bias unless the benchmark itself requires it. In other words, in order to solve the Problem of Bias, unbiased learning must be the dominant strategy on competitive VQA benchmarks.

To this end, it is useful to consider how we might go about designing a next-generation VQA v3 dataset for the VQA Challenge. Our work on the existing VQA Challenge datasets offers a number of insights.

- **If you can't beat 'em, join 'em.** As discussed in Section 1.3, it is neither realistic nor desirable for a real-world VQA dataset to be perfectly "unbiased"; i.e., to contain uniform distributions over answer classes. Instead, VQA v3 should explicitly score models on their ability to generalize to new domains with different priors. This "generalization score" could be measured on an additional VQA-CP style test set with priors that differ from the main training/testing domain.
- **Provide standard methods for evaluating generalization performance.** As we note in Section 3.3.3, the lack of a proper validation set on VQA-CP makes it difficult to perform model selection in a principled manner. One solution would be to create a validation set that contains priors that are different from both the train and test sets. However, as Ramakrishnan et al. (2018) observe, the existence of binary questions complicates this prospect. Instead, we suggest a method similar to that of Agrawal et al. (2018), who in their paper introducing VQA-CP note that they actually created four different sets of VQA-CP v2 splits using different random seeds (however, they only make one of these splits publicly available). By reserving a subset of these splits for test, and making the rest available for development, VQA v3 could provide researchers with a convenient way of evaluating generalization performance without using the test set.
- **Encourage submission of question-only and image-only baselines.** For the 2018 VQA Challenge, the organizers submitted question-only and prior-only baselines to the competition leaderboard.[2] While this is a start, these baselines tend to get buried at the bottom of the leaderboard as submissions with more powerful models accumulate. Therefore, competitors should also be asked to submit question-only and image-only baselines for their own models. To facilitate this best-practice, VQA v3 should provide unique test splits in which the

[2] http://www.visualqa.org/roe.html

questions and images, respectively, are omitted. Researchers could then run their models on these unimodal splits to obtain baseline scores. Due to the difficulty of verifying what data a model is run on, submission of these baseline scores should be optional, but encouraged.

- **Switch to multiple choice answer format.** In contrast to open-ended VQA, which places no constraints on the possible answer set, multiple choice VQA offers a fixed number of answer choices for each example. Many popular VQA datasets, including COCO-VQA (Ren et al., 2015a), Visual7w (Zhu et al., 2016), and Visual Genome (Krishna et al., 2017), are multiple choice format. (In VQA v1, Antol et al. (2015) also included a multiple choice variant, but this less-popular format was abandoned in VQA v2.) Multiple choice format promises to help dataset authors reduce bias by facilitating fine-grained control over the statistical properties of the answer distribution. In their recent paper titled, "Being Negative but Constructively: Lessons Learnt from Creating Better Visual Question Answering Datasets," Chao et al. (2017) demonstrate that careful selection of the decoy (non-target) answers in multiple choice VQA can dramatically reduce dataset biases. In particular, they argue that the decoys should be both question-only and image-only unresolvable, meaning that the answer cannot be inferred from the question or image alone. To enforce these constraints, VQA v3 should involve a validation crowd test that screens for examples where humans are able to resolve the answer without access to either the question or image. Additionally, in accordance with the "neutrality" principle described by Chao et al. (2017), all possible answers to a given example in VQA v3 should be equally likely *a priori*. Enforcing this global constraint is only possible with multiple choice answers.

- **Treat answers as inputs, not outputs.** While VQA v1 and v2 are theoretically open-ended, as discussed in Section 1.5.4, the vast majority of competitive models adopt a classification approach that considers only the top several thousand answers. This practice, which disregards the low-frequency answers that make up the majority of the answer vocabulary, results in brittle models that depend on the distributional statistics of the training domain. An elegant solution is to include the answer in the input; i.e., feed tuples (I, Q, A_1), (I, Q, A_2), ... (I, Q, A_N) as inputs, and have the model assign likelihood scores as outputs. This approach, which was first introduced in Jabri et al. (2016), means that any answer in the vocabulary, including low-frequency ones, can be scored by the model. Furthermore, this method would allow the model to score unseen answers through open-vocabulary techniques that are well-documented in the neural machine translation literature (Luong & Manning, 2016; Zhao et al., 2018a). Because of the impracticality of scoring every possible answer in the vocabulary,

this approach is best-suited to the multiple choice format. Technically speaking, the decision to treat answers as inputs is a property of the model design, not the dataset. However, by explicitly marking the answers as "inputs," VQA v3 could induce a strong prior on model designs to encourage this practice.

- **Weight score by inverse answer frequency.** Even with the improvements suggested above, any skew in the answer frequency distribution will bias models towards more frequent answers. To counter this effect, VQA v3 should weight example scores by inverse answer frequency. In this way, VQA v3 can encourage robust performance on low-frequency answers.
- **Source examples from geographically-diverse areas.** In their paper, "No Classification without Representation: Assessing Geodiversity Issues in Open Data Sets for the Developing World," Shankar et al. (2017) demonstrate that ImageNet and Open Images contain Western-centric biases that harm classification performance of models deployed to the developing world. Their introduction of the Google Inclusive Images Dataset[3] offers an exemplary standard for promoting culturally equitable ML. In VQA v3, it is important to source both images and questions from geographically heterogeneous areas to reduce cultural bias, and encourage the development of VQA models that work for everyone.

By enacting these reforms, the authors of the next generation of VQA datasets have the opportunity to shape the direction of the future of VQA research. Given that scientific progress occurs in a cultural context, dataset designers have both a scientific and moral imperative to treat bias as a first-class consideration. As legal scholar Cass Sunstein argues, "A choice architect has the responsibility for organizing the context in which people make decisions... The first misconception is that it is possible to avoid influencing people's choices." (Sunstein, 2014). While Sunstein's comments refer to the design of social policy, they apply equally well to the design of ML benchmarks. Indeed, if VQA is to fill the shoes of a general-purpose test of machine intelligence, then we must embrace our role as policymakers in the design of fair and equitable benchmarks.

[3] https://ai.googleblog.com/2018/09/introducing-inclusive-images-competition.html

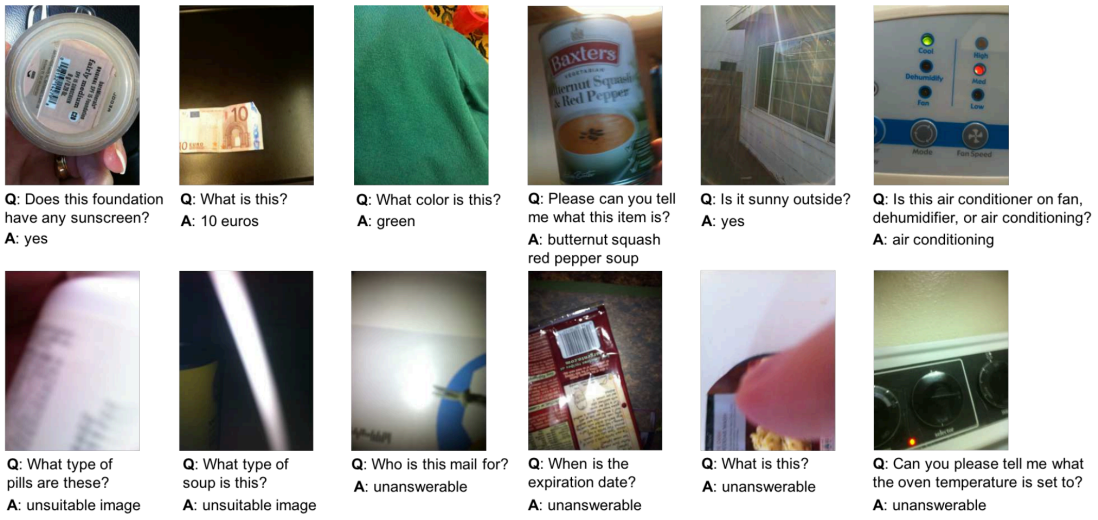


Figure 4.3: Example queries from visually impaired users in the VizWiz Grand Challenge dataset. VizWiz models must either answer the query (top row), or determine that it is unresolvable (bottom row). (Figure from Gurari et al., 2018)

4.3 VQA in the Real World

In this thesis, we have considered VQA in the context of pre-generated text queries for static images. However, the field of VQA is rapidly expanding in directions that push at the boundaries of the existing task formulation. This final section highlights a few exciting areas of research that promise to redefine visual question answering in the coming years.

One of the main areas of practical application for VQA systems is in assisting the visually impaired. In 2010, Bigham et al. (2010) proposed a mobile app called VizWiz that allowed blind users to snap photos and receive answers to queries from online crowd workers. While this innovative system was effective in helping blind users understand visual aspects of their surroundings, it often required users to wait minutes before receiving a response from crowd workers. With the advent of

VQA systems powered by ML, it is now becoming possible for this task to be automated, allowing instantaneous response times. In the coming years, the recently-announced VizWiz Grand Challenge (Gurari et al., 2018) promises to catalyze significant progress towards building VQA systems for visually impaired users. However, in order for VQA systems to function in an assistive capacity, they must be able to unpack their reasoning to users. Telling a user that they are holding a 10 Euro note is a clear value-add, but only if the system can explain that it spotted a "10" on the bill. Moreover, users must be confident that these systems are basing their judgments in reality; when asked, "Are these my blue pills or my red pills?" a reliance on color priors spells disaster.

With VizWiz, real-time VQA has now begun to enter into the realm of technological possibility. However, while responses from automated VQA systems may be instantaneous, they are still frozen in a particular instant in time. There is a growing consensus that VQA on static images fails to capture the dynamic, changing nature of everyday situations. Recently, a handful of researchers have started to take on the ambitious task of *video* question answering (Zeng et al., 2017; Ye et al., 2017; Lei et al., 2018; Zhao et al., 2018b). Generally speaking, video QA systems seek to answer questions about short video segments (generally, 10 - 180s, though Zhao et al. (2018b) consider clips up to 10 minutes in length).

Because the domain is so new, there is no single established video QA benchmark; instead, researchers have tended to create their own video QA datasets on a per-paper basis. While these circumstances have promoted use of a diversity of video content for research, leaving content selection up to individual researchers threatens to exacerbate issues of bias. In their work on video QA, Lei et al. (2018) introduce a dataset called TVQA that is composed of clips from six popular tele-

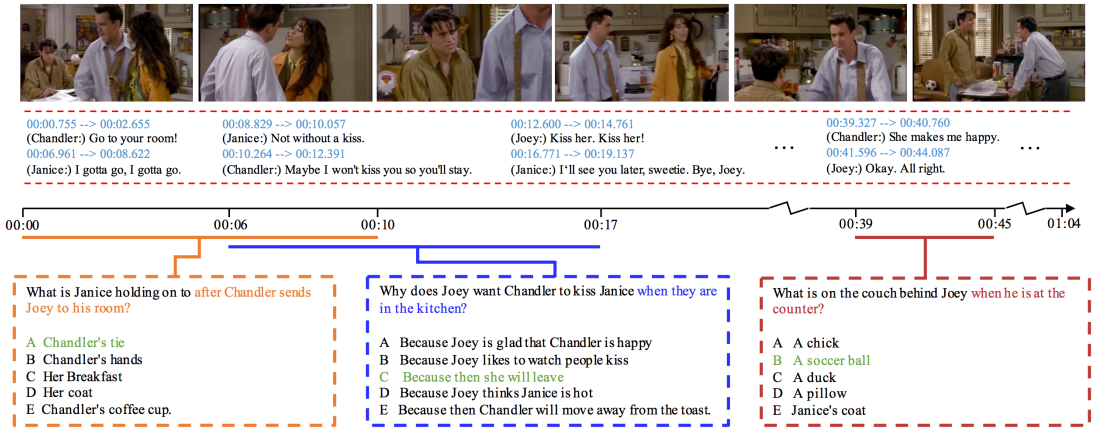


Figure 4.4: Example from the TVQA dataset containing questions about a one-minute clip from *Friends*. Sitcoms, which capture myriad sociocultural biases, are uncharted territory for machine learning. (Figure from Lei et al., 2018)

vision shows (*The Big Bang Theory*, *How I Met Your Mother*, *Friends*, *Grey's Anatomy*, *House*, and *Castle*). For decades, media scholars have studied how questions of bias and representation in popular film and television have affected the perception of marginalized groups in society (Rawles, 1975; Bechdel, 1986; Hamamoto, 1994; Benshoff, 2009; Holtzman & Sharpe, 2014; Mastro, 2017). While these issues appear far removed from the day-to-day concerns of computer scientists, further work on video QA means that a collision of worlds is imminent. In TVQA, for example, the "top unique nouns" associated with Bernadette, a female scientist on *The Big Bang Theory* are: `song, sweater, wedding, child, husband, everyone, necklace, stripper, weekend, airport`; meanwhile, top nouns associated with other female characters include `boyfriend, sex, bathroom, pink, ring, purse, dress, kitchen`, and `bedroom` (Lei et al., 2018). As a discipline, computer science is not well-equipped to handle issues of intersectionality. To wade further into this quagmire without adequate preparation is both irre-

sponsible and unnecessary. By educating themselves about the issues of bias, and consulting with peers in the social sciences, ML researchers can avoid introducing accidental biases into video QA datasets.

Work on video QA represents an attempt to bring multimodal reasoning into the realm of the real world by considering dynamic scenes that change over time. However, in video QA, the AI system still remains a passive observer to the scenes that unfold in front of it. Moreover, as discussed, real-world footage is often laden with sociocultural biases that are difficult to control for. Amusingly, the concerns with video QA are reminiscent of the kinds of concerns parents often raise about kids watching too much TV. In this scenario, the classic directive to "go outside and play" applies equally well to VQA. Recent years have seen the the emergence of 'interactive" or "embodied" VQA tasks that challenge agents to navigate and answer questions in simulated environments (Gordon et al., 2017; Das et al., 2018). Because these tasks involve non-trivial path planning, approaches typically integrate reinforcement learning techniques with existing vision and language models. In this way, this new generation of embodied QA (EQA) benchmarks offer a new way for AI researchers to explore the "embodiment hypothesis," which is "the idea that intelligence emerges in the interaction of an agent with an environment and as a result of sensorimotor activity" (Smith & Gasser, 2005).

Since learning in EQA occurs in simulated environments, researchers have full control over the kinds of biases that are available for models to learn from. While this scenario compares favorably to video QA, there is still the potential for latent biases to artificially inflate our estimation of model capabilities. Recent work by Thomason et al. (2018) shows that image-only and question-only baselines significantly outperform the published majority class baseline that accompanies the EQA task

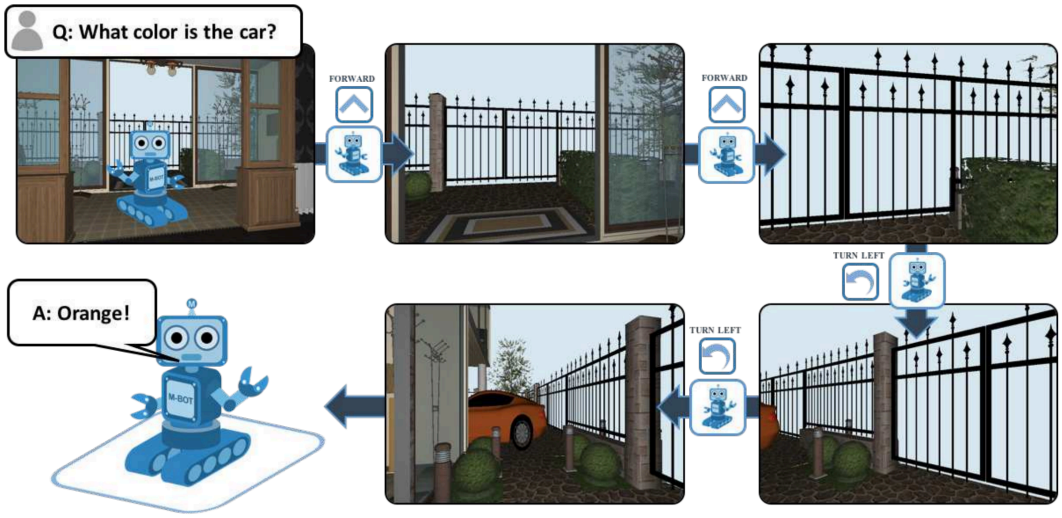


Figure 4.5: In Embodied QA, an agent must navigate through a simulated environment in order to answer a query. (Figure from Das et al., 2018)

(Das et al., 2018). The image-only baseline succeeds at identifying salient colors and objects that allow it to make educated guesses about the answer in the absence of the question. Meanwhile, the question-only baseline is able to exploit world knowledge priors to produce correct answers nearly 50% of the time (e.g., the answer to "What color is the bathtub?" is almost always gray). These baselines paint the achievements of existing EQA models in a much less flattering light. Das et al. (2018) report that their EQA model outperforms the majority class baseline by 44.2%. However, when considered against the unimodal baseline from Thomason et al. (2018), this margin shrinks to 15.2%. These stark findings highlight the need for broader recognition of the Problem of Bias across the spectrum of VQA research, as well as adoption of best practices that serve to curb it.

As VQA systems graduate to the real world, the issues discussed in this thesis concerning interpretability and bias become of foremost importance. If we are to trust these systems to assist the

blind, analyze visual media, and navigate in the physical world—to operate in our reality—then we must be certain that their reasoning is, in fact, grounded in reality. Given the media's growing attention to AI, failure to take this message to heart risks misleading the public about the capabilities of current methods. Such a scenario is reminiscent of an incident that occurred in the early twentieth century involving Clever Hans, a horse that was professed to be capable of arithmetic, reading, and other cognitive feats. After much spectacle, it was revealed that the Clever Hans had simply learned to pick up on spurious cues based on the body language of his trainer (Pfungst, 1911). Today's trainers of ML models can learn a lesson from this episode. In the face of a remarkable display of intelligence, we must retain a healthy sense of scientific skepticism. To this end, it is helpful to remember the wisdom of another clever Hans, who first noted the paradoxical elusiveness of everyday intelligence. For the forseeable future, at least, visually-grounded reasoning remains what Dr. Hans Moravec would call, "a hard problem."

A

Appendix

A.1 Glossary of Notation

VQA Datasets	
I	Image
Q	Question
A	Answer
$d_i = (I, Q, A)$	VQA example
$D = \{d_1, \ldots, d_N\}$	VQA dataset
Counterexample Prediction	
$I_{\text{NN}} = \{I'_1, \ldots, I'_K\}$	k-nearest neighbors to image I
$\{I', A'\}$	Candidate counterexample
$\{I^*, A^*\}$	Ground truth counterexample
$d_i = (I_i, Q_i, A_i, I'_i, A'_i)$	VQA-CX example
VQA Model Components	
$P(\mathcal{A}\|I, Q) = \text{VQA}(I, Q)$	Overall VQA model posterior
$v = f_v(I)$	Image embedding
$q = f_q(Q)$	Question embedding
$z = f_z(v, q)$	Multimodal embedding
$P(\mathcal{A}\|I, Q) = g_{\text{VQA}}(z)$	Answer classification
Adversarial Training	
$P(\mathcal{A}\|Q) = g_{\text{ADV}}(q)$	Adversarial classification
λ_{ADV}	Regularization coefficient for adversarial loss
λ_{GRL}	Gradient reversal layer scaling factor

A.2 Visualizations

Figure A.1: Qualitative results on the counterexample prediction task. Left: The original image and ground truth counterexample from VQA v2, along with the question and ground truth answers. Right: the top 5 counterexamples selected by NeuralCX, with the top 3 answers (as scored by a VQA model) underneath. In the top 4 rows, NeuralCX correctly identifies the correct counterexample (green outline), while in the bottom 4 rows, it fails. See Section 2.5 for a discussion of common failure modes.

Figure A.2: Visualization of dataset priors and model scores for different question types on VQA-CP v1 test. In each example, the leftmost two bars show the prior distribution over answers for the given question type (in bold) for the train and test sets. The rightmost two bars show the scores assigned to different answers by the baseline and regularized models for a particular example of the given type. The baseline model frequently assigns high probability to incorrect answers that are prominent in the training distribution. In contrast, the regularized model is able to make correct inferences in cases where the ground truth answer has low prior probability.

Figure A.3: Common failure modes of adversarial regularization. First row: the regularized model fails to infer the correct form of the answer from the question, answering "beach" and "wedding" to questions that entail animal answers. Second row: the regularized model struggles with questions that rely on real-world language priors; i.e., mustard is yellow, sunset is orange. Third row: salient colors in the image distract the regularized model from attending to the correct image regions. Fourth row: both the baseline and regularized models perform poorly on questions where the answer relates to a localized image region (i.e., inside a TV) as opposed to the global image. In these cases, the regularized model relies on generic visual features in the image in its inferences, while the baseline model relies on language priors.

Q: What kind of containers are holding the oranges?

Original
A: baskets

Counterexample
A: bowls

(10.25) glass
(9.62) plastic
(8.62) basket

(8.37) plastic
(8.13) glass
(7.56) basket

(10.77) plastic
(10.40) glass
(8.16) basket

(9.64) basket
(9.18) plastic
(8.83) glass

(10.03) glass
(9.77) plastic
(8.32) basket

Q: What kind of animals are sitting in the grass?

Original
A: cows

Counterexample
A: zebras

(13.30) zebra
(12.08) zebras
(8.32) giraffe

(15.84) sheep
(12.10) goats
(12.10) cows

(14.64) giraffe
(14.52) giraffes
(8.27) zebra

(12.91) giraffes
(12.73) giraffe
(7.25) cows

(11.75) giraffe
(11.27) giraffes
(11.25) zebra

Q: What sport is the man participating in?

Original
A: snowboarding

Counterexample
A: surfing

(14.35) skiing
(12.49) snowboarding
(9.98) ski

(16.41) surfing
(10.90) surf
(9.74) kayaking

(16.59) skiing
(12.68) ski
(12.42) snowboarding

(13.97) skiing
(12.76) snowboarding
(10.31) ski

(14.01) skiing
(13.30) snowboarding
(10.15) ski

Q: How many people are on the motorcycle?

Original
A: 2

Counterexample
A: 1

(15.01) 1
(14.24) 2
(12.13) 0

(14.33) 1
(13.47) 2
(11.37) 3

(13.72) 2
(13.22) 1
(11.88) 3

(13.92) 1
(13.75) 2
(11.78) 3

(13.19) 1
(13.00) 2
(11.36) 3

Q: What is the second word on the white sign?

Original
A: seat

Counterexample
A: across

(7.68) parking
(7.33) stop
(7.07) street

(9.07) stop
(6.33) one way
(6.20) parking

(8.03) stop
(7.33) street
(7.04) parking

(7.85) stop
(6.70) parking
(6.46) street

(7.20) parking
(7.14) stop
(6.68) street

Q: Is this plane equipped with cruise missiles?

Original
A: no

Counterexample
A: yes

(14.54) yes
(13.10) no
(6.38) unknown

(14.64) yes
(13.20) no
(6.41) unknown

(14.44) yes
(13.53) no
(6.63) unknown

(14.06) yes
(13.09) no
(6.15) unknown

(14.54) yes
(13.46) no
(6.64) unknown

Q: Is the toilet lid down?

Original
A: no

Counterexample
A: yes

(13.21) yes
(13.15) no
(5.34) unknown

(13.54) no
(13.13) yes
(5.69) up

(13.09) no
(12.90) yes
(5.31) unknown

(13.83) yes
(13.48) no
(6.11) up

(13.03) yes
(12.63) no
(5.40) up

Q: What is the girl doing?

Original
A: playing game

Counterexample
A: cooking

(12.41) smiling
(11.27) eating
(10.07) cutting cake

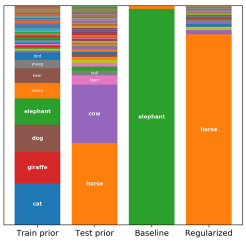
(10.98) sitting
(10.03) playing
(8.81) playing wii

(10.97) cutting cake
(10.34) smiling
(9.63) eating

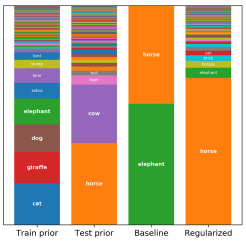
(10.37) playing
(9.61) sitting
(9.20) standing

(11.28) playing
(10.39) playing wii
(9.20) reading

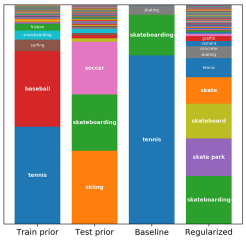
Q: **what animal is** used?
A: horse



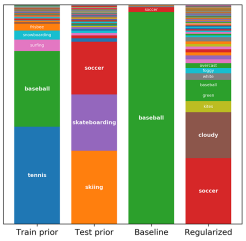
Q: **what animal is** shown?
A: horse



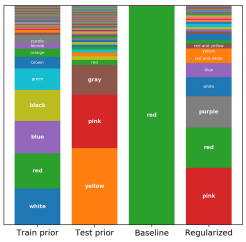
Q: **what sport is** being played?
A: skateboarding



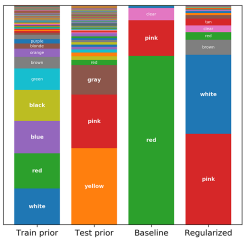
Q: **what sport is** being played?
A: soccer



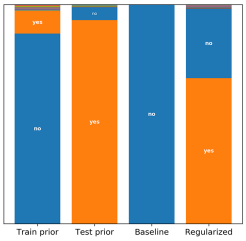
Q: **what color is** this double Decker bus?
A: pink



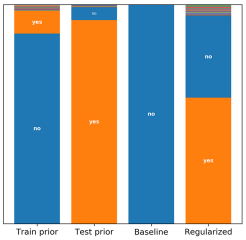
Q: **what color is** hydrant?
A: pink



Q: **is there a** painting?
A: yes

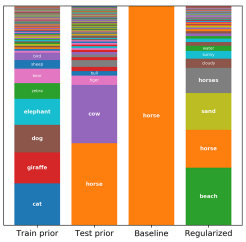


Q: **is there a** bridge in the picture?
A: yes

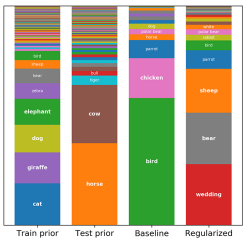


Regularized model fails to infer the correct form of the answer.

Q: **what animal is** pictured?
A: horses

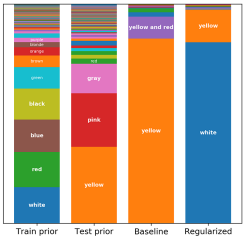


Q: **what animal is** this?
A: parrot

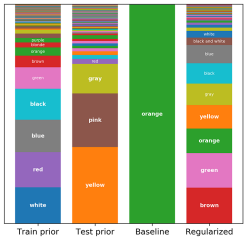


Regularized model fails to utilize real-world language priors.

Q: **what color is** mustard?
A: yellow

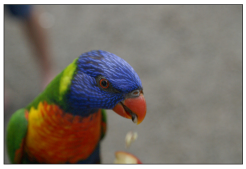


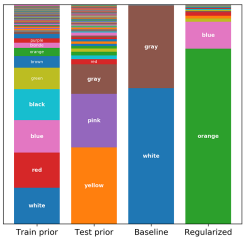
Q: **what color is** sunset?
A: orange



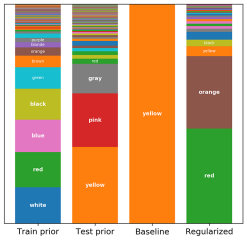
Regularized model distracted by visually-salient image features.

Q: **what color is** in the background of this photo?
A: gray



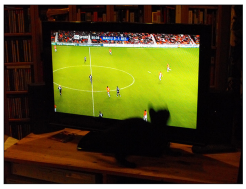


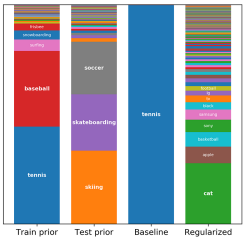
Q: **what color is** his helmet?
A: yellow



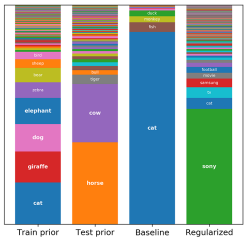
Regularized model relies on image features, while baseline model relies on language priors.

Q: **what sport is** being played on the television?
A: soccer





Q: **what animal is** on the TV?
A: geese



References

Agarwal, B. & Mittal, N. (2016). Machine learning approach for sentiment analysis. In *Prominent feature extraction for sentiment analysis* (pp. 21–45). Springer.

Agrawal, A., Batra, D., & Parikh, D. (2016). Analyzing the behavior of visual question answering models. *arXiv preprint arXiv:1606.07356*.

Agrawal, A., Batra, D., Parikh, D., & Kembhavi, A. (2018). Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4971–4980).

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., & Zhang, L. (2018). Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, volume 3 (pp.6).

Andreas, J., Rohrbach, M., Darrell, T., & Klein, D. (2016). Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 39–48).

Angwin, J., Larson, J., Mattu, S., & Kirchner, L. (2016). Machine bias. *ProPublica, May*, 23.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., & Parikh, D. (2015). VQA: visual question answering. *CoRR*, abs/1505.00468.

Apter, M. J. (1970). The computer simulation of behaviour.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Barocas, S., Hardt, M., & Narayanan, A. (2018). *Fairness and Machine Learning*. fairmlbook.org. http://www.fairmlbook.org.

Bazrafkan, S., Nedelcu, T., Filipczuk, P., & Corcoran, P. (2017). Deep learning for facial expression recognition: A step closer to a smartphone that knows your moods. In *Consumer Electronics (ICCE), 2017 IEEE International Conference on* (pp. 217–220).: IEEE.

Bechdel, A. (1986). *Dykes to watch out for*. Ithaca, N.Y.: Firebrand Books.

Belinkov, Y., Poliak, A., Shieber, S. M., & Durme, B. V. (2019). Mitigating bias in natural language inference using adversarial learning. Under review.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., & Vaughan, J. W. (2010). A theory of learning from different domains. *Machine learning*, 79(1-2), 151–175.

Ben-David, S., Blitzer, J., Crammer, K., & Pereira, F. (2007). Analysis of representations for domain adaptation. In *Advances in neural information processing systems* (pp. 137–144).

Ben-younes, H., Cadène, R., Cord, M., & Thome, N. (2017). MUTAN: multimodal tucker fusion for visual question answering. *CoRR*, abs/1705.06676.

Benshoff, H. M. (2009). *America on film : representing race, class, gender, and sexuality at the movies*. Chichester, UK ; Malden, MA: Wiley-Blackwell, 2nd ed. edition.

Berk, R., Heidari, H., Jabbari, S., Kearns, M., & Roth, A. (2017). Fairness in criminal justice risk assessments: the state of the art. *arXiv preprint arXiv:1703.09207*.

Bigham, J. P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R. C., Miller, R., Tatarowicz, A., White, B., White, S., et al. (2010). Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology* (pp. 333–342).: ACM.

Block, N. (1981). Psychologism and behaviorism. *The Philosophical Review*, 90(1), 5–43.

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., & Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems* (pp. 4349–4357).

Bowman, S. R., Angeli, G., Potts, C., & Manning, C. D. (2015). A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Buolamwini, J. A. (2017). *Gender shades: intersectional phenotypic and demographic evaluation of face datasets and gender classifiers*. PhD thesis, Massachusetts Institute of Technology.

Buschman, T. J. & Miller, E. K. (2007). Top-down versus bottom-up control of attention in the prefrontal and posterior parietal cortices. *science*, 315(5820), 1860–1862.

Caesar, H., Uijlings, J., & Ferrari, V. (2016). Coco-stuff: Thing and stuff classes in context. *CoRR, abs/1612.03716*, 5, 8.

Caliskan, A., Bryson, J. J., & Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334), 183–186.

Chakraborty, S., Tomsett, R., Raghavendra, R., Harborne, D., Alzantot, M., Cerutti, F., Srivastava, M., Preece, A., Julier, S., Rao, R. M., et al. (2017). Interpretability of deep learning models: a survey of results. In *IEEE Smart World Congress 2017 Workshop: DAIS*.

Chandrasekhar, V., Lin, J., Liao, Q., Morere, O., Veillard, A., Duan, L., & Poggio, T. (2017). Compression of deep neural networks for image instance retrieval. In *Data Compression Conference (DCC), 2017* (pp. 300–309).: IEEE.

Chandu, K. R., Pyreddy, M. A., Felix, M., & Joshi, N. N. (2018). Textually enriched neural module networks for visual question answering. *arXiv preprint arXiv:1809.08697*.

Chao, W.-L., Hu, H., & Sha, F. (2017). Being negative but constructively: Lessons learnt from creating better visual question answering datasets. *arXiv preprint arXiv:1704.07121*.

Chopra, S., Hadsell, R., & LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1 (pp. 539–546).: IEEE.

Chouldechova, A. (2017). Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2), 153–163.

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

Corbetta, M. & Shulman, G. L. (2002). Control of goal-directed and stimulus-driven attention in the brain. *Nature reviews neuroscience*, 3(3), 201.

Das, A., Datta, S., Gkioxari, G., Lee, S., Parikh, D., & Batra, D. (2018). Embodied question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 5 (pp.6).

Doshi-Velez, F. & Kim, B. (2017). Towards a rigorous science of interpretable machine learning.

Duke, B. & Taylor, G. W. (2018). Generalized hadamard-product fusion operators for visual question answering. *arXiv preprint arXiv:1803.09374*.

Elliott, D., Frank, S., Barrault, L., Bougares, F., & Specia, L. (2017). Findings of the second shared task on multimodal machine translation and multilingual image description. *arXiv preprint arXiv:1710.07177*.

Fathallah, A., Abdi, L., & Douik, A. (2017). Facial expression recognition via deep learning. In *Computer Systems and Applications (AICCSA), 2017 IEEE/ACS 14th International Conference on* (pp. 745–750).: IEEE.

Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., Lally, A., Murdock, J. W., Nyberg, E., Prager, J., et al. (2010). Building watson: An overview of the deepqa project. *AI magazine*, 31(3), 59–79.

Fukui, A., Park, D. H., Yang, D., Rohrbach, A., Darrell, T., & Rohrbach, M. (2016). Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., & Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1), 2096–2030.

Gao, H., Mao, J., Zhou, J., Huang, Z., Wang, L., & Xu, W. (2015). Are you talking to a machine? dataset and methods for multilingual image question. In *Advances in neural information processing systems* (pp. 2296–2304).

Gautam, G. & Yadav, D. (2014). Sentiment analysis of twitter data using machine learning approaches and semantic analysis. In *2014 Seventh International Conference on Contemporary Computing (IC3)* (pp. 437–442).

Gelman, S. A. & Markman, E. M. (1987). Young children's inductions from natural kinds: The role of categories and appearances. *Child development*, (pp. 1532–1541).

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672–2680).

Gordon, D., Kembhavi, A., Rastegari, M., Redmon, J., Fox, D., & Farhadi, A. (2017). Iqa: Visual question answering in interactive environments. *arXiv preprint arXiv:1712.03316*, 1.

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., & Parikh, D. (2016). Making the V in VQA matter: Elevating the role of image understanding in visual question answering. *CoRR*, abs/1612.00837.

Grand, G., Blank, I. A., Pereira, F., & Fedorenko, E. (2018a). Semantic projection: recovering human knowledge of multiple, distinct object features from word embeddings. *arXiv preprint arXiv:1802.01241*.

Grand, G., Szanto, A., Kim, Y., & Rush, A. (2018b). On the flip side: Identifying counterexamples in visual question answering.

Gunderson, K. (1964). The imitation game. *Mind*, 73(290), 234–245.

Gunning, D. (2017). Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA), nd Web*.

Gupta, A. K. (2017). Survey of visual question answering: Datasets and techniques. *CoRR*, abs/1705.03865.

Gurari, D., Li, Q., Stangl, A. J., Guo, A., Lin, C., Grauman, K., Luo, J., & Bigham, J. P. (2018). Vizwiz grand challenge: Answering visual questions from blind people. *arXiv preprint arXiv:1802.08218*.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S. R., & Smith, N. A. (2018). Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*.

Hamamoto, D. Y. (1994). *Monitored peril: Asian Americans and the politics of TV representation*. University of Minnesota Press.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. arxiv preprint. arxiv preprint. *arXiv preprint arXiv:1703.06870*.

He, K., Zhang, X., Ren, S., & Sun, J. (2015a). Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

He, K., Zhang, X., Ren, S., & Sun, J. (2015b). Deep residual learning for image recognition. corr, vol. abs/1512.03385.

Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735–1780.

Holtzman, L. & Sharpe, L. (2014). *Media messages: What film, television, and popular music teach us about race, class, gender, and sexual orientation*. Routledge.

Hoy, M. B. (2018). Alexa, siri, cortana, and more: An introduction to voice assistants. *Medical reference services quarterly*, 37(1), 81–88.

Hu, R., Andreas, J., Rohrbach, M., Darrell, T., & Saenko, K. (2017). Learning to reason: End-to-end module networks for visual question answering. *CoRR, abs/1704.05526*, 3.

Hutchinson, M. L., Antono, E., Gibbons, B. M., Paradiso, S., Ling, J., & Meredig, B. (2017). Overcoming data scarcity with transfer learning. *arXiv preprint arXiv:1711.05099*.

Jabri, A., Joulin, A., & van der Maaten, L. (2016). Revisiting visual question answering baselines. In *European conference on computer vision* (pp. 727–739).: Springer.

Jiang, Y., Natarajan, V., Chen, X., Rohrbach, M., Batra, D., & Parikh, D. (2018a). Pythia. https://github.com/facebookresearch/pythia.

Jiang, Y., Natarajan, V., Chen, X., Rohrbach, M., Batra, D., & Parikh, D. (2018b). Pythia v0.1: the winning entry to the vqa challenge 2018. *arXiv preprint arXiv:1807.09956*.

Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C. L., & Girshick, R. (2017). Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on* (pp. 1988–1997).: IEEE.

Kafle, K. & Kanan, C. (2017). Visual question answering: Datasets, algorithms, and future challenges. *Computer Vision and Image Understanding*, 163, 3–20.

Karpathy, A. (2016). Quora response: In your opinion, what are the most interesting topics to research in computer vision? https://www.quora.com/In-your-opinion-what-are-the-most-interesting-topics-to-research-in-computer-vision/answer/Andrej-Karpathy.

Kelly III, J. E. & Hamm, S. (2013). *Smart machines: IBM's Watson and the era of cognitive computing*. Columbia University Press.

Kim, J.-H., Lee, S.-W., Kwak, D., Heo, M.-O., Kim, J., Ha, J.-W., & Zhang, B.-T. (2016a). Multimodal residual learning for visual qa. In *Advances in Neural Information Processing Systems* (pp. 361–369).

Kim, J.-H., On, K.-W., Lim, W., Kim, J., Ha, J.-W., & Zhang, B.-T. (2016b). Hadamard product for low-rank bilinear pooling. *arXiv preprint arXiv:1610.04325*.

Kingma, D. P. & Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Kipf, T. N. & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kiros, R., Salakhutdinov, R., & Zemel, R. (2014). Multimodal neural language models. In *International Conference on Machine Learning* (pp. 595–603).

Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R. S., Torralba, A., Urtasun, R., & Fidler, S. (2015). Skip-thought vectors. *CoRR*, abs/1506.06726.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., et al. (2017). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1), 32–73.

Kryściński, W., Paulus, R., Xiong, C., & Socher, R. (2018). Improving abstraction in text summarization. *arXiv preprint arXiv:1808.07913*.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, e253.

Larson, H. J. & Larson, H. J. (1969). *Introduction to probability theory and statistical inference*, volume 12. Wiley New York.

Lasecki, W. S., Zhong, Y., & Bigham, J. P. (2014). Increasing the bandwidth of crowdsourced visual question answering to better support blind users. In *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility* (pp. 263–264).: ACM.

Lei, J., Yu, L., Bansal, M., & Berg, T. L. (2018). Tvqa: Localized, compositional video question answering. *arXiv preprint arXiv:1809.01696*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740–755).: Springer.

Lloyd, G. A. & Sasson, S. J. (1978). Electronic still camera. US Patent 4,131,919.

López, G., Quesada, L., & Guerrero, L. A. (2017). Alexa vs. siri vs. cortana vs. google assistant: a comparison of speech-based natural user interfaces. In *International Conference on Applied Human Factors and Ergonomics* (pp. 241–250).: Springer.

Lu, J., Lin, X., Batra, D., & Parikh, D. (2015). Deeper lstm and normalized cnn visual question answering model. https://github.com/VT-vision-lab/VQA_LSTM_CNN.

Lu, J., Xiong, C., Parikh, D., & Socher, R. (2017). Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 6 (pp.2).

Lu, J., Yang, J., Batra, D., & Parikh, D. (2016). Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems* (pp. 289–297).

Lu, P., Ji, L., Zhang, W., Duan, N., Zhou, M., & Wang, J. (2018). R-vqa: Learning visual relation facts with semantic attention for visual question answering. *arXiv preprint arXiv:1805.09701*.

Luong, M.-T. & Manning, C. D. (2016). Achieving open vocabulary neural machine translation with hybrid word-character models. *arXiv preprint arXiv:1604.00788*.

Ma, C., Shen, C., Dick, A., Wu, Q., Wang, P., van den Hengel, A., & Reid, I. (2017). Visual question answering with memory-augmented networks. *arXiv preprint arXiv:1707.04968*.

Machinery, C. (1950). Computing machinery and intelligence-am turing. *Mind*, 59(236), 433.

Malinowski, M. & 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* (pp. 1682–1690).

Mansimov, E., Parisotto, E., Ba, J. L., & Salakhutdinov, R. (2015). Generating images from captions with attention. *arXiv preprint arXiv:1511.02793*.

Markman, E. M. & Hutchinson, J. E. (1984). Children's sensitivity to constraints on word meaning: Taxonomic versus thematic relations. *Cognitive psychology*, 16(1), 1–27.

Markoff, J. (2011). Computer wins on 'jeopardy!': trivial, it's not. *New York Times*, 16.

Mascharka, D., Tran, P., Soklaski, R., & Majumdar, A. (2018). Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4942–4950).

Mastro, D. (2017). Race and ethnicity in u.s. media content and effects.

Moor, J. (1976). An analysis of the turing test. *Philosophical Studies*, 30(4).

Moravec, H. P. (1988). *Mind children : the future of robot and human intelligence*. Cambridge, Mass.: Harvard University Press.

Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B., et al. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Narasimhan, M., Lazebnik, S., & Schwing, A. G. (2018). Out of the box: Reasoning with graph convolution nets for factual visual question answering. *arXiv preprint arXiv:1811.00538*.

Norcliffe-Brown, W., Vafeias, E., & Parisot, S. (2018). Learning conditioned graph structures for interpretable visual question answering. *arXiv preprint arXiv:1806.07243*.

Oshikawa, R., Qian, J., & Wang, W. Y. (2018). A survey on natural language processing for fake news detection. *arXiv preprint arXiv:1811.00770*.

Parliament & of the European Union, C. (2016). Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46. *Official Journal of the European Union (OJ)*, 59(1-88), 294.

Paszke, A., Gross, S., Chintala, S., & Chanan, G. (2017). Pytorch.

Paulus, R., Xiong, C., & Socher, R. (2017). A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proc. of NAACL*.

Pfungst, O. (1911). *Clever Hans:(the horse of Mr. Von Osten.) a contribution to experimental animal and human psychology*. Holt, Rinehart and Winston.

Poliak, A., Naradowsky, J., Haldar, A., Rudinger, R., & Van Durme, B. (2018). Hypothesis only baselines in natural language inference. *arXiv preprint arXiv:1805.01042*.

Ramakrishnan, S., Agrawal, A., & Lee, S. (2018). Overcoming language priors in visual question answering with adversarial regularization. *arXiv preprint arXiv:1810.03649*.

Ranzato, M., Chopra, S., Auli, M., & Zaremba, W. (2015). Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Rawles, B. (1975). The media and its effect on black images.

Raymond, E. (1991). Jargon file version 2.8.1. http://catb.org/esr/jargon/oldversions/jarg282.txt.

Ren, M., Kiros, R., & Zemel, R. (2015a). Image question answering: A visual semantic embedding model and a new dataset. *Proc. Advances in Neural Inf. Process. Syst*, 1(2), 5.

Ren, S., He, K., Girshick, R., & Sun, J. (2015b). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91–99).

Shaikhina, T. & Khovanova, N. A. (2017). Handling limited datasets with neural networks in medical applications: A small-data approach. *Artificial intelligence in medicine*, 75, 51–63.

Shankar, S., Halpern, Y., Breck, E., Atwood, J., Wilson, J., & Sculley, D. (2017). No classification without representation: Assessing geodiversity issues in open data sets for the developing world. *arXiv preprint arXiv:1711.08536*.

Shieber, S. M. (1994). Lessons from a restricted turing test. *Commun. ACM*, 37(6), 70–78.

Shieber, S. M. (2004). *The Turing test: Verbal behavior as the hallmark of intelligence*. MIT Press.

Shieber, S. M. (2006). Does the turing test demonstrate intelligence or not? In *Proceedings of the National Conference on Artificial Intelligence*, volume 21 (pp. 1539).: Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Simonyan, K. & Zisserman, A. (2014a). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Simonyan, K. & Zisserman, A. (2014b). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Smith, L. & Gasser, M. (2005). The development of embodied cognition: Six lessons from babies. *Artificial life*, 11(1-2), 13–29.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929–1958.

Sunstein, C. R. (2014). *Why nudge?: The politics of libertarian paternalism*. Yale University Press.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 27* (pp. 3104–3112). Curran Associates, Inc.

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

Teney, D., Anderson, P., He, X., & van den Hengel, A. (2017a). Tips and tricks for visual question answering: Learnings from the 2017 challenge. *arXiv preprint arXiv:1708.02711*.

Teney, D., Liu, L., & van den Hengel, A. (2017b). Graph-structured representations for visual question answering. *arXiv preprint*.

Thomason, J., Gordan, D., & Bisk, Y. (2018). Shifting the baseline: Single modality performance on visual navigation & qa. *arXiv preprint arXiv:1811.00613*.

Tsuchiya, M. (2018). Performance impact caused by hidden bias of training data for recognizing textual entailment. *arXiv preprint arXiv:1804.08117*.

Tucker, L. R. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3), 279–311.

Wang, P., Wu, Q., Shen, C., Dick, A., & van den Hengel, A. (2017). Fvqa: Fact-based visual question answering. *IEEE transactions on pattern analysis and machine intelligence*.

Wang, P., Wu, Q., Shen, C., Hengel, A. v. d., & Dick, A. (2015). Explicit knowledge-based reasoning for visual question answering. *arXiv preprint arXiv:1511.02570*.

Weber, B. (1997). Swift and slashing, computer topples kasparov. *New York Times*, 12.

Windsor, H. H. (1944). Robot works problems never before solved. *Popular Mechanics*, 82(4), 13.

Wu, Q., Teney, D., Wang, P., Shen, C., Dick, A., & van den Hengel, A. (2017). Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding*, 163, 21–40.

Wu, Q., Wang, P., Shen, C., Dick, A., & 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* (pp. 4622–4630).

Xu, H. & Saenko, K. (2016). Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *European Conference on Computer Vision* (pp. 451–466).: Springer.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., & Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning* (pp. 2048–2057).

Yang, Z., He, X., Gao, J., Deng, L., & Smola, A. (2016). Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 21–29).

Ye, Y., Zhao, Z., Li, Y., Chen, L., Xiao, J., & Zhuang, Y. (2017). Video question answering via attribute-augmented attention network learning. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 829–832).: ACM.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? In *Advances in neural information processing systems* (pp. 3320–3328).

You, Q., Jin, H., Wang, Z., Fang, C., & Luo, J. (2016). Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4651–4659).

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zeng, K.-H., Chen, T.-H., Chuang, C.-Y., Liao, Y.-H., Niebles, J. C., & Sun, M. (2017). Leveraging video descriptions to learn video question answering. In *AAAI*.

Zhang, P., Goyal, Y., Summers-Stay, D., Batra, D., & Parikh, D. (2016). Yin and yang: Balancing and answering binary visual questions. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on* (pp. 5014–5022).: IEEE.

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., & Chang, K.-W. (2017). Men also like shopping: Reducing gender bias amplification using corpus-level constraints. *arXiv preprint arXiv:1707.09457*.

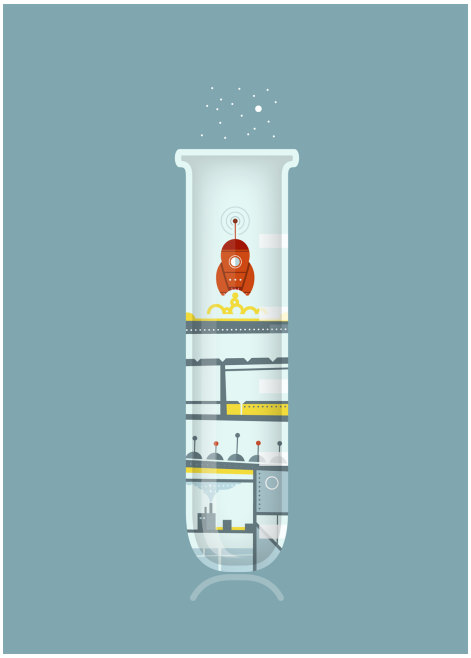
Zhao, Y., Zhang, J., He, Z., Zong, C., & Wu, H. (2018a). Addressing troublesome words in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 391–400).

Zhao, Z., Zhang, Z., Xiao, S., Yu, Z., Yu, J., Cai, D., Wu, F., & Zhuang, Y. (2018b). Open-ended long-form video question answering via adaptive hierarchical reinforced networks. In *IJCAI* (pp. 3683–3689).

Zhou, B., Tian, Y., Sukhbaatar, S., Szlam, A., & Fergus, R. (2015). Simple baseline for visual question answering. *arXiv preprint arXiv:1512.02167*.

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

Zhu, Y., Zhang, C., Ré, C., & Fei-Fei, L. (2015). Building a large-scale multimodal knowledge base system for answering visual queries. *arXiv preprint arXiv:1507.05670*.



This thesis was typeset using LaTeX, originally developed by Leslie Lamport and based on Donald Knuth's TeX. The body text is set in 11 point Egenolff-Berner Garamond, a revival of Claude Garamont's humanist typeface. The above illustration, *Science Experiment 02*, was created by Ben Schlitter and released under CC BY-NC-ND 3.0. A template that can be used to format a PhD dissertation with this look & feel has been released under the permissive AGPL license, and can be found online at github.com/suchow/Dissertate or from its lead author, Jordan Suchow, at suchow@post.harvard.edu.