Shortcut Learning in Visual Question Answering

Corentin Dancette

SCIENCES SORBONNE UNIVERSITÉ

THÈSE DE DOCTORAT DE SORBONNE UNIVERSITÉ
Spécialité **Informatique**
École Doctorale Informatique, Télécommunications et Électronique (Paris)

Shortcut Learning in Visual Question Answering

Étude des biais dans les systèmes de questions-réponses visuelles

Présentée par
Corentin Dancette

Dirigée par
Matthieu Cord

Pour obtenir le grade de
DOCTEUR de SORBONNE UNIVERSITÉ

Présentée et soutenue publiquement le 31 mars 2023

Devant le jury composé de :

Élisa Fromont *Professeure, Université de Rennes 1 – INRIA*	Rapportrice
Christian Wolf *Principal Scientist, Naver Labs*	Rapporteur
Damien Teney *Research Scientist, Idiap Research Institute*	Examinateur
Marcus Rohrbach *Professeur, Technischen Universität Darmstadt*	Examinateur
Nicolas Thome *Professeur, Sorbonne Université*	Examinateur
Matthieu Cord *Professeur, Sorbonne Université*	Directeur de thèse

ABSTRACT

This thesis is focused on the task of Visual Question Answering (VQA): it consists in answering textual questions about images. We investigate Shortcut Learning in this task: the literature reports the tendency of models to learn superficial correlations leading them to correct answers in most cases, but which can fail when encountering unusual input data.

We first propose two methods to reduce shortcut learning on VQA. The first, which we call Reducing Unimodal Biases (RUBi), consists of an additional loss to encourage the model to learn from the most difficult and less biased examples – those which cannot be answered solely from the question. We show that our method can reduce question-based shortcuts in existing VQA models, especially when tested on data with a distribution shift. We then propose a model for the more specific task of visual counting – a subset of VQA consisting only of counting questions. We design Spatial Counting Network (SCN), a model which incorporates architectural priors designed to make it more robust to distribution shifts. We show that SCN has superior performances on out-of-distribution benchmarks compared to existing models.

We then study the existence of multimodal shortcuts in the VQA dataset. We show that shortcuts are not only based on correlations between the question and the answer but can also involve image information. We design an evaluation benchmark to measure the robustness of models to multimodal shortcuts. We show that existing models are vulnerable to multimodal shortcut learning.

The learning of those shortcuts is particularly harmful when models are evaluated in an out-of-distribution context. Therefore, it is important to evaluate the reliability of VQA models, i.e. the ability to assess their confidence in the given answer. We propose a method to improve the reliability of VQA models, i.e. their ability to abstain from answering when their confidence is too low. It consists of training an external "selector" model to predict the confidence of the VQA model. This selector is trained using a cross-validation-like scheme in order to avoid overfitting on the training set but still using all the available data. We show that our method can improve the reliability of existing VQA models, in both in-distribution and out-of-distribution settings.

RÉSUMÉ

Cette thèse se concentre sur la tâche de Visual Question Answering (VQA), c'est à dire les systèmes questions-réponses visuelles. Elle consiste à répondre à des questions à propos de photographies. Nous étudions l'apprentissage des biais dans cette tâche. La littérature montre que les modèles ont tendance à apprendre des corrélations superficielles les conduisant à des réponses correctes dans la plupart des cas, mais qui peuvent échouer lorsqu'ils rencontrent des données d'entrée inhabituelles.

Nous proposons deux méthodes pour réduire l'apprentissage par raccourci sur le VQA. La première, RUBi, consiste à encourager le modèle à apprendre à partir des exemples les plus difficiles et les moins biaisés grâce à une *loss* spécifique. Nous montrons que notre méthode peut réduire les biais basés sur les questions dans les modèles VQA, en particulier lorsqu'ils sont testés sur des données avec un changement de distribution. Nous proposons ensuite un modèle pour la tâche de comptage visuel – un sous-ensemble de VQA composé uniquement de questions de comptage. Nous proposons SCN, un modèle doté d'une architecture conçue pour être robuste aux changements de distribution. Nous montrons que SCN a des performances supérieures à celles des modèles existants sur les benchmarks *out-of-distribution*.

Nous étudions ensuite les raccourcis multimodaux dans le VQA. Nous montrons qu'ils ne sont pas seulement basés sur des corrélations entre la question et la réponse, mais qu'ils peuvent aussi impliquer des informations sur l'image. Nous concevons un benchmark d'évaluation pour mesurer la robustesse des modèles aux raccourcis multimodaux. Nous montrons que les modèles existants y sont particulièrement vulnérables.

L'apprentissage de ces raccourcis est particulièrement problématique lorsque les modèles sont testés dans un contexte de changement de distribution. C'est pourquoi il est important de pouvoir évaluer la fiabilité des modèles VQA, c'est-à-dire notre capacité à évaluer leur confiance dans la réponse donnée. Nous proposons une méthode pour améliorer cette fiabilité, afin de leur permettre de s'abstenir de répondre lorsque leur confiance est trop faible. Cette méthode consiste à entraîner un modèle externe, dit "sélecteur", pour prédire la confiance du modèle VQA, à l'aide d'un système similaire à la validation croisée afin d'éviter un surajustement du modèle tout en utilisant toutes les données disponibles. Nous montrons que notre méthode peut améliorer la fiabilité des modèles VQA existants, à la fois dans le cadre de la distribution et hors de la distribution.

REMERCIEMENTS

Je souhaite remercier toutes les personnes qui m'ont accompagné pendant cette thèse. Tout d'abord, mon directeur de thèse, Matthieu Cord. Il a su me guider pendant ces trois années de thèse, apportant toujours de bons conseils, à la fois pendant le travail préliminaire de recherche, et lors de la rédaction et la soumission des articles.

Je remercie les rapporteurs de ma thèse, Elisa Fromont et Christian Wolf, pour leur relecture et leurs commentairs constructifs, ainsi qu'aux examintateurs de mon jury, Damien Teney, Marcus Rorhbach, et Nicolas Thome. Un grand merci à Marcus Rorhbach, pour m'avoir accueilli pendant quatre mois dans son équipe à Meta AI pour une collaboration très enrichissante et fructueuse, ainsi qu'à Damien Teney pour sa collaboration sur un de mes papiers. Merci aux collègues et amis que j'ai pu rencontrer pendant mon stage en Californie et qui ont rendu cette expérience enrichissante : Alexis, pour ta visite et m'avoir initié au surf ; Medhini, Pierre, Spencer, Casey, Polina, et tous les autres.

Merci à toute l'équipe du MLIA pour les bons moments passés ensemble, les discussions scientifiques et les parties de babyfoot. En particulier, aux *Chordettes* : Rémi Cadène, avec qui j'ai pu collaborer pendant le début de ma thèse et de qui j'ai beaucoup appris ; Arthur Douillard, Alexandre Ramé, Antoine Saporta, Guillaume Couairon, Yifu Chen, Rémy Sun, Mustafa Shukor, et à tous les autres doctorants de l'équipe. Un grand merci également a tous mes amis, qui m'ont soutenu pendant ces trois années, et qui ont rendu ces années si agréables. Louis, Hugo, Luca, Rachid, qui ont eux aussi suivi cette aventure qu'est la thèse et avec qui j'ai beaucoup échangé, mais aussi à Adrien, Pierre, Ayaz, David, Sarah, Thomas, Arnault, Wenceslas, Jonathan, Alexandre, Ghislain, Flavie, et à tous les autres. Enfin, merci à ma famille : à ma soeur, ma mère et mon père, pour leur soutien pendant ce travail, pour leur intéret pour la science qu'il m'ont transmis, et sans qui je n'aurais pas pu mener à bien cette thèse.

CONTENTS

ACRONYMS

AP	Average Precision
CNN	Convolutional Neural Network
CV	Computer Vision
DNN	Deep Neural Network
LYP	Learning from Your Peers
MLP	Multi-Layer Perceptron
NLP	Natural Language Processing
OOD	out-of-distribution
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
RUBi	Reducing Unimodal Biases
VQA	Visual Question Answering
VQA-CE	VQA-CounterExamples
LSTM	Long Short-Term Memory network
SAN	Stacked Attention Network
SVM	Support Vector Machine
SCN	Spatial Counting Network
UpDn	Bottom-Up and Top-Down Attention

CHAPTER 1

INTRODUCTION

> A computer would deserve to be called intelligent if it could deceive a human into believing that it was human. — *Alan Turing*

Deep Learning — the use of Deep Neural Networks trained on large amounts of data — has led to major breakthroughs in many fields, such as computer vision (Krizhevsky et al. 2012a) and natural language processing (Mikolov et al. 2013). This pushed researchers to explore multimodal tasks that combine both visual and textual modalities (Kiros et al. 2015b; Karpathy and Fei-Fei 2015; C. Lu et al. 2016; Das et al. 2017b; Vries et al. 2017). Among these tasks, VQA has attracted increasing attention. The goal of the VQA task is to answer a question about an image. It requires a high-level understanding of the visual scene and the question, and also to link the words from the question with the regions in the image and use both modalities adequately. Studying VQA is important for two reasons. First, it is a challenging task that requires complex processing of the scene and the question in order to answer correctly. It has been referred to as a *visual Turing test* (Geman et al. 2015). It is an important benchmark for advances in multimodal understanding and reasoning. Second, solving the VQA task could have direct and tremendous impacts on real-world applications such as aiding visually impaired users in understanding their physical and online surroundings (Gurari et al. 2018), searching through large quantities of visual data via natural language interfaces, or even communicating with robots using more efficient and intuitive interfaces. Multiple datasets have been proposed by the community, ranging from synthetic datasets such as CLEVR (Justin Johnson et al. 2017a), large-scale realistic datasets like VQA v1 and v2 (Antol et al. 2015a; Goyal et al. 2017a), datasets focused on visually impaired users (Gurari et al. 2018), or datasets targeting the medical domain (Abacha et al. 2019).

Visual Question Answering has been tackled using Deep Learning methods: huge statistical models, trained to output the desired answer based on a large number of examples. However, those models are particularly sensitive to what

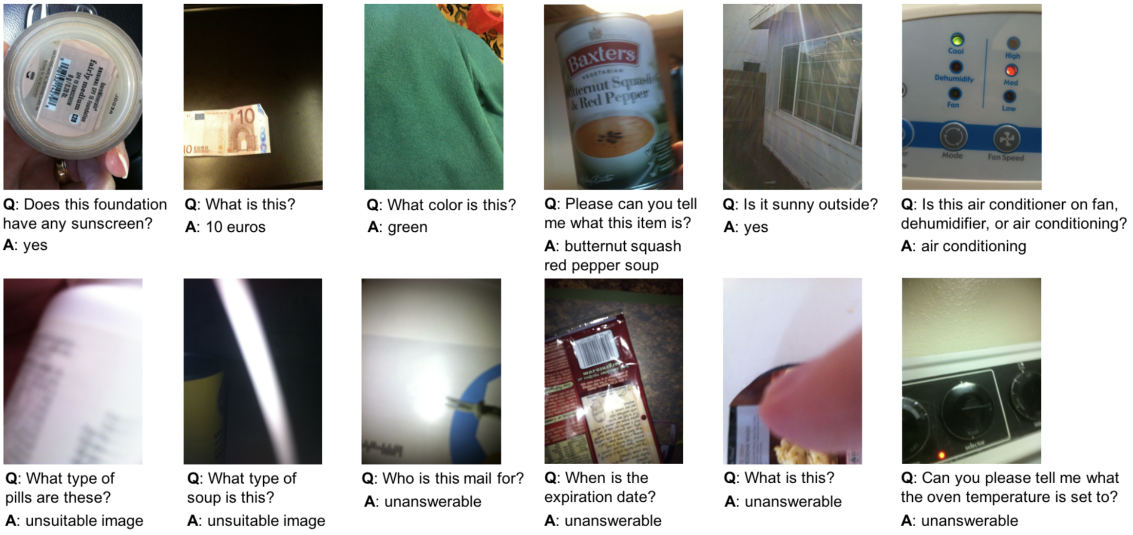


Figure 1.1. – Example of a VQA task from the VizWiz datasets: visually impaired users ask questions about images taken with their smartphone.

has been coined *Shortcut Learning* (Geirhos et al. 2020): they will find the simplest correlations between the input data and the answers. This is often a desired property, but it can also lead to biases in the model when those correlations are not *causal*. For example, if the model is trained on a dataset where all answers for *"What is the color of the sky?"* is *blue*, then the simplest behavior for the model would be to always output *blue* for this question. This is a problem, as it would not be able to answer correctly if the sky is not blue. Additionally, when evaluating the models on testing sets with the same distributions, using those shortcuts will lead to high accuracy. However, when using the model in the real world, it might lead to catastrophic failure. Shortcut learning is a problem that is not specific to VQA, but it is particularly important in this task, as it combines multiple modalities and requires a high-level understanding of the scene and reasoning to answer correctly.

1.1 Contributions

In this Thesis, we tackle the problem of shortcut learning in the context of VQA. We propose to explore various directions related to shortcut learning in VQA: strategies to mitigate shortcut learning, by influencing the model's preferred solutions using inductive biases, methodologies to detect shortcuts in VQA datasets, and models' reliability in out-of-distribution settings. The goal is to better understand the problem of shortcut learning in VQA and to propose solutions to mitigate it, making models closer to real-world usage.

The first axis of our thesis, in Chapters 3 and 4, is centered around reducing shortcut learning in VQA models using various inductive biases. Then, we focus on detecting shortcuts and evaluating models' performance and reliability in out-of-distribution settings, in Chapters 5 and 6. We summarize our contributions in the following list:

- Chapter 3: A LEARNING STRATEGY TO REDUCE UNIMODAL BIASES IN VQA
  First, we explore the reduction of shortcut learning in VQA models. We use the VQA-CP benchmark, which is designed with a distribution shift between the training set and the testing set. It penalizes models that rely on statistical regularities between the question and the answer. This allows us to test the robustness of a model and its learning procedure. In this context, we propose a strategy to reduce shortcut learning in VQA coming from the question modality and encourage the model to use the visual input to answer correctly. Our method reduces the importance of the most biased examples, i.e. examples that can be correctly classified without looking at the image and increases the importance of the most difficult examples, i.e. examples that require the model to use the image to answer correctly. We leverage a question-only model that captures the language biases by identifying when these unwanted regularities are used. It prevents the base VQA model from learning them by influencing its predictions. This leads to dynamically adjusting the loss to compensate for biases.

- Chapter 4: REDUCING SHORTCUT LEARNING WITH ARCHITECTURAL PRIORS FOR VISUAL COUNTING
  We then focus on Visual Counting, a sub-task of Visual Question Answering. It consists in answering counting questions about an image, for example, "How many cats are to the left of the car?", the output being a number. This task is also subject to the same kind of shortcuts as the main VQA task. We use it to explore a second way of reducing shortcut learning with inductive biases: architectural priors. First, we propose a benchmark to evaluate shortcut learning and out-of-distribution generalization, similar to the VQA dataset VQA-CP, but for visual counting. Then, we propose a model which incorporates inductive biases in the deep architecture to guide the model and ground its decision in the image: we structure the model's architecture around the selection of individual objects based on the textual question.

- Chapter 5: DETECTING MULTIMODAL SHORTCUTS FOR VQA
  Most previous work on VQA focuses on the issue of question-based shortcuts: superficial correlations between the question words and the answer. We

investigate the existence of multimodal shortcuts in VQA datasets: simple vision and language patterns that are associated with high certainty with a given answer. We propose a method to find simple patterns in the data: for example, the presence of a racket in the image, with the words "what" and "sports" in the question will most likely lead to the answer "tennis". Those patterns might not hold in all examples, but might be learned by VQA models. We can use examples that contradict those patterns to evaluate the robustness of the model: if it relies on those patterns, it will fail on the examples that contradict them.

- Chapter 6: RELIABILITY FOR VISUAL QUESTION ANSWERING
  Finally, we explore a complementary problem of shortcut learning: the reliability of VQA models. Reliability is the capacity of a model to return a confidence score in addition to an answer. This makes it possible for the model to abstain when the risk of failure is too high. We assess models' reliability under distribution shift, in an out-of-distribution (OOD) setting. The VQA models might be overconfident, especially if they learned simple shortcuts which do not work on the OOD dataset. We evaluate how large pre-trained vision-and-language models perform on this reliability task, and propose a method to improve reliability for VQA.

1.2 Related publications

This thesis is based on the material published in the following papers:

- Remi Cadene, Corentin Dancette, Hedi Ben-Younes, Matthieu Cord, and Devi Parikh (2019c). "RUBi: Reducing Unimodal Biases for Visual Question Answering". In: *Advances in Neural Information Processing Systems (NeurIPS)*

- Corentin Dancette, Remi Cadene, Xinlei Chen, and Matthieu Cord (2021a). "Learning Reasoning Mechanisms for Unbiased Question-based Counting". In: *VQA Workshop, Conference on Computer Vision and Pattern Recognition (CVPR)*

- Corentin Dancette, Remi Cadene, Damien Teney, and Matthieu Cord (2021b). "Beyond Question-Based Biases: Assessing Multimodal Shortcut Learning in Visual Question Answering". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*

- Corentin Dancette, Spencer Whitehead, Rishabh Maheshwary, Ramakrishna Vedantam, Stefan Scherer, Xinlei Chen, Matthieu Cord, and Marcus Rohrbach

(2023). "Improving Selective VQA by learning from your peers". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*

Additionally, we worked on other projects, that are not included in this thesis, as they are not directly focused on the main topic of this thesis, but tackle related problems, such as domain generalization, or compute efficiency of the transformer's architecture.

- **Fishr: Invariant gradient variances for out-of-distribution generalization**. We explore the task of domain generalization: given multiple training domains with different distributions, we want to learn a model that performs well on another unseen test distribution. For this, models must learn to use *invariant* features, that are not specific to a given domain, but equally predictive in all domains. For instance, the Colored-MNIST dataset proposes splits where the color is predictive of the number, but with different colors for each split. We propose a method that constrains the gradient variances of the model across environments to be similar. This work led to the publication of a conference article:
  Alexandre Rame, Corentin Dancette, and Matthieu Cord (2022). "Fishr: Invariant gradient variances for out-of-distribution generalization". In: *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 18347–18377

- **Dynamic Query Selection for Fast Visual Perceive**. The Transformer is an effective architecture for deep learning but has a quadratic complexity in the number of tokens, which is problematic for large inputs. The perceiver is a transformer-based model designed to reduce the number of input tokens, by having a smaller and fixed number of "queries". In this work, we analyze the perceiver architecture for computer vision and show how to make it more efficient by selecting dynamically only the most important queries. This article led to the publication of a workshop article:
  Corentin Dancette and Matthieu Cord (2022). "Dynamic Query Selection for Fast Visual Perceiver". In: *CVPR Workshop, Transformers for Vision*

We open-sourced the code concerning the following chapters:

- Chapter 3: `https://github.com/cdancette/rubi.bootstrap.pytorch`
- Chapter 4: `https://github.com/cdancette/spatial-counting-network`
- Chapter 5: `https://github.com/cdancette/detect-shortcuts`

CHAPTER 2

BACKGROUND AND CONTEXT

In this chapter, we present a literature review of the works related to the thesis. First, we introduce the topic of Deep Learning for Computer Vision and Natural Language Processing. We then discuss works related to Visual Question Answering (VQA). Finally, we discuss the literature on biases and shortcut learning in deep neural networks, especially in the VQA task.

2.1 Deep Learning for Text and Image

Deep Learning (LeCun et al. 2015) is a subfield of machine learning that focuses on learning representations from data using deep neural networks. Neural networks are statistical models (Vapnik 1999), loosely inspired by the human brain. They are high-dimensional functions composed of multiple layers of linear transformations called *neurons* and non-linear *activation functions*. This basic building block of deep neural networks is called the Multi-Layer Perceptron (MLP).

In this thesis we mainly use the *Supervised Learning* setting: we have a training dataset $\mathcal{D}$ containing samples. Each sample contains an input x and with a ground-truth label y that the model is trained to predict. For example, for image classification, each image is associated with a class describing its content. Other settings include unsupervised or self-supervised learning, when no labeled data is available (Hastie et al. 2009), and semi-supervised learning, the middle ground between the two previous settings where the data is partially annotated (Chapelle et al. 2009). In supervised learning, a model f with parameters θ is trained using a loss function $\mathcal{L}$ that measures the distance between the predictions of the model and the target, using a variant of stochastic gradient descent (Bottou et al. 1998). The gradient is usually computed using the backpropagation algorithm (Rumelhart et al. 1986). Computing the output of the neural network f for a given input x is called the *forward pass*, and computing $\nabla_\theta \mathcal{L}$, the gradients of the loss with respect to its weights, is called the *backward* pass. The weights are updated in the opposite direction of the gradient to minimize the loss $\mathcal{L}$.

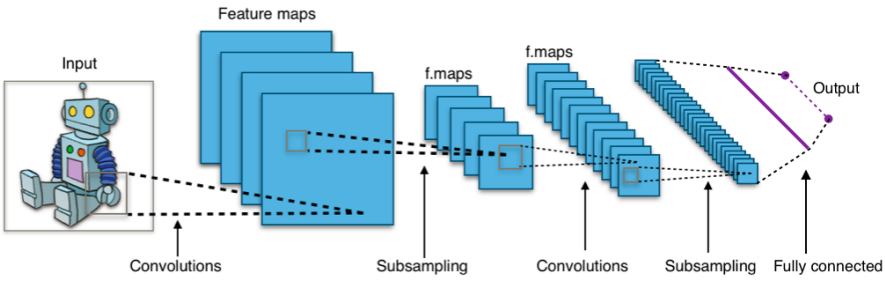


Figure 2.1. – A typical convolutional neural network architecture. It is composed of multiple convolution layers, that match local patterns, pooling layers that reduce the spatial dimension of the feature maps, and fully connected layers at the end, to return a classification output. Image from https://en.wikipedia.org/wiki/Convolutional_neural_network

In the following sections, we introduce the domains of Computer Vision and Natural Language Processing and explain how Deep Learning is used in those domains.

2.1.1 Computer Vision

Computer Vision (CV), the study of image processing and understanding, has been studied for multiple decades. Until 2012, the state-of-the-art algorithms for image classification were based on hand-crafted features, such as Histogram of Oriented Gradients (HOG) (Dalal and Triggs 2005) or Bag-of-Words-based strategies using local features like SIFT (Lowe 1999), projected on a Visual Dictionary (Fournier et al. 2001; Sivic and Zisserman 2003) with linear classifiers learned on top, like Support Vector Machine (SVM) (Boser et al. 1992). At the ILSVRC 2012 challenge (Russakovsky et al. 2015a), a large-scale image classification benchmark, (Krizhevsky et al. 2012b) won the competition with AlexNet, a Deep Neural Network. Since then, deep neural networks have been used in many Computer Vision tasks, such as image classification, object detection, image segmentation, and many others, surpassing hand-crafted feature representations. Most architectures used in Computer Visions are based on the Convolutional Neural Network (CNN) architecture: a neural network composed of learned convolution layers and non-linear activation functions such as the Rectified Linear Unit (ReLU). We display a typical CNN architecture in Figure 2.1.

2.1.2 Natural Language Processing

Natural Language Processing (NLP) is the study of text understanding. It ranges from tasks like summarization, language modeling or text generation, translation, text classification and many others. The first NLP systems used fixed and hand-crafted rule-based systems (Weizenbaum 1966). Early probabilistic systems used in NLP were based on n-grams or hidden Markov models (Cavnar, Trenkle, et al. 1994; Robertson and Willett 1998; Witten et al. 1999). The first popular *Deep Learning* models for NLP were Recurrent Neural Network (RNN), like the Long Short-Term Memory network (LSTM) (Gers and Schmidhuber 2001). These architectures process each word or token sequentially, one by one, using the same model with a memory saved between each forward pass. It is trained with a variant of backpropagation called backpropagation through time, as the network is called multiple times with the same weights. Other variants of RNN include the GRU architecture (Cho et al. 2014). These models can take as input token ids, or word embeddings: vectors that represent the semantic meaning of a word (Turian et al. 2010). The word2vec approach (Mikolov et al. 2013) is a popular technique that uses neural networks to learn word embeddings based on word co-occurrences.

More recently was proposed the Transformer architecture (Vaswani et al. 2017): it is composed of multiple attention layers, and all tokens are processed in parallel instead of sequentially like in a RNN. At each layer of the model, a token receives updates from its own representation using an MLP, and from the other tokens of the sequence using a Multi-Head Self-Attention mechanism. We display the full architecture in Figure 2.2. The attention is computed as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{2.1}$$

where Q, K, and V are respectively the queries, keys and values, and are obtained by linear transformations from input tokens.

This results in more efficient forward and backward passes, which makes it possible to train them on larger datasets. Those models are usually pre-trained using self-supervised learning (they learn representations from a stream of text without any labels) using one of the following strategies. (a) Masked Language Modeling: a word is masked in the input text, and the whole input is used to predict the word. (b) Next Token Prediction or Autoregressive generation: the model can only use the previous words to predict the next word. This is often used for language generation. This approach has been shown to scale fairly well with very large models and datasets, (Devlin et al. 2019; Radford et al. 2018; Radford et al. 2019; Brown et al. 2020), reaching a hundred billion parameters.

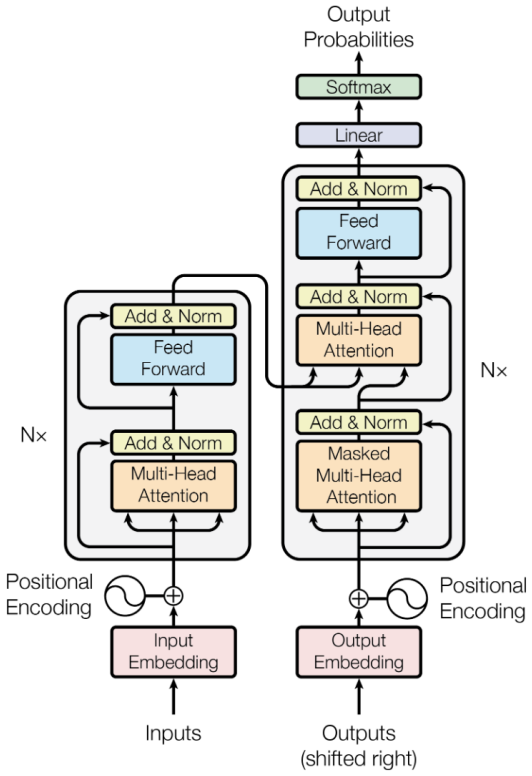


Figure 2.2. – The Transformer architecture from Vaswani et al. 2017. It is composed of multiple Attention and Feed Forward (MLP) layers.

For instance, GPT-3 (Brown et al. 2020) contains 175 billion parameters, Palm (Chowdhery et al. 2022) contains 540 billion, and Switch Transformer reaches 1.5 trillion parameters with a sparse model (Fedus et al. 2021). Those models, pre-trained on large-scale unlabeled data, and fine-tuned for many downstream tasks, are sometimes referred to as *foundation models*. Emerging properties have been shown to appear at those scales, like few-shot or zero-shot evaluation with prompt conditioning (Brown et al. 2020).

The Transformer architecture is also very general and can be adapted to many other modalities, as it just requires partitioning the input as a set of tokens. It is now used for computer vision tasks, using image patches. Vision Transformer architectures now reach similar accuracies to the best CNN models (Dosovitskiy et al. 2021; Touvron et al. 2021). It has also been used for vision-and-language learning, like CLIP (Radford et al. 2021). We will detail this further in the next section.

2.2 Visual Question Answering

While the two fields of Computer Vision and Natural Language Processing have been studied for a long time, the task of bridging the gap between the two has been studied more recently, with tasks like Image-text retrieval, Image Captioning, Text to image generation, and others.

In this thesis, we focus on the task of Visual Question Answering (VQA). It consists in answering a natural language question about an image, for example, asking *How many slices of pizza are there?*, like in the example from Figure 2.3. In order to answer this complex question, the network must be able to process the image, understand the text, and also model the interactions between the two modalities. This task, referred to as a *Visual Turing Test* by (Geman et al. 2015), requires high-level text and image understanding. This task is interesting for multiple reasons. First, it is a benchmark to test jointly the textual, visual and reasoning abilities of artificial models. Then, it has many direct applications, such as assistance for visually impaired users (Gurari et al. 2018), communicating with robots, searching the web more efficiently, and many others.

The very first works to study the Visual Question Answering task had a restricted focus: Malinowski and Fritz 2014a introduced the DAQUAR dataset, and Geman et al. 2015 introduced their *Visual Turing Test*. Both of those datasets are built with questions or answer that come from a small-sized fixed vocabulary. Also, their size is limited to a few thousand images. This scale is not sufficient to train a deep neural network that will capture the complexity of the task. Then, the following works proposed larger datasets that enabled the successful development of Deep Learning approaches for VQA.

2.2.1 VQA datasets

VQA v1 and v2 The main datasets used by the community in recent years are VQA v1(Antol et al. 2015a) and VQA v2 (Goyal et al. 2017a). They are composed of 123K and 443K images respectively for their training sets, and involve "free-form" and "open-ended questions and answers provided by humans". The images are from the COCO dataset (T.-Y. Lin et al. 2014a) and come from various "domains", such as indoor scenes, outdoor scenes, animals, and people. We display in Figure 2.3 two images with their associated questions in the VQA v1 dataset. Note that in the thesis, we focus on "Open-ended" VQA, which is the most challenging task. Antol et al. 2015a also provides a multiple-choice version of the task, for which each image-question pair is associated with a list of 18 possible answers.

What color are her eyes?
What is the mustache made of?

How many slices of pizza are there?
Is this a vegetarian pizza?

Figure 2.3. – Images and their associated questions from the VQA v1 dataset (Antol et al. 2015a). The questions and images cover a very large variety of objects and topics.

Dataset collection The two datasets were collected in a two-step process: first, the images were shown to people with the following prompt (from (Antol et al. 2015a)).

> "We have built a smart robot. It understands a lot about images. It can recognize and name all the objects, it knows where the objects are, and it can recognize the scene (e.g., kitchen, beach), people's expressions and poses, and properties of objects (e.g., the color of objects, their texture). Your task is to stump this smart robot! Ask a question about this scene that this smart robot probably can not answer, but any human can easily answer while looking at the scene in the image."

Then, for each pair of question and image, were gathered 10 answers from 10 different people, to account for the diversity of possible answers. Answers may be simple words like "yes", "no", or a short phrase, but annotators were asked to avoid complete sentences.

Evaluating a model We have 10 possible answers for each question, therefore the evaluation of a model's predictions must be done in a way that takes into account the number of ground truth answers that match the model's predictions. The VQA accuracy is computed using the following formula:

$$\text{accuracy} = \min\left(\frac{\#\text{ humans that provided that answer}}{3}, 1\right)$$

Therefore, for a model to get a perfect score, it must predict the answer that was given by at least 3 people. To give an idea of the best score a model could get,

the average human performance is around 83.30 percent overall, (95.77 for yes-no questions, 83.39 for number questions and 72.67 for other questions).

We explain in Section 2.4.1 the main differences between VQA v1 and VQA v2.

CLEVR (Justin Johnson et al. 2017b) is a synthetic dataset where both images and questions are generated. The images are composed of a background, a set of simple objects (cubes, cylinders and spheres) with a few attributes (color, texture, position). The questions refer to those attributes and objects and can be compositional. We show an example of an image and question from CLEVR in Figure 2.4. It is a good benchmark to study compositional reasoning.

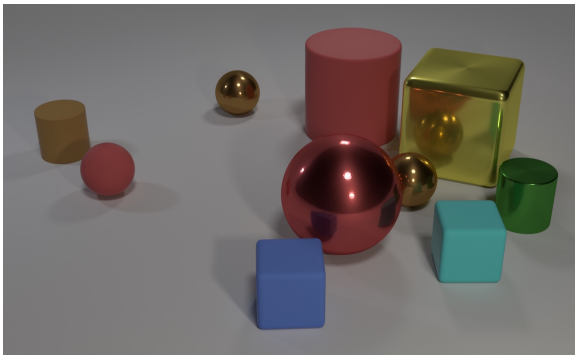


Figure 2.4. – An image from the CLEVR dataset. One of the associated questions is "Q: What size is the cylinder that is left of the brown metal thing that is left of the big sphere?". Image and question from Justin Johnson et al. 2017b.

GQA (Drew A Hudson and Manning 2019) is a VQA dataset that is built from real images from the Visual Genome dataset (Krishna et al. 2017a), but questions are synthetically generated from the scene graph. It has a much more varied number of objects and attributes than CLEVR and is therefore more challenging in that regard while containing more compositional questions than VQA v1 and v2. We show an image and question from GQA in Figure 2.5.

Visual Genome (Krishna et al. 2017a) is a dataset containing 108K images, with multiple kinds of annotations: bounding boxes, object attributes, relationships between objects, etc. It also contains natural language questions and answers on 101K images.

VizWiz (Gurari et al. 2018) is a real-world VQA dataset collected by visually impaired users. They were asked to take pictures with their smartphone and ask questions when they needed information about what was in the picture.

Figure 2.5. – An image from the GQA dataset. One of the associated questions is "Q: Is there any fruit to the left of the tray the cup is on top of? A: yes?". Image and question from Drew A Hudson and Manning 2019.

Answers were then annotated by sighted users. This dataset is very challenging, as questions vary a lot in terms of difficulty, and the quality of images is not always good: some images are blurry, or the objects are not well centered or partially occluded. We show an example from VizWiz in Figure 1.1.

TDIUC (*Task Driven Image Understanding*) (Kafle and Kanan 2017) propose a VQA dataset with questions from 12 distinct tasks, like object presence, scene classification, activity recognition, counting...

2.2.2 VQA Architectures

The VQA task can be formalized as a supervised learning problem. The dataset $\mathcal{D}$ is composed of n triplets $(v_i, q_i, a_i)_{i\in[1,n]}$ with $v_i \in \mathcal{V}$ an image, $q_i \in \mathcal{Q}$ a question in natural language and $a_i \in \mathcal{A}$ an answer. One must learn a function $f : \mathcal{V} \times \mathcal{Q} \rightarrow \mathcal{A}$ with parameters θ to produce accurate predictions.

Most of the literature considers the VQA v1 and v2 datasets as single-class or multi-class classification problems: they keep only the K most common answers, K usually being around 3000. We thus have $|\mathcal{A}| = K$. Each question is associated with single or multiple answers, and $\mathbf{a_i} \in \mathbb{R}^{|\mathcal{A}|}$ represents the target probability distribution and $f : \mathcal{V} \times \mathcal{Q} \rightarrow \mathbb{R}^{|\mathcal{A}|}$. The function f is then learned using a Cross-Entropy loss:

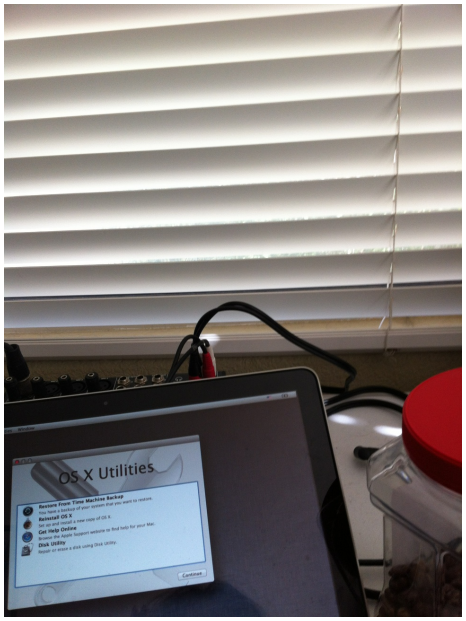


Figure 2.6. – An image from the VizWiz dataset. One of the associated questions is "Q: If there is any text on the screen, what does it say? A: os x utilities". Image and question from Gurari et al. 2018.

$$\mathcal{L}(\theta; \mathcal{D}) = -\frac{1}{n} \sum_{i=1}^{n} \mathbf{a_i} \cdot \log(\text{softmax}(f(v_i, q_i))). \tag{2.2}$$

2.2.2.1 Specialized Fusion-based architectures

Early architectures proposed for the VQA task were specialized architectures designed solely for the task. They use pre-trained unimodal models to embed the image and the text into an embedding space, and then propose a fusion strategy to combine the two modalities and output an answer. (Antol et al. 2015a) in the original VQA paper proposes the following approach: visual features are extracted with VGG (Simonyan and Zisserman 2015), a deep convolutional neural network, pre-trained on ImageNet (Russakovsky et al. 2015b). This gives a 4096-dimensional feature vector. Textual features are extracted with a LSTM (Gers and Schmidhuber 2001), which is initialized randomly and trained for the VQA task. Both features are projected to a 1024-dimensional vector with a linear layer, then pointwise multiplied. A fully-connected linear layer then projects this vector into answer space. The architecture is displayed in Figure 2.7. The model is trained with standard Cross-Entropy on the most common ground truth answer for each input example.

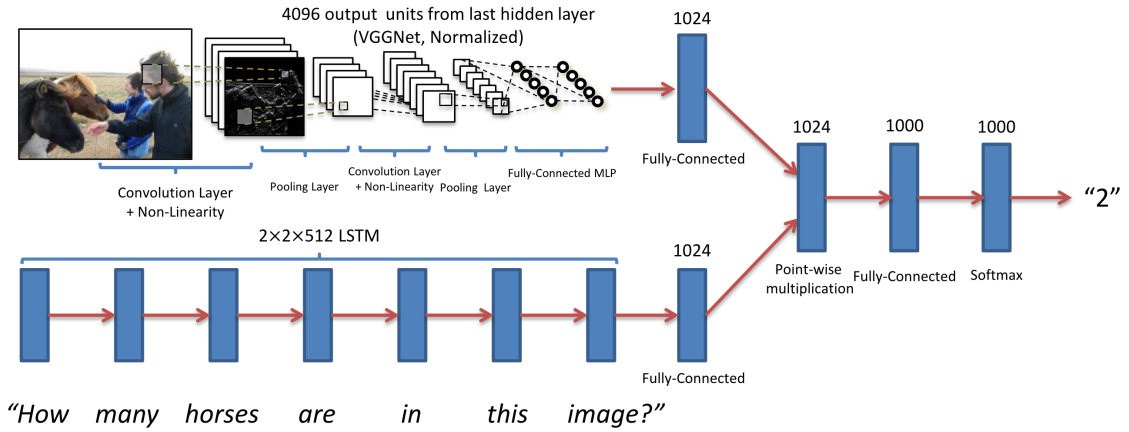


Figure 2.7. – LSTM + CNN: architecture proposed by Antol et al. 2015a to solve the VQA task. It is composed of a pre-trained VGG (Simonyan and Zisserman 2015) CNN to process the image, and an LSTM to process the question. The two resulting embeddings are merged with a point-wise multiplication, then projected to the answer space using a linear layer.

The multimodal fusion strategy here is a simple point-wise multiplication of the two vectors. Further works have proposed more elaborate fusion strategies: Ben-Younes et al. 2017a proposed the MUTAN fusion scheme for VQA. Instead of a point-wise multiplication, The image $\mathbf{v}$ and question $\mathbf{q}$ are merged using a Bilinear model using a tensor operator $\mathcal{T}$. The output of the fusion operation is $\mathbf{y} = (\mathcal{T} \times_1 \mathbf{q}) \times_2 \mathbf{v}$ The final model is displayed in Figure 2.8.

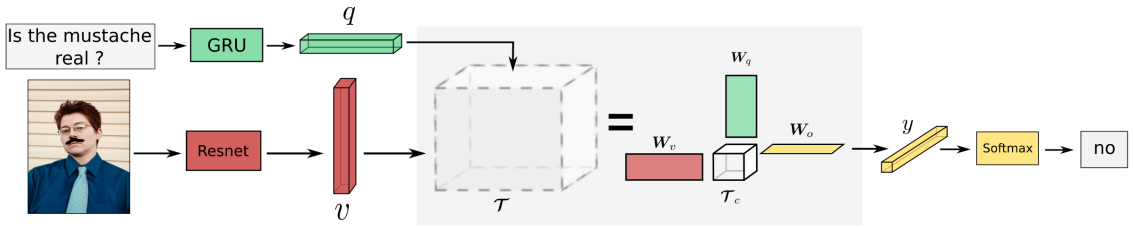


Figure 2.8. – Architecture of MUTAN, by Ben-Younes et al. 2017a. It is similar to the LSTM + CNN architecture, but the fusion operation is a bilinear fusion, with a factorization that allows to drastically reduce the number of parameters, from billions to a few millions.

Other works using bilinear fusion include MCB (Fukui et al. 2016b), MLB (J.-H. Kim et al. 2017), MUREL (Cadene et al. 2019a) and others (Yu et al. 2017; Yu et al. 2018).

2.2.2.2 Iterative reasoning and neural-symbolic approaches

VQA is a reasoning task, that sometimes requires composing multiple operations to answer questions like *What is hanging on the wall above the bed?*. First, it requires locating the bed in the image, then analyzing and identifying what is above it. Some works propose an architecture that takes inspiration from this intuitive idea of iterative reasoning. Yang et al. 2016 proposed an architecture composed of multiple blocks, with different weights, where each block can query different parts of the image. Cadene et al. 2019a proposed MUREL a multi-step recurrent architecture that uses bilinear fusion blocks, applied recursively.

Another class of VQA architecture which is sometimes referred to as "neural-symbolic", consists in having blocks dedicated to specific functions, like object identification, positional reasoning, counting, etc. and then combining those neural network blocks with symbolic reasoning. (Andreas et al. 2016; R. Hu et al. 2017; R. Hu et al. 2018; Jiaxin Shi 2019). The blocks are often applied recursively. By design, those architectures are also more explainable than other approaches: the reasoning process is more explicit, as the block operations can be interpreted. We show in Figure 2.9 the architecture of Neural Module Networks (Andreas et al. 2016).

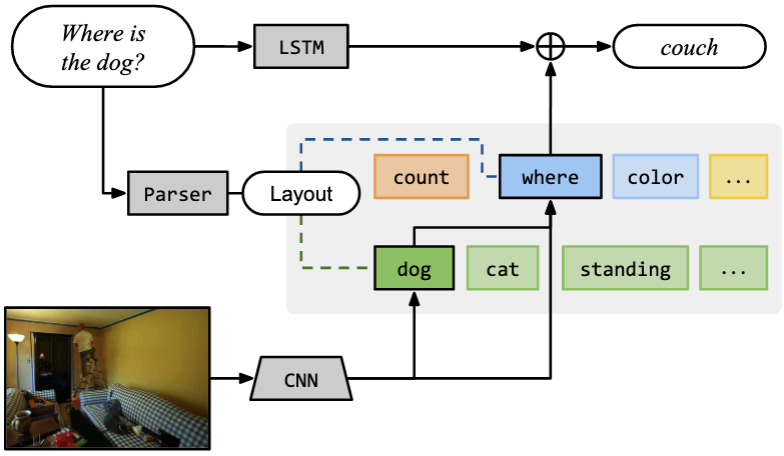


Figure 2.9. – Architecture of Neural Module Network, by Andreas et al. 2016. It is composed of a parser module, which creates from the question a layout of computation modules to apply to the image representation to answer the question.

2.2.2.3 Attention mechanism and transformer architecture

Recently, the attention mechanism has been used to model the VQA task. Attention is notoriously used in the transformer architecture, initially proposed for

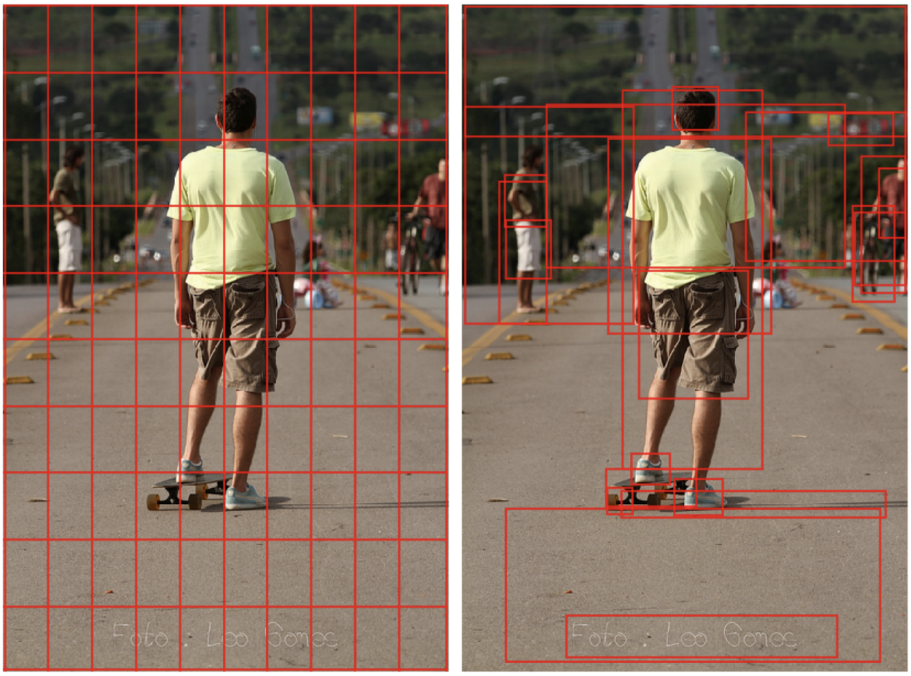


Figure 2.10. – Grid-like features versus Object Detection features. Illustration from Anderson et al. 2018a.

language modeling (Vaswani et al. 2017). Transformer architectures have been shown to be very effective for many tasks, including vision and language tasks, as they can model long-range dependencies better than LSTM and CNN (Vaswani et al. 2017).

Some works like (Yang et al. 2016; Anderson et al. 2018a; Y. Jiang et al. 2018) first included attention layers in the original VQA pipeline, as a form of fusion between the image and text embeddings. The attention is now cross-modal, between text and image tokens. Anderson et al. 2018a also proposed a novel image encoding method. Previous works were mostly using the "grid-like" features from a pre-trained convolutional neural network. Instead, they propose to use a pre-trained object detection model, Faster R-CNN (Ren et al. 2015), to extract object-based features: each detected object is associated with a position, a feature vector and a label. Most VQA questions are based on objects, and this approach significantly improves the performances of VQA models. We display an illustration of this approach in Figure 2.10.

Yu et al. 2019 then proposed MCAN, a modified transformer architecture for VQA that directly takes the image and question embeddings as input. It is made of a series of Transformer encoder-decoder blocks, with cross-attention to merge the image and question representations. The architecture is displayed in Figure 2.11. This model is trained end-to-end on the VQA task. The study of multimodal

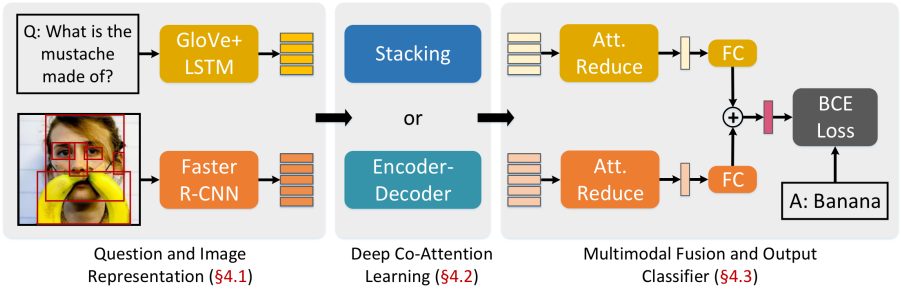


Figure 2.11. – Architecture of MCAN, by Yu et al. 2019. After the image and text embeddings using an LSTM and a Faster R-CNN, a series transformer block with cross-attention is used to merge the image and question representations.

transformer architectures, with cross-modal attention to merge modalities is an active research area today.

2.2.2.4 Large pre-trained / Foundation models

The most recent works in VQA leverage *large-scale pretraining* on vision-and-language datasets. This approach is inspired by the success of large pre-trained language models for NLP such as BERT (Devlin et al. 2019) and the GPT models (Radford et al. 2018; Radford et al. 2019; Brown et al. 2020). Those text models are trained on a large amount of data, using unsupervised objectives, as explained in Section 2.1.2. This approach was also used for Computer Vision, using contrastive learning (Grill et al. 2020; Caron et al. 2021) or masked modeling (He et al. 2022).

Those models are sometimes referred to as *foundation models*: they serve as a base for evaluation or fine-tuning on many downstream tasks.

For vision-and-language, this includes LXMERT (Tan and Bansal 2019), UNITER (Y.-C. Chen et al. 2019), OSCAR (X. Li et al. 2020), FLAVA (Singh et al. 2022), OFA (P. Wang et al. 2022), and many others. They are trained on large-scale vision-and-language datasets such as COCO (T.-Y. Lin et al. 2014a), Visual Genome (Krishna et al. 2017a), Conceptual Captions (Sharma et al. 2018), and many other datasets. They are trained on tasks like Masked Language Modeling, Masked Image Modeling, Image-Text matching, Cross-modal alignment, and Image classification, as well as unimodal tasks like image classification, or language modeling.

One of the most recent approaches, OFA, is called a unified model: it is pre-trained on a variety of text-only, image-only and multimodal tasks, using no task-specific head: it unifies all task outputs into a single vocabulary and can perform text or image generation, object detection, VQA, and many other tasks as shown in Figure 2.12. It can then be used in a zero-shot fashion if the downstream

task is similar to the ones it was trained on, or fine-tuned on a specific task to gain additional performance. Its architecture is an encoder-decoder transformer model. Its transformer weights are initialized using the BART (Lewis et al. 2019), a pre-trained language model, and a CNN is added to pre-process the image modality. It is then trained further on image-only, text-only and multimodal tasks. All the code and weights of pre-trained models are available online, which makes it possible to evaluate or fine-tune the model on many vision-and-language downstream tasks, such as Visual Question Answering. The architectures released cover a wide range of model sizes, from 33M to 930M parameters,

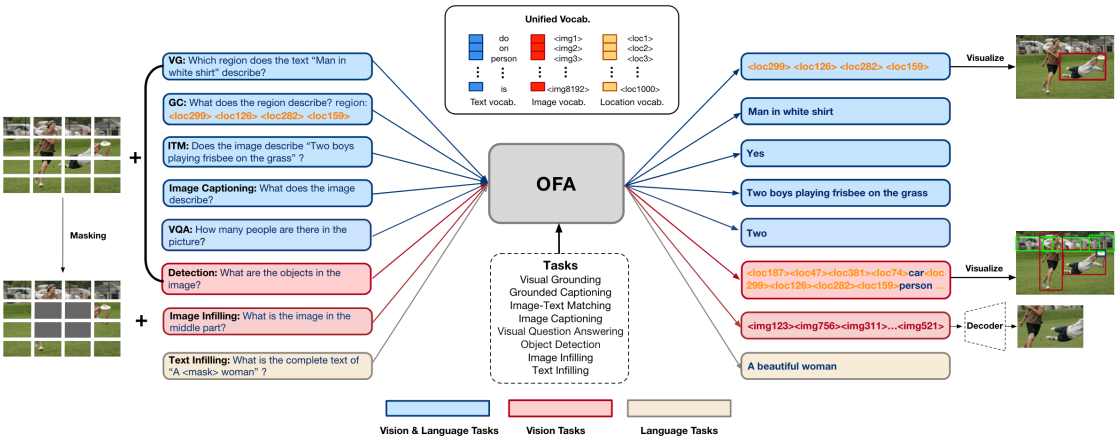


Figure 2.12. – Architecture and pre-training tasks of OFA (P. Wang et al. 2022) It is an encoder-decoder transformer model with a unified vocabulary for images, text and bounding box locations.

2.2.3 Progress in the VQA task

We display in Figure 2.13 the progress made in recent years on the VQA v2 dataset (Goyal et al. 2017a). We observe there has been a steady improvement over the years, the first models having around 65% accuracy, reaching now over 80%. We also observe the trends in model types: the first models were based on bilinear fusion (Fukui et al. 2016a; Ben-Younes et al. 2017b; J.-H. Kim et al. 2018), then the attention models appeared (Anderson et al. 2018c; Y. Jiang et al. 2018; Yu et al. 2019; H. Jiang et al. 2020), and the best models today are the large pre-trained vision-and-language transformers (Z. Wang et al. 2022; P. Wang et al. 2022; W. Wang et al. 2022). We also observe an increasing trend in the number of parameters of the models, the most recent models reaching over 1 billion parameters. This was made possible by the use of large pre-training datasets. For example, BeIT-3 is trained on 21 million image-text pairs, plus 14M

of image and $160GB$ of text. We observe that most recent models now surpass human accuracy on VQA. While this suggests that the task is now solved, we will see in the next section that might not really be the case: the VQA task contains a lot of *shortcuts*, that enable models to reach high accuracy without necessarily using the right mechanism to answer.

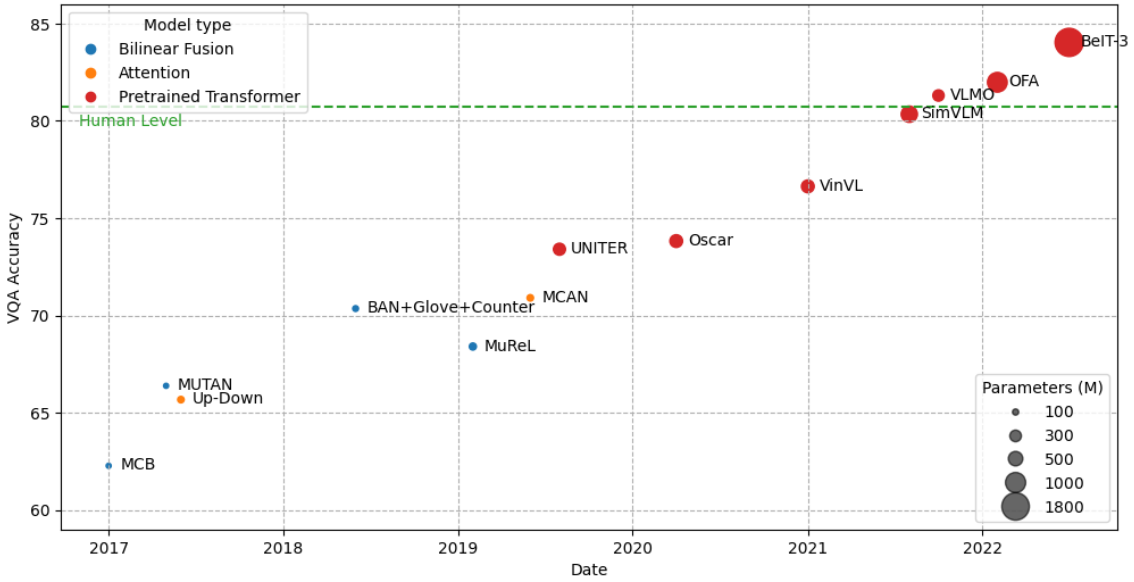
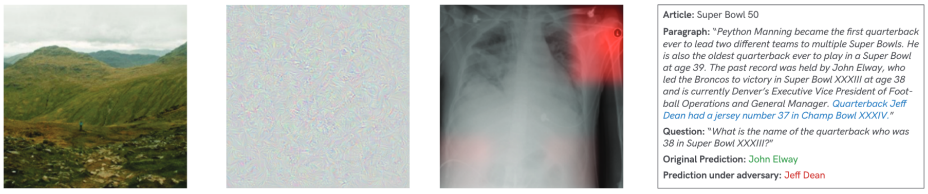


Figure 2.13. – Progress over the years for the VQA v2 dataset test-std split. Scores are from the VQA v2 Leaderboard. Only single models are reported (no ensemble).

2.3 Shortcut learning and biases

Shortcut Learning is a common problem in Deep Learning (Geirhos et al. 2020): real-world datasets display some form of inherent biases due to their data acquisition process (Gordon and Van Durme 2013; Chao et al. 2018; Torralba and Efros 2011a). As a result, machine learning models tend to reflect these biases because they capture often undesirable correlations, or shortcuts, between the inputs and the ground truth annotations (Stock and Cisse 2018a; Jia et al. 2018a; Manjunatha et al. 2019a; Torralba and Efros 2011b; Jia et al. 2018b). We can see in Figure 2.14 examples of potential shortcuts in image tasks: in the first example, for captioning, the model learns to recognize the background, instead of the primary object, as those appear together most of the time. The model then fails in a case where the background appears without the common object. In this thesis, **we use the terms *shortcut* and *bias* interchangeably**. This is a misnomer, as bias



Task for DNN	Caption image	Recognise object	Recognise pneumonia	Answer question
Problem	Describes green hillside as grazing sheep	Hallucinates teapot if certain patterns are present	Fails on scans from new hospitals	Changes answer if irrelevant information is added
Shortcut	Uses background to recognise primary object	Uses features irrecognisable to humans	Looks at hospital token, not lung	Only looks at last sentence and ignores context

Figure 2.14. – Shortcut learning: the model learns to exploit the correlation between the texture and the class of the object. From Geirhos et al. 2020

.

in statistical learning is a more general term, but they are both used in the VQA literature to describe a similar issue.

Shortcut learning is related to the notions of **causality** and **out-of-distribution (OOD)**. In statistical learning, the usual assumption is that the data are *i.i.d.* (independent and identically distributed). This means that all the data are sampled from the same distribution, especially since the testing set has the same distribution as the training set. This is rarely the case in the real world: environments are constantly evolving, and new situations will occur that were not present in the training set. For a model to be reliable in both in-distribution and out-of-distributions, it needs to learn *causal features* to be able to *generalize* well. When a model is learning shortcuts from the training distribution, it might reach a good performance when evaluated on in-distribution data, that contains the same biases. This will however lead to poor generalization performance on OOD data, or when used in the real world on a large scale. Deep Learning models tend to learn the simplest solution—a property called *simplicity bias*—(Arpit et al. 2017; Valle-Perez et al. 2018; Soudry et al. 2018; Kalimeris et al. 2019; Shah et al. 2020). This is often desirable and helps generalization to in-distribution data, but in some cases, this makes the model learn only superficial and spurious features, and will lead to poor generalization performance on OOD data (Pezeshki et al. 2021). For instance, in image classification tasks, models were shown to be often more biased towards the texture of objects than their shape (Geirhos et al. 2019). We show an example in Figure 2.15.

Shortcuts might be harmful in many ways. They can lead to dangerous failures when deployed in the real world, and can also reinforce harmful social biases towards gender or race (Zhao et al. 2017b; Hendricks et al. 2018b)

(a) Texture image
81.4% **Indian elephant**
10.3% indri
8.2% black swan

(b) Content image
71.1% **tabby cat**
17.3% grey fox
3.3% Siamese cat

(c) Texture-shape cue conflict
63.9% **Indian elephant**
26.4% indri
9.6% black swan

Figure 2.15. – A Standard ResNet-50 trained on ImageNet. The model is biased towards the texture of an object and classifies the last example as an elephant instead of a cat. From Geirhos et al. 2019.

Procedures exist to identify certain kinds of biases and to reduce them. For instance, some methods are focused on gender biases (Hendricks et al. 2018a; Zhao et al. 2017a), some others on the human reporting biases (Misra et al. 2016), and also on the shift in distribution between lab-curated data and real-world data (Gupta et al. 2018). In the language and vision context, some works evaluate unimodal baselines (Anand et al. 2018; Thomason et al. 2019) or investigate how language priors create object hallucinations (Rohrbach et al. 2018).

No general method to reduce shortcut learning without additional information It is not possible to distinguish causal from spurious correlations in a fixed dataset (Schölkopf et al. 2021). Thus, extra information or *inductive biases* is required to guide the learning and improve the generalization to out-of-distribution data. Geirhos et al. 2020 proposes to classify the inductive biases of models that have an impact on shortcut learning into four components: architecture, training data, loss function and optimization procedure. Thus, the knowledge we have about a task can be incorporated into the final model by changing one of these components.

2.3.1 Measuring shortcut learning in neural networks

We distinguish two challenges here: the first is, given a neural network already trained on a fixed dataset, how can we evaluate if it has learned the correct mechanism, using causal features, or if its decisions are based on shortcuts? The second setup is how can we evaluate the inductive biases of a training procedure and architecture.

Evaluating a trained model For the first challenge, there were multiple proposed solutions. The first approach consists in leveraging explainability methods (Ribeiro et al. 2016; Fong and Vedaldi 2017; Stock and Cisse 2018b; Manjunatha et al. 2019b), such as attributions methods like LIME. Those methods highlight parts of the input that were important in the model's prediction. For instance, LIME (Ribeiro et al. 2016) shows which pixels from an image contribute the most to the classification output. A user can then use this to determine if the model is using spurious correlations, like using the background to predict the class of the object. We show an example of this in Figure 2.16 These methods often require the intervention of a human or the collection of expensive annotations (Das et al. 2017a), but don't require much prior knowledge of the source of the bias.

(a) Husky classified as wolf (b) Explanation

Figure 2.16. – Explanation of a model's prediction using LIME for the task "Wolf vs Husky". The model uses the background to predict the class, as wolves appear usually in the snow, while huskies are usually in the grass. Here, a husky appears in the snow, and the model incorrectly predicts it as a wolf. From Ribeiro et al. 2016.

Another strategy to evaluate shortcut learning is to create out-of-distribution evaluation datasets that do not contain the biases that need to be avoided, or that we hypothesize the system to exploit. (McCoy et al. 2019; Alcorn et al. 2019). It simulates the kind of shifts in distribution that can potentially be encountered when deployed in the real world. For example, the FairFace dataset (Kärkkäinen and Joo 2019) has multiple groups of faces with various races, genders and ages to evaluate face analysis models. ImageNet-C (Hendrycks and Dietterich 2019) is a benchmark that contains images from ImageNet, but with a specific corruption applied to them. The corruption is chosen to be a shortcut that the model might learn to exploit. It makes it possible to evaluate how well the models rely on low-level features that are not always relevant to the task. ObjectNet (Barbu et al. 2019) is a benchmark that contains images from ImageNet, but with unusual backgrounds, object poses or viewpoints.

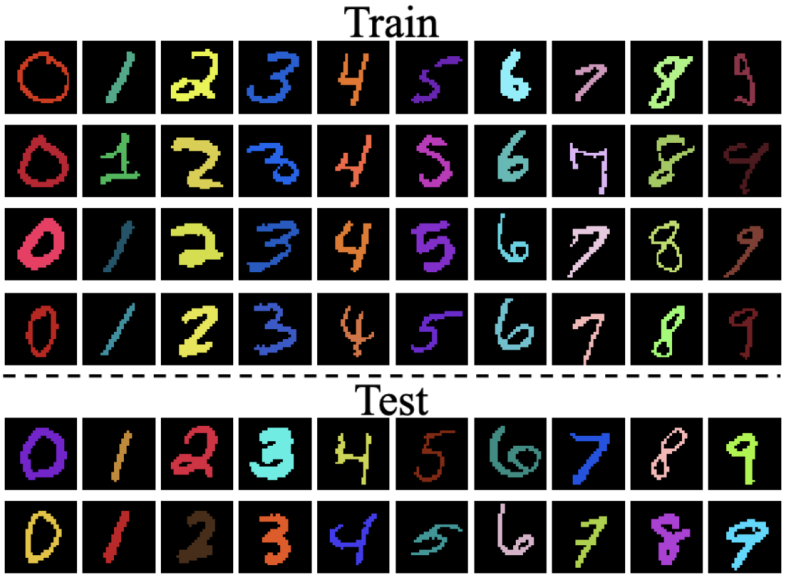


Figure 2.17. – Colored-MNIST dataset. The color of the digit is correlated with its label, but the correlation is reversed from the training set to the testing set. Image from B. Kim et al. 2019.

Evaluating the inductive biases of a training procedure and architecture on controlled biases In this challenge, the objective is to evaluate how much a given training procedure and model architecture are sensitive to a certain class of shortcuts. This is related to domain generalization. A simple way to achieve this is to control the source of biases: the "biased" variable is fixed and a pair of (training, testing sets) that do not contain the same correlation between this variable and the answer.

An example displayed in Figure 2.17 is Colored-MNIST (Arjovsky et al. 2019; B. Kim et al. 2019). It is a toy dataset where the color of the digit is correlated with its label, but the correlation is reversed from the training set to the testing set. Therefore, a model using the color of the digit to predict the label will perform poorly on the testing set. Another example is Biased Activity Recognition (BAR) (Nam et al. 2020). It is an activity recognition dataset, where the activity is correlated with the background scene, but the correlations are changed in the testing set.

DomainBed (Gulrajani and Lopez-Paz 2021) is another benchmark for domain generalization. It contains seven datasets, each containing at least three domains with different biases. Models learn on N-1 datasets and are tested on the held-out domain. This benchmark is designed to 'test if models are able to learn *invariant* correlations that hold across all domains and generalize to the test domain, or if they learn spurious domain-specific correlations that do not generalize to the test domain.

Those methods make it possible to study how inductive biases influence shortcut learning in deep neural networks. But they require to control the source of biases. Next, we will explore methods aiming at reducing shortcut learning.

2.3.2 Reducing shortcut learning

Geirhos et al. 2020 explain what components, or "inductive biases" have an impact on shortcut learning: the model architecture, the training data, the loss function and the optimization algorithm. Thus, modifying those components might help to reduce shortcut learning. One important thing to understand is that to reduce shortcut learning, we need to make a hypothesis about the source of the biases, or add domain knowledge to the training procedure or the architecture on how the task should be solved to make it more robust to those biases. Some methods also use bias labels to reduce their impact in the final model.

And bias-reduction methods will often degrade performance on in-distribution testing sets, as they will reduce the ability of the model to exploit the shortcuts that are common between the training and the testing set.

Some methods include (for multi-environment / bias label) GROUP-DRO (Sagawa et al. 2020), IRM (Arjovsky et al. 2019), JTT (E. Z. Liu et al. 2021), LfF (Nam et al. 2020), Mixed capacity ensembles (C. Clark et al. 2020).

A different strategy is to make models "explainable by design" (Angelov 2021; Fauvel et al. 2022). This makes it possible for users to understand the decision of the model and assess its correctness. Using domain knowledge as architecture priors to the model can make it harder for the model to learn spurious correlations.

2.4 Shortcut learning in VQA

VQA is an interesting task to study shortcut learning. It requires performing reasoning, which is difficult to model: learning simple spurious correlations can be an easier way for models to achieve good performances. Additionally, it is a vision-and-language task, which makes possible the existence of complex multimodal shortcuts. The VQA v1 dataset (Antol et al. 2015a) was collected without controls on the correlations between questions, images and answers. This led to a dataset that contains many biases, that can be exploited by models: A. Agrawal et al. 2016 study the behavior of VQA models: they show that VQA models "seem to be heavily reliant on the language model, perhaps not deeply understanding the image", and that on the VQA v1 dataset(Antol et al. 2015a), then there is not

a large gap between the performance of the question-only model and a regular VQA model. Additionally, they show that VQA models rely mostly on the first few words of the question. Multiple approaches have been proposed to reduce biases in VQA datasets and models. Most works focus on reducing the learning of spurious correlations between the question and the answer, to force models to rely more on the image. We classify the methods following three types of "inductive biases" proposed by Geirhos et al. 2020: **Training data**, **architectural priors** and **learning strategies**.

2.4.1 Methods to reduce biases in Visual Question Answering

2.4.1.1 Acting on training data to reduce biases

VQA v2 dataset A first approach that tackles those issues is to change the inductive biases contained in the training data. The VQA v2 dataset (Goyal et al. 2017a), follows this approach: They build a more balanced dataset, where it is harder to answer a question using only the image. It is built by collecting complementary images such that every question is associated with a second image leading to a different answer. This approach is expensive, and it can be difficult to collect the rare examples required to reduce biases but is partially effective to reduce simple question-answer shortcuts. We display examples from this dataset in Figure 2.18. The VQA v2 dataset contains 443K train, 214K val and 453K test pairs of images and questions.

Figure 2.18. – Pair of examples from the VQA v2 dataset. Each question is associated with two images with different answers. Image from Goyal et al. 2017a.

We observe in Figure 2.19 that overall, the answer distributions associated with a question type are more balanced for this new dataset. This gives less opportunity for models to learn to answer the questions without analyzing the image.

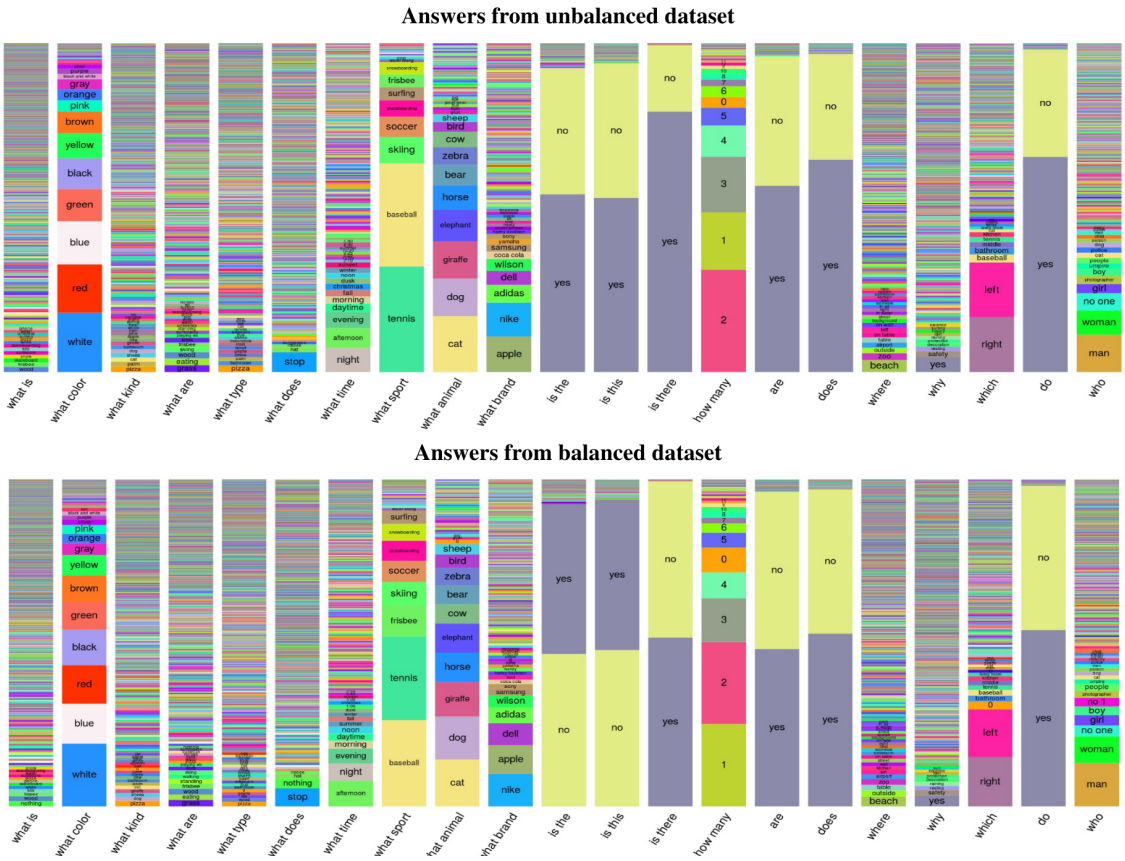


Figure 2.19. – Comparison of the distributions of answers between VQA v1 and VQA v2. Illustration from (Goyal et al. 2017a).

Other datasets and methods Other more balanced datasets have been created. For instance, the synthetic datasets for VQA (Justin Johnson et al. 2017b; Drew A Hudson and Manning 2019) minimize question-conditional biases via rejection sampling within families of related questions to avoid simple shortcuts to the correct answer.

Other methods of reducing biases by modifying the training data include data augmentation techniques. RandImg (Teney et al. 2020b) proposes to replace the image with a random image and maximize the loss, the model avoids trusting too much the textual input without looking at the image. Other works create counterfactual samples (L. Chen et al. 2020; Teney et al. 2020a), or leverage explanations to guide the model to focus on the right regions (Selvaraju et al. 2019).

2.4.1.2 Architectural priors to reduce shortcut learning

Another approach is to incorporate architectural priors in models to encourage learning the correct mechanisms. This is related to explainability *by design*: enforcing the architecture to use the correct mechanism to answer the question and making it explainable are often related and can benefit from the same domain knowledge. The GVQA model (A. Agrawal et al. 2018a) contains restrictions in the architecture to prevent the learning of question biases. First, a *Visual Concept Classifier* extracts a set of visual concepts from the image that are relevant to the given question. In parallel, another model extracts a group of possible answers from the question, such as "object", "color" or others. Then, an answer predictor merges those two pieces of information to predict the correct answer. This makes it harder for the model to select an answer directly from the question and forces it to analyze the image. The architecture of this model is displayed in Figure 2.20. This architecture improves results on the VQA-CP dataset compared to previous VQA models but reduces significantly the accuracy on in-distribution data. Note that this approach requires training multiple sub-models separately, and is not trained in an end-to-end fashion.

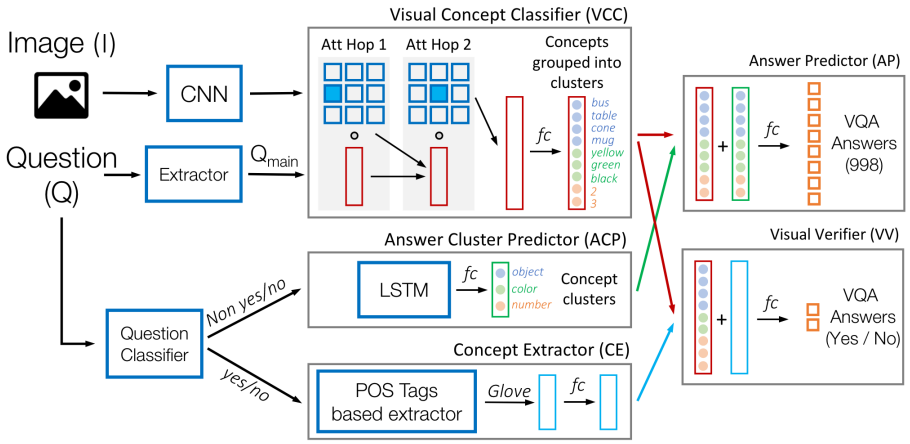


Figure 2.20. – The GVQA architecture from A. Agrawal et al. 2018a. Two models are learned separately: the first one extracts visual concepts from the answer and creates a list of possible answers. The second one extracts categories of answers from the question. The two predictions are then merged to produce the final answer.

2.4.1.3 Learning strategies to reduce shortcut learning

Ramakrishnan et al. 2018 propose a learning strategy to reduce shortcut learning, displayed in Figure 2.21. First, they add an adversarial loss that penalizes the question encoder if it can predict the answer only from the question. Then they

add an entropy regularization loss that forces the output distribution from the full model to have lower entropy than the output distribution from the question-only model, to encourage the use of the additional information contained in the image. This strategy improves the accuracy of their baseline model by a few points. Kervadec et al. 2020 proposes to use semantic loss that penalizes differently the answers based on their semantic similarity with the ground truth answer. For example, if the ground truth answer is "red", the model will be penalized more if it predicts "blue" than if it predicts "pink".

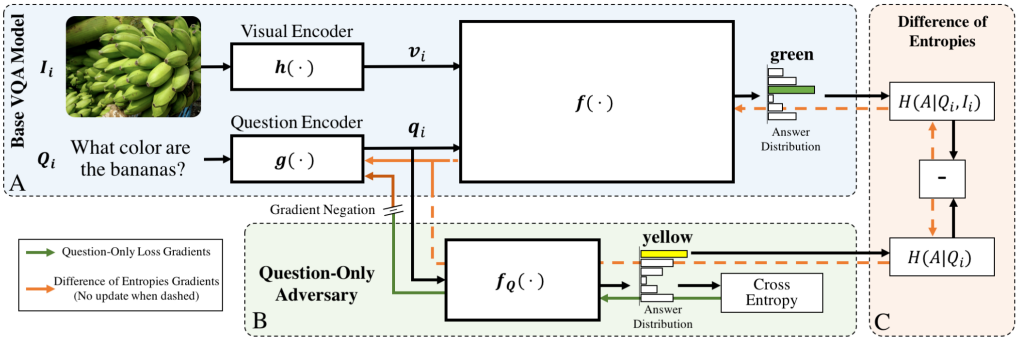


Figure 2.21. – The Q-Adv + DoE adversarial strategy. An Adversarial loss is added that prevents predicting the answer using only the question encoder. Additionally, a *Difference of Entropy* loss encourages the predictions from the question-only model and the main model to have a different distribution. Image from Ramakrishnan et al. 2018.

2.4.2 Benchmarks to measure shortcut learning in VQA

VQA-CP A strategy to evaluate a model's reliance on shortcuts is to evaluate it on examples that contradict the shortcuts. But this is not always possible, as shortcuts can be subtle and hard to detect. Therefore, a simpler strategy can be to create a training set with known shortcuts and evaluate the model's performance on a testing set that does not follow those shortcuts. This is the approach used in VQA-CP (A. Agrawal et al. 2018a). In this dataset, the distribution of answers conditioned on the question type is different between the training and the testing split. Therefore, a model using only the question and not the image to answer the question will perform much worse on the testing split. We show the results from their work in Figure 2.23. All existing models suffer from a huge drop in performance compared to their scores on the original VQA setting.

GQA-OOD The GQA-OOD dataset (Kervadec et al. 2021) similar goal as VQA-CP: propose a testing set for the GQA dataset that has a different distribution from

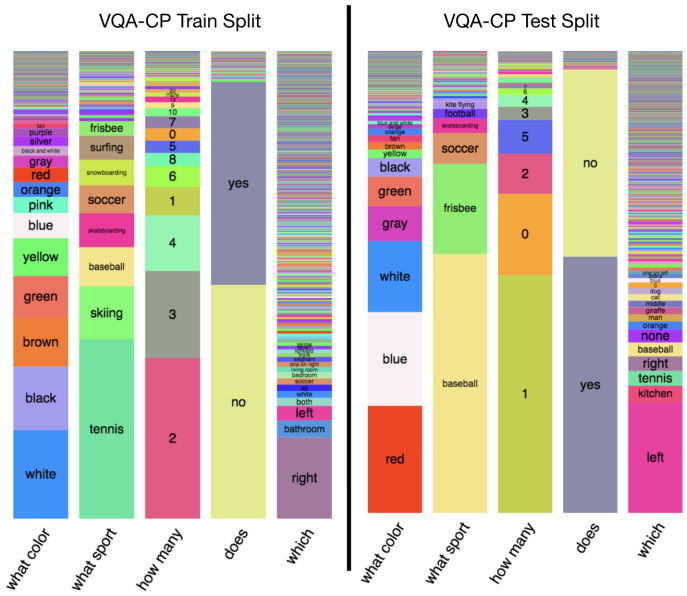


Figure 2.22. – The VQA-CP v2 dataset training and testing distribution, per question type. Image from A. Agrawal et al. 2018a.

the GQA training set but corrects some of its shortcomings. First, it has a separate validation and testing set. This makes it possible to select hyperparameters and avoid adaptive overfitting. Then, it doesn't propose a different training set: all models trained on the regular GQA dataset can be tested on this benchmark.

VQA-Hat The VQA-HAT (Human Attention) dataset (Das et al. 2017a) is designed to evaluate if the VQA models focus on the same regions as a human would do when answering the same question. It can be used as a proxy to evaluate if the model is learning textual shortcuts without using the image information.

Model	Dataset	Overall	Yes/No	Number	Other	Dataset	Overall	Yes/No	Number	Other
per Q-type prior [5]	VQA v1	35.13	71.31	31.93	08.86	VQA v2	32.06	64.42	26.95	08.76
	VQA-CP v1	08.39	14.70	08.34	02.14	VQA-CP v2	08.76	19.36	11.70	02.39
d-LSTM Q [5]	VQA v1	48.23	79.05	33.70	28.81	VQA v2	43.01	67.95	30.97	27.20
	VQA-CP v1	20.16	35.72	11.07	08.34	VQA-CP v2	15.95	35.09	11.63	07.11
d-LSTM Q + norm I [24]	VQA v1	54.40	79.82	33.87	40.54	VQA v2	51.61	73.06	34.41	39.85
	VQA-CP v1	23.51	34.53	11.40	17.42	VQA-CP v2	19.73	34.25	11.39	14.41
NMN [3]	VQA v1	54.83	80.39	33.45	41.07	VQA v2	51.62	73.38	33.23	39.93
	VQA-CP v1	29.64	38.85	11.23	27.88	VQA-CP v2	27.47	38.94	11.92	25.72
SAN [39]	VQA v1	55.86	78.54	33.46	44.51	VQA v2	52.02	68.89	34.55	43.80
	VQA-CP v1	26.88	35.34	11.34	24.70	VQA-CP v2	24.96	38.35	11.14	21.74
MCB [11]	VQA v1	60.97	81.62	34.56	52.16	VQA v2	59.71	77.91	37.47	51.76
	VQA-CP v1	34.39	37.96	11.80	39.90	VQA-CP v2	36.33	41.01	11.96	40.57

Figure 2.23. – The VQA-CP dataset results for existing VQA models at the time of its release. From A. Agrawal et al. 2018b.

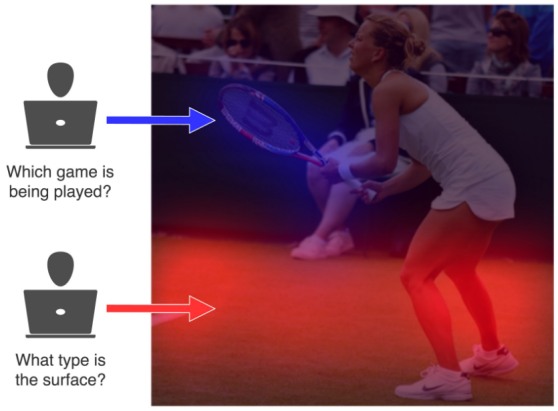


Figure 2.24. – Example from the VQA-Hat dataset. Humans do not focus on the same region of the image depending on the question being asked. We can compare the attention maps of the VQA models to those of human attention. Image from Das et al. 2017a.

2.5 Positioning

In this thesis, we propose multiple contributions related to the topics of Visual Question Answering and Shortcut Learning. We show a figure of how our contributions relate to each other in Figure 2.25.

1. At the start of this thesis, the VQA-CP dataset had been proposed (A. Agrawal et al. 2018b), and the state-of-the-art approach was Q-Adv (Ramakrishnan et al. 2018), which attained slight improvements in results over the baseline model. We present both those works in Section 2.4. The loss they propose affects the question encoder, preventing it from learning shortcuts, but does not impact the main model. We propose a learning method that discourages the whole model from learning question-based shortcuts (Cadene et al. 2019c).
2. Then, we focus on Visual Counting a sub-task of Visual Question Answering. Previous models for this task are trained with a classification loss, predicting numbers as classes: they do not take into account the structure of the output domain (M. Acharya et al. 2019; Y. Zhang et al. 2018). This makes them more susceptible to learning shortcuts. Additionally, previous datasets did not have out-of-distribution evaluation sets to evaluate shortcut learning. (M. Acharya et al. 2019; Trott et al. 2018). We propose to analyze shortcut learning in visual counting, a benchmark to evaluate shortcut learning in counting models, and propose a new architecture that contains inductive biases to help alleviate this issue (Dancette et al. 2021a).
3. We then study the problem of multimodal shortcuts: most if not all works on VQA focus on question-based biases, i.e. shortcuts between the question words and the answer. We propose the first analysis of multimodal (question-image) shortcuts in VQA, and a new benchmark to evaluate whether VQA models learn those shortcuts (Dancette et al. 2021b).
4. Finally, we investigate the reliability of VQA models in the context of out-of-distributions example: how to estimate the confidence of answers returned by the model, and abstain in case it is too low. We build upon the work of Whitehead et al. 2022a, on reliability for VQA in-distribution. A model might have high reliability (i.e. the ability to estimate its own confidence) on in-distribution test sets, partially because they can use all spurious correlations, but give wrong confidence values for out-of-distributions examples where the spurious correlations do not hold.

In addition to those contributions, we show in Figure 2.25 our work on domain generalization Fishr (Rame et al. 2022), where we propose a new learning strategy to learn invariant models across multiple training distributions, in order to

generalize to a new testing distribution. We also worked on training efficiency of vision transformer architectures (Dancette and Cord 2022).

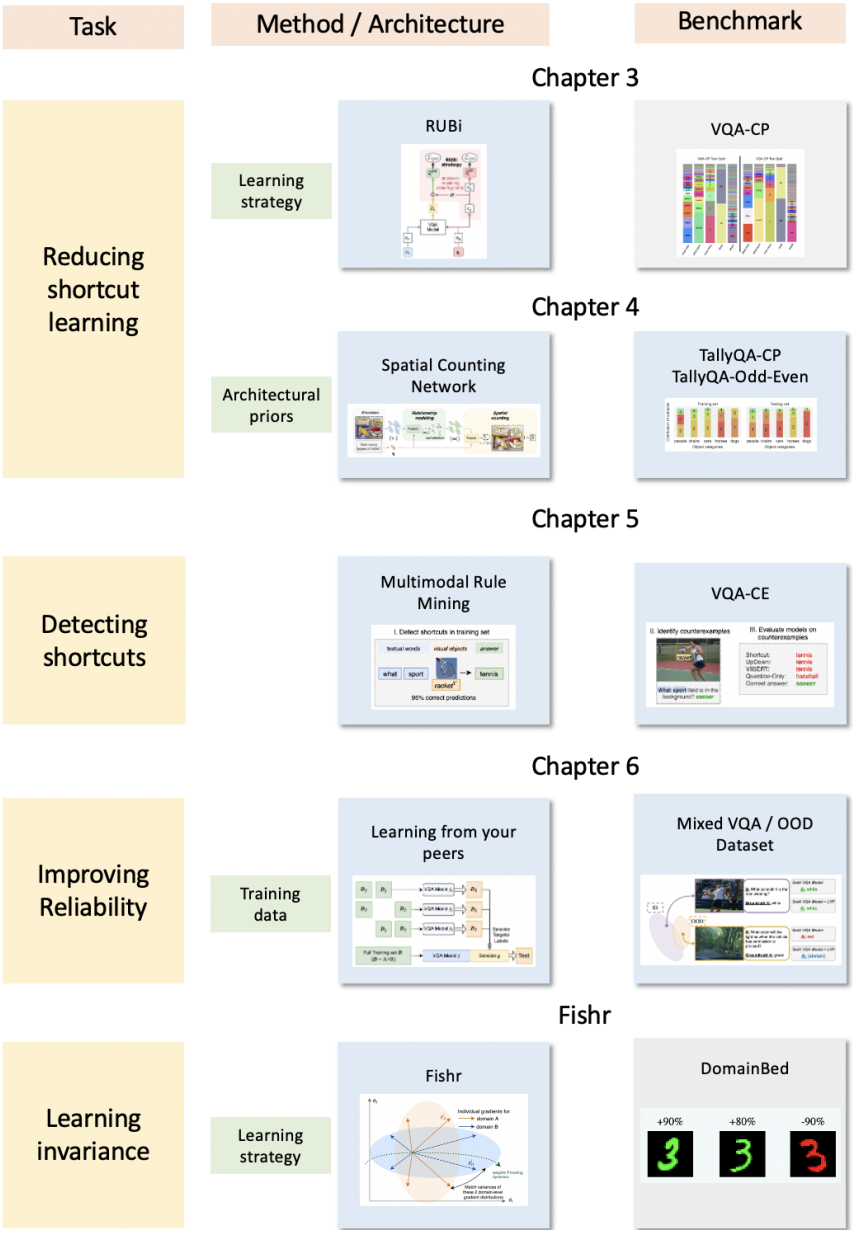


Figure 2.25. – Contributions of this thesis. Our contributions are in blue boxes. Grey boxes are prior works.

CHAPTER 3

A LEARNING STRATEGY TO REDUCE UNIMODAL BIASES IN VQA

Chapter abstract

*In this chapter, we propose to explore a first strategy to shortcut in Visual Question Answering (VQA): **loss functions**. We propose a learning strategy to reduce the importance of the most biased examples, i.e. examples that can be correctly classified using only the question, i.e. without looking at the image. It implicitly forces the VQA model to use the two input modalities instead of relying on statistical regularities between the question and the answer. We leverage a question-only model that captures language biases to identify these unwanted regularities. This model is learned in parallel with the VQA model and prevents it from learning biases by influencing its predictions. This leads to dynamically adjusting the loss in order to compensate for biases. We validate our contributions by surpassing the reference methods on VQA-CP v2.*

The work in this chapter has led to the publication of this conference paper (denotes equal contribution)*:

- Remi Cadene*, Corentin Dancette*, Hedi Ben-Younes, Matthieu Cord, and Devi Parikh (2019). "RUBi: Reducing Unimodal Biases for Visual Question Answering". In: *Advances in Neural Information Processing Systems (NeurIPS)*

Contents

3.1 Introduction

As we reported in Chapter 2, VQA models achieve impressive results on the VQA v2 benchmark. However, we explained in Section 2.4 that they tend to exploit statistical regularities between answer occurrences and certain patterns in the question. Those models are designed to merge information from both modalities, but in practice, they often answer mainly using the question modality. When most of the bananas are *yellow*, a model does not need to learn the correct behavior to reach high accuracy for questions asking about the color of bananas. Instead of looking at the image, detecting a banana and assessing its color, it is much easier to learn from the statistical shortcut linking the words *what*, *color* and *bananas* with the most occurring answer *yellow*. We illustrate this issue in Figure 3.1: a model answering the question "What color is the banana?", that seen during its training 80% of yellow bananas, will most likely answer *yellow*, even if the banana is green like in figure in the middle. Thus, as the right figure shows, there is a crucial need to develop new strategies to reduce the amount of biases coming from the question modality in order to learn better behaviors. As we reported in Section 2.3, one of the inductive biases that influence a model's behavior is the **loss function**. We explore this direction for VQA.

One way to quantify the amount of statistical shortcuts from each modality is to train unimodal models. For instance, a question-only model trained on the widely used VQA v2 dataset predicts the correct answer approximately 44% of the time over the test set. We propose a learning strategy that takes advantage

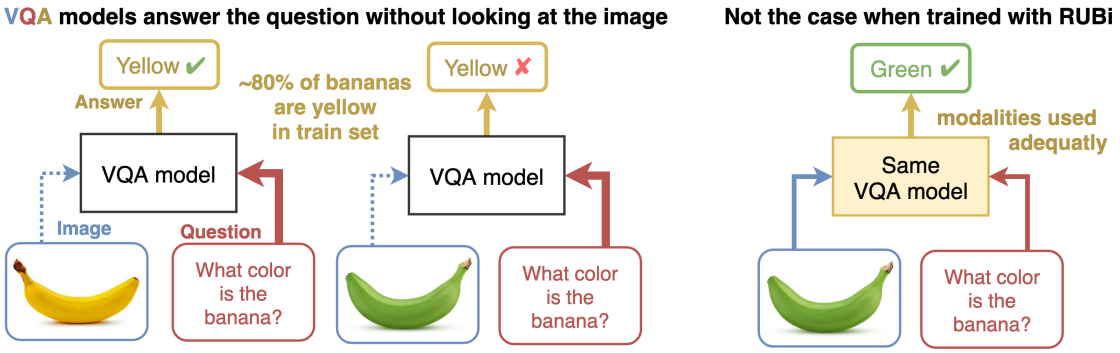


Figure 3.1. – As depicted, current VQA models often rely on unwanted statistical correlations between the question and the answer instead of using both modalities. We aims at reducing the amount of unimodal biases learned by a VQA model during training.

of this to reduce the amount of biases learned by VQA models that we call RUBi – *Reducing Unimodal Biases*. Our strategy reduces the importance of the most biased examples, i.e. examples that can be correctly classified without looking at the image modality. It implicitly forces the VQA model to use the two input modalities instead of relying on statistical regularities between the question and the answer. We take advantage of the fact that question-only models are by design biased towards the question modality: we add a question-only branch on top of a base VQA model during training only. This branch influences the VQA model, dynamically adjusting the loss to compensate for biases. As a result, the gradients backpropagated through the VQA model are reduced for the most biased examples and increased for the less biased ones. At the end of the training, we simply remove the question-only branch.

In Section 3.2, we review related work on mitigating shortcut learning for VQA. In Section 3.3, we describe our RUBi learning strategy. In Section 3.4, we evaluate our approach on multiple models and standard benchmarks, VQA-CP v1 and v2. Finally, in Section 3.4.4, we evaluate the impact of RUBi on visual grounding – the ability of models to use the correct image regions to answer the question.

3.2 Related work

In the following, we discuss related works that assess and reduce unimodal biases learned by VQA models. We discuss VQA models and datasets in Section 2.2, and give a general introduction on shortcut learning in Section 2.3, and especially for VQA in Section 2.4.

Assessing unimodal biases in datasets and models Despite being designed to merge the two input modalities, it has been found that VQA models often rely on superficial correlations between inputs from one modality and the answers without considering the other modality (Jabri et al. 2016; Manjunatha et al. 2019a). An interesting way to quantify the amount of unimodal biases that can potentially be learned by a VQA model consists of training models using only one of the two modalities (Antol et al. 2015a; Goyal et al. 2017a). The question-only model is a particularly strong baseline because of the large amount of statistical regularities that can be leveraged from the question modality. With the RUBi learning strategy, we take advantage of this baseline model to prevent VQA models from learning question biases.

Unfortunately, biased models that exploit statistical shortcuts from one modality usually reach impressive accuracy on most of the current benchmarks. VQA-CP v2 and VQA-CP v1 (A. Agrawal et al. 2018b), presented in Section 2.4.2, were introduced as diagnostic datasets containing different answer distributions for each question type between train and test splits. Consequentially, models biased towards the question modality fail on these benchmarks. We use the more challenging VQA-CP v2 dataset extensively in order to show the ability of our approach to reduce the learning of biases coming from the question modality.

Balancing datasets to avoid unimodal biases Once the unimodal biases have been identified, one method to overcome these biases is to create more balanced datasets, as presented in Section 2.4.1.1. However, even with this additional balancing done in VQA v2, statistical biases from the question remain and can be leveraged (A. Agrawal et al. 2018b). That is why we propose an approach to reduce unimodal biases during training. It is designed to learn unbiased models from biased datasets.

Architectures and learning strategies to reduce unimodal biases In parallel with these previous works on balancing datasets, an important effort has been carried out to design VQA models to overcome biases from datasets. (A. Agrawal et al. 2018b) proposed a hand-designed architecture called Grounded VQA (GVQA), presented in Section 2.4.1.2. It breaks the task of VQA down into a first step of locating and recognizing the visual regions needed to answer the question, and a second step of identifying the space of plausible answers based on a question-only branch. This approach requires training multiple sub-models separately. In contrast, our learning strategy is end-to-end. Their complex design is not straightforward to apply to different architectures while our approach is model-agnostic. While we rely on a question-only branch, we remove it at the end of the training.

The work most related to ours in terms of approach is (Ramakrishnan et al. 2018), presented in Section 2.4.1.3. The authors propose a learning strategy to overcome language priors in VQA models. They first introduce an adversary question-only branch. It takes as input the question encoding from the VQA model and produces a question-only loss. They use a gradient negation of this loss to discourage the question encoder to capture unwanted biases that could be exploited by the VQA model. They also propose a loss based on the difference of entropies between the VQA model and the question-only branch output distributions. These two losses are only backpropagated to the question encoder. In contrast, our learning strategy targets the full VQA model parameters to reduce the impact of unwanted biases more effectively. Instead of relying on these two additional losses, we use the question-only branch to dynamically adapt the value of the classification loss in order to reduce the learning of biases in the VQA model. A visual comparison between (Ramakrishnan et al. 2018) and RUBi can be found in Figure 3.4.

3.3 Reducing Unimodal Biases Approach

We consider the common formulation of the Visual Question Answering (VQA) defined in Chapter 2, Section 2.2. We consider the task as a single-label classification problem. We define additional notations: for a single example (v_i, q_i, a_i), VQA models use an image encoder $e_v : \mathcal{V} \rightarrow \mathbb{R}^{n_v \times d_v}$ to output a set of n_v vectors of dimension d_v, a question encoder $e_q : \mathcal{Q} \rightarrow \mathbb{R}^{n_q \times d_q}$ to output a set of n_q vectors of dimension d_q, a multimodal fusion $f_m : \mathbb{R}^{n_v \times d_v} \times \mathbb{R}^{n_q \times d_q} \rightarrow \mathbb{R}^{d_m}$, and a classifier $c : \mathbb{R}^{d_m} \rightarrow \mathbb{R}^{|\mathcal{A}|}$. These functions are composed as follows:

$$f(v_i, q_i) = c(f_m(e_v(v_i), e_q(q_i))) \tag{3.1}$$

Each one of them can be defined to instantiate most of the specialized VQA models, such as Bottom-Up and Top-Down Attention (UpDn) (Anderson et al. 2018a) or MUREL (Cadene et al. 2019a).

We recall the classical learning strategy of VQA models, depicted in Figure 3.2: it consists in minimizing the standard cross-entropy criterion over a dataset of size n:

$$\mathcal{L}(\theta; \mathcal{D}) = -\frac{1}{n} \sum_{i=1}^{n} \log(\text{softmax}(f(v_i, q_i)))[a_i] \tag{3.2}$$

As explained in Section 2.4, VQA models are inclined to learn unimodal biases from the datasets (A. Agrawal et al. 2018b). They do not learn to use the image information because there are too few examples in the dataset where the banana

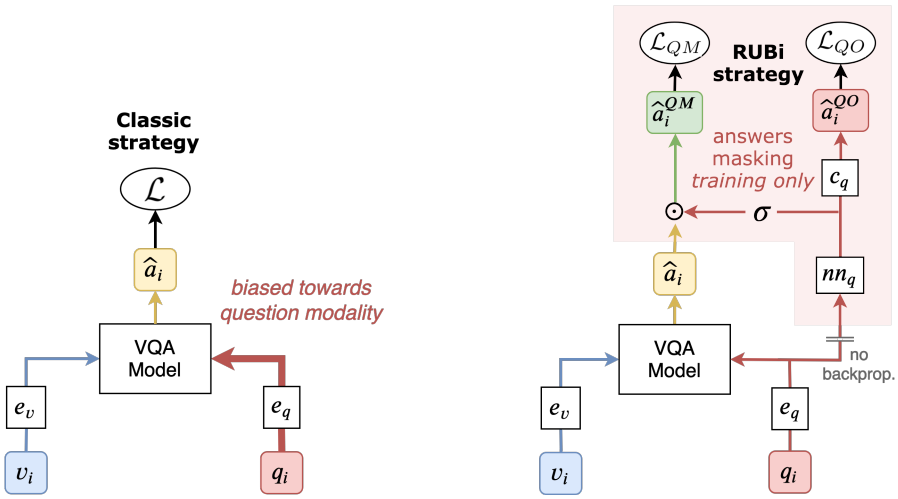


Figure 3.2. – Visual comparison between the classical learning strategy of a VQA model and our RUBi learning strategy. The red highlighted modules are removed at the end of the training. The output $\hat{a}_i$ is used as the final prediction.

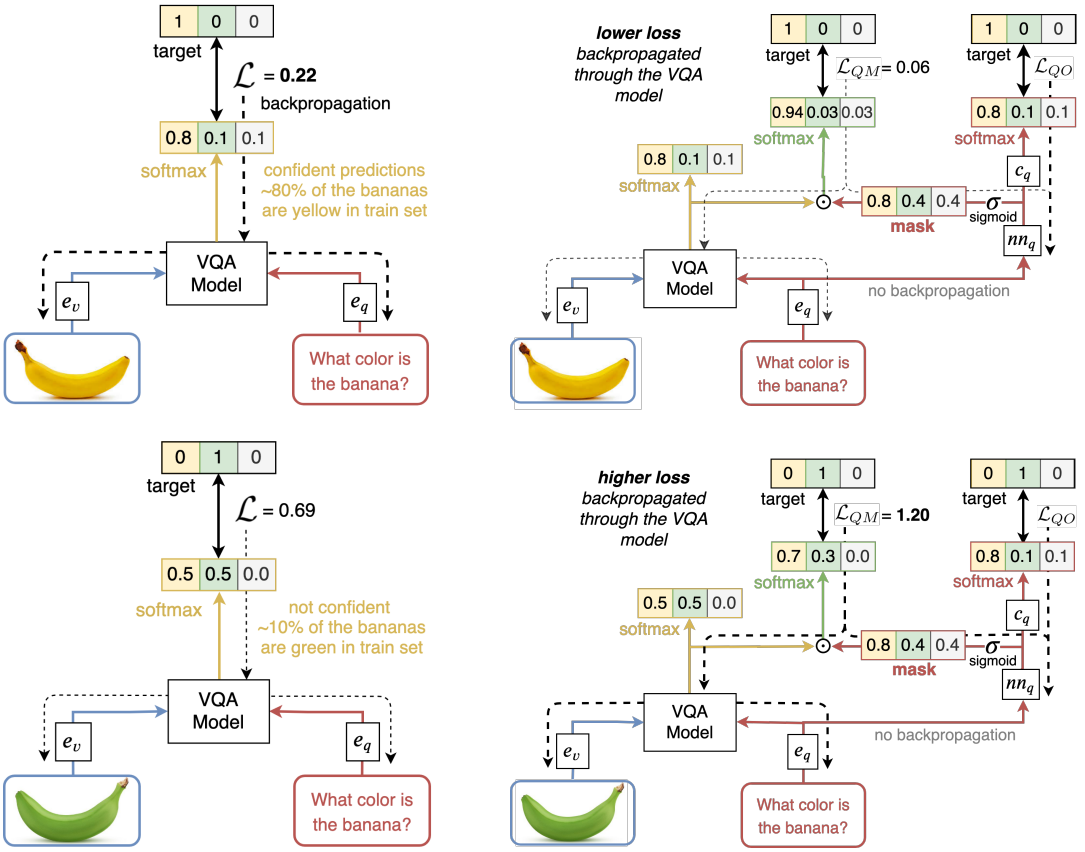
is not yellow. Once trained, their inability to use the two modalities adequately makes them inoperable on data coming from different distributions such as real-world data. Our contribution consists in modifying this cost function $\mathcal{L}$ to avoid the learning of these biases.

3.3.1 RUBi learning strategy

Capturing biases with a question-only branch One way to measure the unimodal biases in VQA datasets is to train a unimodal model which takes only one of the two modalities as input.

The key idea of our approach, depicted in Figure 3.2, is to adapt a question-only model as a branch of our VQA model, that will alter the main model's predictions. By doing so, the question-only branch captures the question biases, allowing the VQA model to focus on the examples that cannot be answered correctly using the question modality only. The question-only branch can be formalized as a function $f_Q : \mathcal{Q} \rightarrow \mathbb{R}^{|\mathcal{A}|}$ parameterized by θ_Q, and composed of a question encoder $e_q : \mathcal{Q} \rightarrow \mathbb{R}^{n_q \times d_q}$ to output a set of n_q vectors of dimension d_q, a neural network nn_q: $\mathbb{R}^{n_q \times d_q} \rightarrow \mathbb{R}^{|\mathcal{A}|}$ and a classifier c_q: $\mathbb{R}^{|\mathcal{A}|} \rightarrow \mathbb{R}^{|\mathcal{A}|}$.

$$f_Q(q_i) = c_q(nn_q(e_q(q_i))) \tag{3.3}$$



(a) Classical learning strategy (b) RUBi learning strategy

Figure 3.3. – Detailed illustration of the RUBi impact on the learning. In the first row, we illustrate how RUBi reduces the loss for examples that can be correctly answered without looking at the image. In the second row, we illustrate how RUBi increases the loss for examples that cannot be answered without using both modalities.

During training, the branch acts as a proxy preventing any VQA model of the form presented in Equation (3.1) from learning biases. At the end of the training, we simply remove the branch and use the predictions from the base VQA model.

Preventing biases by masking predictions Before passing the predictions of our base VQA model to the loss function defined in Equation (3.2), we merge them with a mask of length $|\mathcal{A}|$ containing a scalar value between 0 and 1 for each answer. This mask is obtained by passing the output of the neural network nn_q through a sigmoid function σ. The goal of this mask is to dynamically alter the loss by modifying the predictions of the VQA model. To obtain the new predictions, we simply compute an element-wise product $\odot$ between the mask and the original predictions as defined in the following equation.

$$f_{QM}(v_i, q_i) = f(v_i, q_i) \odot \sigma(nn_q(e_q(q_i))) \tag{3.4}$$

Our method modifies the predictions in this specific way to prevent the VQA model to learn biases from the question. To better understand the impact of our approach on learning, we examine two scenarios. First, we reduce the importance of the most biased examples, i.e. examples that can be correctly classified without using the image modality. To do so, the question-only branch outputs a mask to increase the score of the correct answer while decreasing the scores of the others. As a result, the loss is much lower for these biased examples. In other words, the gradients backpropagated through the VQA model are smaller, thereby reducing the importance of these examples during training. As illustrated in the first row of Figure 3.3, given the question *what color is the banana*, the mask takes a high value of 0.8 for the answer *yellow* which is the most likely answer for this question in the training set. On the other hand, the value for the other answers *green* and *white* are smaller. We see that the mask influences the VQA model to produce new predictions where the score associated with the answer *yellow* increases from 0.8 to 0.94. Compared to the classical learning approach, the loss is smaller with RUBi and decreases from 0.22 to 0.06. Secondly, we increase the importance of examples that cannot be answered without using both modalities. For these examples, the question-only branch outputs a mask that increases the score of the wrong answer. As a result, the loss is much higher and the VQA model is encouraged to learn from these examples. We illustrate this behavior in the second row of Figure 3.3 for the same question about the color of the banana. When the image contains a green banana, RUBi increases the loss from 0.69 to 1.20.

Joint learning procedure We jointly optimize the parameters of the base VQA model and its question-only branch using the gradients computed from two losses. The main loss $\mathcal{L}_{QM}$ refers to the cross-entropy loss associated with the predictions of $f_{QM}(v_i, q_i)$ from Equation 3.4. We backpropagate this loss to optimize all the parameters θ_{QM} which contributed to this loss. θ_{QM} is the union of the parameters of the base VQA model, the encoders, and the neural network nn_q of the question-only branch. In our setup, we share the parameters of the question encoder e_q between the VQA model and the question-only branch. The question-only loss $\mathcal{L}_{QO}$ is a cross-entropy loss associated with the predictions of $f_Q(q_i)$ from Equation 3.3. We use this loss to only optimize θ_{QO}, the union of the parameters of c_q and nn_q. By doing so, we further improve the question-only branch's ability to capture biases. Note that we do not backpropagate this loss to the question encoder e_q preventing it from directly learning question biases.

We obtain our final loss $\mathcal{L}_{\text{RUBi}}$ by summing the two losses together in the following equation:

$$\mathcal{L}_{\text{RUBi}}(\theta_{QM}, \theta_{QO}; \mathcal{D}) = \mathcal{L}_{QM}(\theta_{QM}; \mathcal{D}) + \mathcal{L}_{QO}(\theta_{QO}; \mathcal{D}) \tag{3.5}$$

3.3.2 Baseline architecture

Most VQA architectures from the state of the art are compatible with our RUBi learning strategy. To test our strategy, we design a fast and simple architecture inspired by the MuRel architecture (Cadene et al. 2019b). Our baseline architecture encodes the image as a bag of n_v visual features $\mathbf{v}_i \in \mathbb{R}^{d_v}$ using the pre-trained Faster R-CNN from Anderson et al. 2018b, and encodes the question as a vector $\mathbf{q} \in \mathbb{R}^{d_q}$ using a GRU, pre-trained on the Skip-thought task (Kiros et al. 2015b). It computes a bilinear fusion between the question vector and the visual features for each region. The bilinear fusion module is a BLOCK (Ben-Younes et al. 2019a) composed of 15 chunks, each of rank 15. The dimension of the projection space is 1000, and the output dimension is 2048. The output of the bilinear fusion is aggregated using a max pooling over n_v regions. The resulting vector is then fed into a Multi-Layer Perceptron (MLP) classifier composed of three layers of size (2048, 2048, 3000), with Rectified Linear Unit (ReLU) activations. It outputs the predictions over the space of the 3000 answers. While most of our experiments are done with this fast and simple baseline architecture, we experimentally demonstrate that the RUBi learning strategy is effective on two other VQA architectures, Bottom-Up and Top-Down Attention (UpDn) (Anderson et al. 2018a) and Stacked Attention Network (SAN) (Yang et al. 2016).

3.4 Results

3.4.1 Experimental setup

We train and evaluate our models on VQA-CP v2 (A. Agrawal et al. 2018b), described in Section 2.4.2. This dataset was developed to evaluate the models' robustness to question biases. We follow the same training and evaluation protocol as Ramakrishnan et al. 2018, who also propose a learning strategy to reduce biases. For each model, we report the standard VQA evaluation metric (Antol et al. 2015a). We also evaluate our models on the standard VQA v2 (Goyal et al. 2017a), as well as VQA-CP v1 and VQA-HAT (Das et al. 2016).

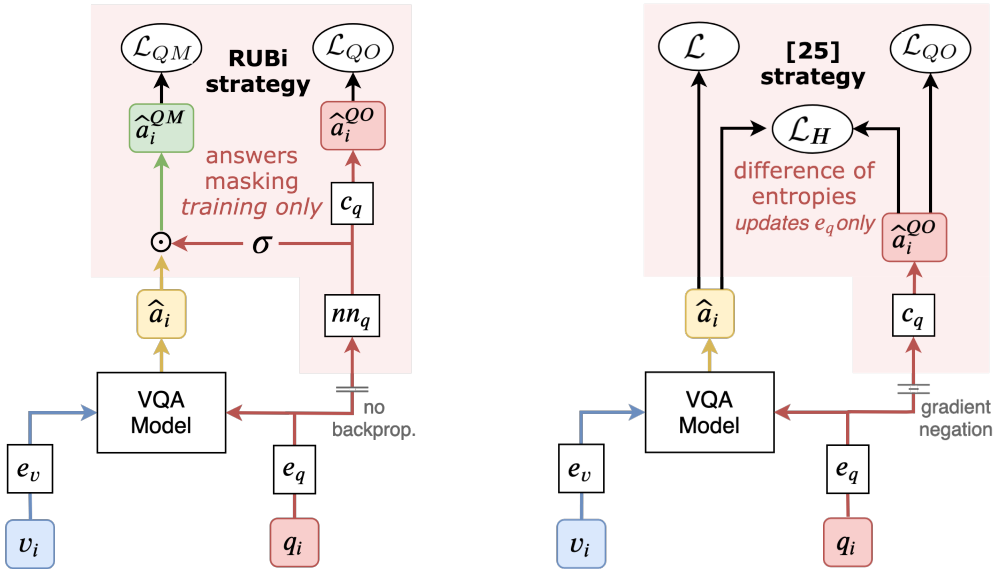


Figure 3.4. – Visual comparison between RUBi and Q-Adv+DoE (Ramakrishnan et al. 2018).

Optimization process We train all our models with the Adam optimizer. We train our baseline architecture with the learning rate scheduler of Cadene et al. 2019b. We use a learning rate of 1.5×10^{-4} and a batch size of 256. During the first 7 epochs, we linearly increase the learning rate to 6×10^{-4}. After epoch 14, we apply a learning rate decay strategy which multiplies the learning rate by 0.25 every two epochs. We train our models until convergence as we do not have a validation set for VQA-CP v2.

We fine-tune the question encoder during training, but we do not fine-tune the image extractor.

For the UpDn and SAN architectures, we follow the optimization procedure described in Ramakrishnan et al. 2018.

Software and hardware We use pytorch 1.1.0 to implement our algorithms in order to benefit from the GPU acceleration. We use a single NVidia Titan Xp GPU GPU for each experiment. A single experiment from Table 1 with the baseline architecture trained with or without RUBi takes less than five hours to run.

3.4.2 Evaluation of RUBi on VQA-CP

In Table 3.1, we evaluate our approach consisting of our baseline architecture trained with RUBi on VQA-CP v2 against previous methods and VQA models. We compute the average accuracy over 5 experiments with different random

seeds. Our RUBi approach reaches an average overall accuracy of 47.11% with a low standard deviation of ±0.51. This accuracy corresponds to a gain of +5.94 percentage points over the previous reference approach UpDn + Q-Adv + DoE. It also corresponds to a gain of +15.88 over GVQA (A. Agrawal et al. 2018b), which is a specific architecture designed for VQA-CP. RUBi reaches a +8.65 improvement over our baseline model trained with the classical cross-entropy. In comparison, the second-best approach UpDn + Q-Adv + DoE only achieves a +1.43 gain in overall accuracy over their baseline UpDn. In addition, our approach does not significantly reduce the accuracy over our baseline for the answer type *Other*, while the second-best approach reduces it by 10.57 points.

Model	Overall	Answer type		
		Yes/No	Number	Other
Question-Only	15.95	35.09	11.63	7.11
UpDn **	38.01	.	.	.
MuRel	39.54	42.85	13.17	45.04
GVQA	31.30	57.99	13.68	22.14
UpDn *	39.74	42.27	11.93	**46.05**
UpDn + Q-Adv + DoE	41.17	65.49	15.48	35.48
Balanced Sampling	40.38	57.99	10.07	39.23
Q-type Balanced Sampling	42.11	61.55	11.26	40.39
Baseline architecture	38.46 ± 0.07	42.85 ± 0.18	12.81 ± 0.20	43.20 ± 0.15
RUBi	**47.11** ± 0.51	**68.65** ± 1.16	**20.28** ± 0.90	43.18 ± 0.43

Table 3.1. – Results on VQA-CP v2 `test`. All reported models use the same features from (Anderson et al. 2018b). Models with * have been trained by (Ramakrishnan et al. 2018). Models with ** have been trained by (Shrestha et al. 2019). Models are Question-Only (A. Agrawal et al. 2018b), UpDn (Anderson et al. 2018b), BAN (J.-H. Kim et al. 2018), MuRel (Cadene et al. 2019b), RAMEN (Shrestha et al. 2019), BAN (J.-H. Kim et al. 2018), GVQA (A. Agrawal et al. 2018b), UpDn + Q-Adv + DoE (Ramakrishnan et al. 2018)

Additional baselines We compare our results to two sampling-based training methods. In the *Balanced Sampling* method, we sample the questions such that the answer distribution is uniform. In the *Question-Type Balanced Sampling* method, we sample the questions such that for every question type, the answer distribution is uniform, but the question type distribution remains the same overall Both methods are tested with our baseline architecture. We can see that the *Question-Type Balanced Sampling* improves the result from 38.46 in accuracy to 42.11. This

gain is already +0.94 higher than the previous reference method (Ramakrishnan et al. 2018) but remains significantly lower than our proposed method.

Architecture agnostic RUBi can be used on existing VQA models without changing the underlying architecture. In Table 3.2, we experimentally demonstrate the generality and effectiveness of our learning scheme by showing results on two additional architectures, Stacked Attention Network (SAN) (Yang et al. 2016) and Bottom-Up and Top-Down Attention (UpDn) (Anderson et al. 2018b). First, we show that applying RUBi on these architectures leads to important gains over the baselines trained with their original learning strategy. We report a gain of +11.73 accuracy points for SAN and +4.5 for UpDn. This lower gap in accuracy may show that UpDn is less driven by biases than SAN. This is consistent with results from (Ramakrishnan et al. 2018). Secondly, we show that these architectures trained with RUBi obtain better accuracy than with the reference strategy from (Ramakrishnan et al. 2018). We report a gain of +3.4 with SAN + RUBi over SAN + Q-Adv + DoE, and +3.06 with UpDn + RUBi over UpDn + Q-Adv + DoE.

Model	Overall	Yes/No	Number	Other
SAN (Yang et al. 2016)	24.96	38.35	11.14	21.74
SAN + Q-Adv+DoE	33.29	56.65	15.22	26.02
SAN + RUBi	**37.63**	**59.49**	**13.71**	**32.74**
UpDn (Anderson et al. 2018b)	39.74	42.27	11.93	**46.05**
UpDn + Q-Adv+DoE	41.17	65.49	15.48	35.48
UpDn + RUBi	**44.23**	**67.05**	**17.48**	39.61

Table 3.2. – Overall accuracy top1 on VQA-CP v2 for the SAN and UpDn architectures.

Results on VQA-CP v1 In Table 3.4, we report results on the VQA-CP v1 dataset (A. Agrawal et al. 2018b). Our RUBi approach consistently leads to significant gains over the classical learning strategy with a gain of +9.8 overall accuracy point with our baseline architecture, +19.2 with SAN and +7.66 with UpDn. Additionally, RUBi leads to a gain of +2.65 over the adversarial regularization method (Q-Adv + DoE) from (Ramakrishnan et al. 2018) with SAN. A visual comparison between RUBi and (Ramakrishnan et al. 2018) can be found in Figure 3.4. Finally, all three architectures trained with RUBi reach a higher accuracy than GVQA (A. Agrawal et al. 2018b) which has been hand-designed to overcome biases.

Impact on VQA v2 We report the impact of our method on the standard VQA v2 dataset in Table 3.3. VQA v2 train, val and test sets follow the same distribution,

Model	val	test-dev
Baseline (ours)	**63.10**	**64.75**
RUBi (ours)	61.16	63.18

Table 3.3. – Overall accuracy of the RUBi learning strategy on VQA v2 val and test-dev splits.

.

Model	Overall	Yes/No	Number	Other
GVQA (A. Agrawal et al. 2018b)	39.23	64.72	11.87	24.86
Baseline (ours)	37.13	41.96	12.54	41.35
Baseline + RUBi	**46.93**	**66.78**	**20.98**	**43.64**
SAN	26.88	35.34	11.34	24.70
SAN + Q-Adv+DoE	43.43	74.16	12.44	25.32
SAN + RUBi	**46.08**	**75.00**	**13.30**	**30.49**
UpDn (ours)	37.15	41.13	12.73	**43.00**
UpDn + RUBi	**44.81**	**69.65**	**14.91**	32.13

Table 3.4. – Overall accuracy top1 on VQA-CP v1. SAN+Q-Adv+DoE (Ramakrishnan et al. 2018)

contrarily to VQA-CP v2 train and test sets. In this context, we usually observe a drop in accuracy using approaches focused on reducing biases. This is because exploiting unwanted correlations from the VQA v2 train set is not discouraged and often leads to a higher accuracy on the test set. Nevertheless, our RUBi approach leads to a comparable drop to what can be seen in other comparable strategies. We report a drop of 1.94 percentage points with respect to our baseline, while (A. Agrawal et al. 2018b) report a drop of 3.78 between GVQA and their SAN baseline. (Ramakrishnan et al. 2018) report drops of 0.05, 0.73 and 2.95 for their three learning strategies with the UpDn architecture which uses the same visual features as RUBi. As shown in this section, RUBi improves the accuracy on VQA-CP v2 by a large margin, while maintaining competitive performance on the standard VQA v2 dataset compared to similar approaches.

3.4.3 Ablation study

Validation of the masking strategy We compare different fusion techniques to combine the output of nn_q with the output from the VQA model. We report a drop of 7.09 accuracy points on VQA-CP v2 by replacing the sigmoid with

a ReLU on our best-scoring model. Using an element-wise sum instead of an element-wise product leads to a further performance drop. These results confirm the effectiveness of our proposed masking method which relies on a sigmoid and an element-wise sum.

Validation of the question-only loss In Table 3.5, we validate the ability of the question-only loss $\mathcal{L}_{QO}$ to reduce the question biases. The absence of $\mathcal{L}_{QO}$ implies that the question-only classifier c_q is never used, and nn_q only receives gradients from the main loss $\mathcal{L}_{QM}$. Using $\mathcal{L}_{QO}$ leads to consistent gains on all three architectures. We report a gain of +0.89 for our Baseline architecture, +0.22 for SAN, +4.76 for UpDn.

Model	$\mathcal{L}_{QO}$	Overall	Yes/No	Number	Other
Baseline + RUBi	✓	**47.11**	68.65	20.28	**43.18**
	✗	46.11	**69.18**	**26.85**	39.31
SAN + RUBi	✓	**37.63**	59.49	**13.71**	**32.74**
	✗	36.96	**59.78**	12.55	31.69
UpDn + RUBi	✓	**44.23**	**67.05**	**17.48**	**39.61**
	✗	39.47	60.27	16.01	35.01

Table 3.5. – Ablation study of the question-only loss $\mathcal{L}_{QO}$ on VQA-CP v2.

3.4.4 Analysis of grounding on VQA-HAT

We conduct additional studies to evaluate the grounding ability of models trained with RUBi. We follow the experimental protocol of VQA-HAT (Das et al. 2016), described in Section 2.4.2. This dataset contains human attention maps for images from the VQA v1 dataset, indicating which regions humans found relevant for answering the question. We train our models on VQA v1 train set and evaluate them using rank-correlation on the VQA-HAT val set, which is a subset of the VQA v1 val set. This metric compares attention maps computed from a model against human annotations. In Table 3.6, we report a gain of +0.012 with our baseline architecture trained with RUBi, a gain of +0.019 with SAN and a loss of -0.003 with UpDn architecture.

We display in Figure 3.5 and Figure 3.6 some manually selected VQA triplets associated to the human attention maps provided by VQA-HAT (Das et al. 2016) and the attention maps computed from our baseline architecture when trained with and without RUBi. In Figure 3.5, we observe that the attention maps with

Model	RUBi	Rank-Corr.
Random (Das et al. 2016)		0.000
Human (Das et al. 2016)		0.623
Baseline	✗	0.431
	✓	**0.443**
SAN	✗	0.191
	✓	**0.210**
UpDn	✗	**0.449**
	✓	0.446

Table 3.6. – Correlation with Human Attention Maps on VQA-HAT `val` set (Das et al. 2016).

RUBi are closer to the human attention maps than without RUBi. On the contrary, we observe in Figure 3.6 some failure to improve grounding ability.

3.4.5 Qualitative examples

To better understand the impact of our RUBi approach, we compare in Figure 3.7 the answer distribution on VQA-CP v2 for some specific question patterns. We also display interesting behaviors on some examples using attention maps extracted as in Cadene et al. 2019b. In the first row, we show the ability of RUBi to reduce biases for the *is this person skiing* question pattern. Most examples in the train set have the answer *yes*, while in the test set, they have the answer *no*. Nevertheless, RUBi outputs 80% of *no*, while the baseline almost always outputs *yes*. Interestingly, the best scoring region from the attention map of both models is localized on the shoes. To get the answer right, RUBi seems to reason about the absence of skis in this region. It seems that our baseline gets it wrong by not seeing that the skis are not locked under the ski boots. This unwanted behavior could be due to question biases. In the second row, similar behaviors occur for the *what color are the bananas* question pattern. 80% of the answers from the train set is *yellow*, while most of them are *green* in the test set. We show that the amount of *green* and *white* answers from RUBi are much closer to the ones from the test set than with our baseline. In the example, it seems that RUBi relies on the color of the banana, while our baseline misses it. In the third row, it seems that RUBi is able to ground the textual concepts such as *top part of the fire hydrant* and *color* on the right visual region, while the baseline relies on the correlations between the fire hydrant, the yellow color of its core and the answer *yellow*. Similarly, on the fourth row, RUBi grounds *color*, *star*, *fire hydrant* on the right region, while our

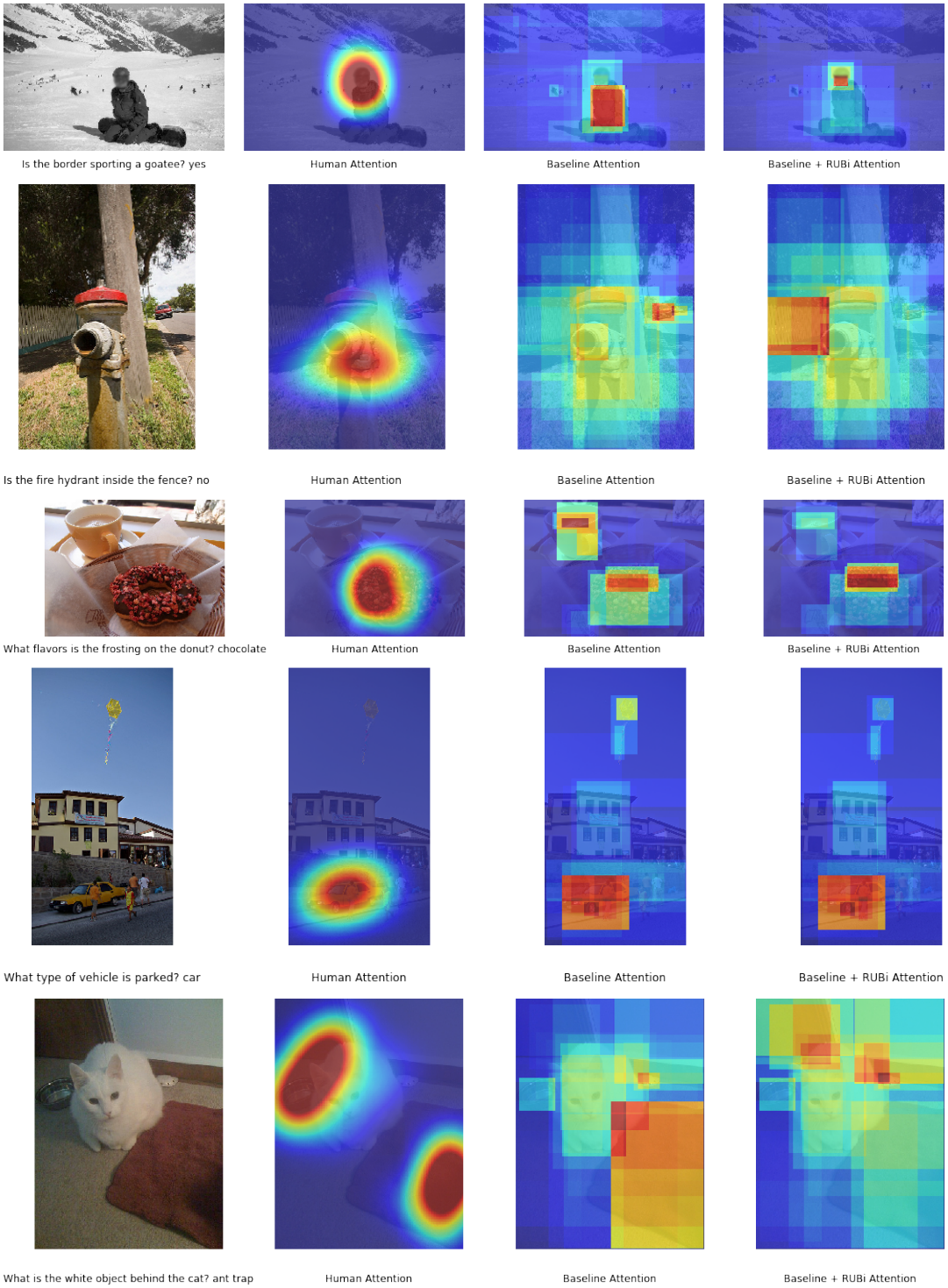


Figure 3.5. – Examples of better grounding ability on VQA-HAT implied by RUBi. From the left column to the right: image-question-answer triplet, human attention map from (Das et al. 2016), attention map from our baseline, attention map from our baseline trained with RUBi.

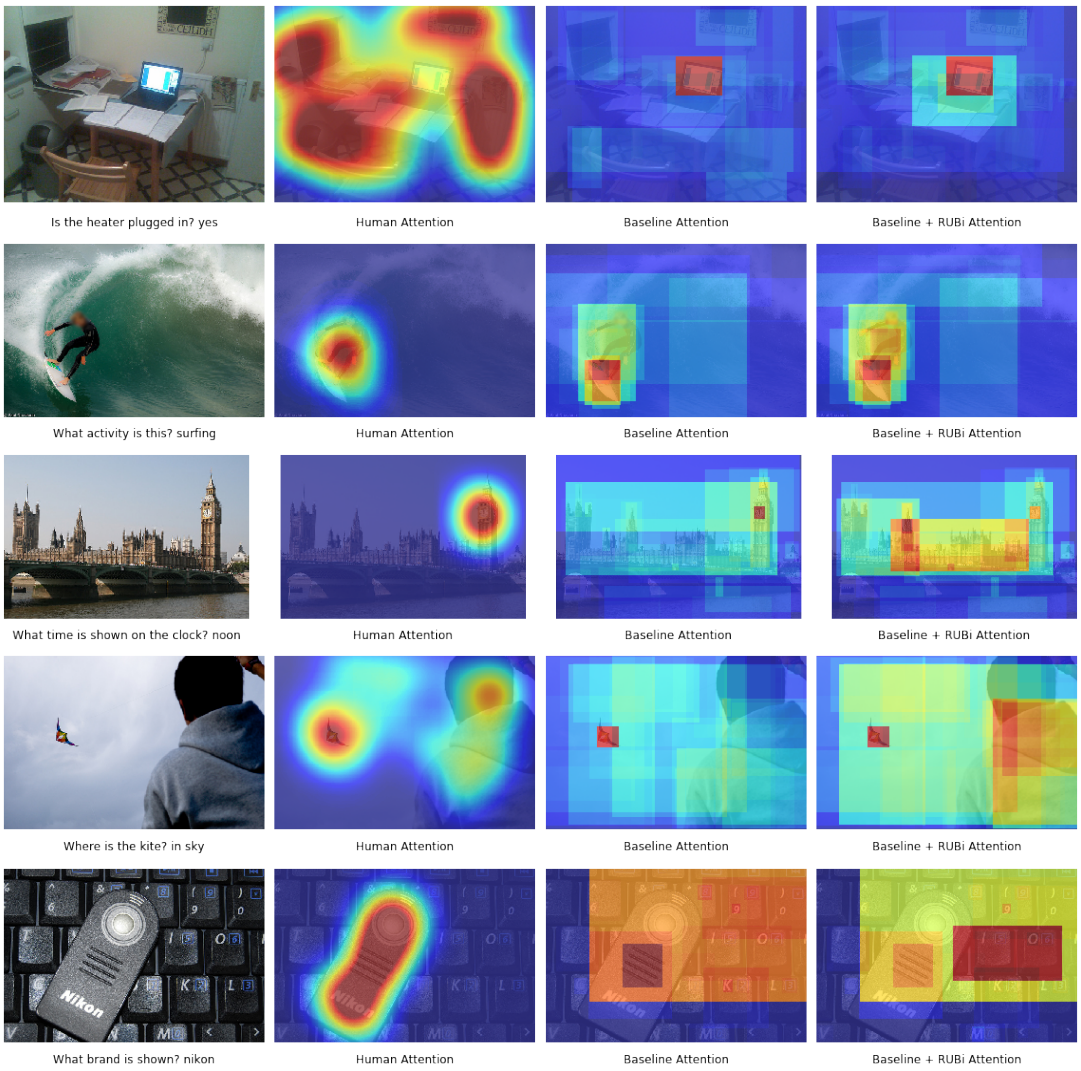


Figure 3.6. – Examples of failure to improve grounding ability on VQA-HAT. From the left column to the right: image-question-answer triplet, human attention map from (Das et al. 2016), attention map from our baseline, attention map from our baseline trained with RUBi.

baseline relies on correlations between *color*, *fire hydrant*, the yellow color of the top part region and the answer *yellow*. Interestingly, there is no similar question that involves the color of a star on a fire hydrant in the training set. It shows the capacity of RUBi to generalize to unseen examples by composing and grounding existing visual and textual concepts from other kinds of question patterns.

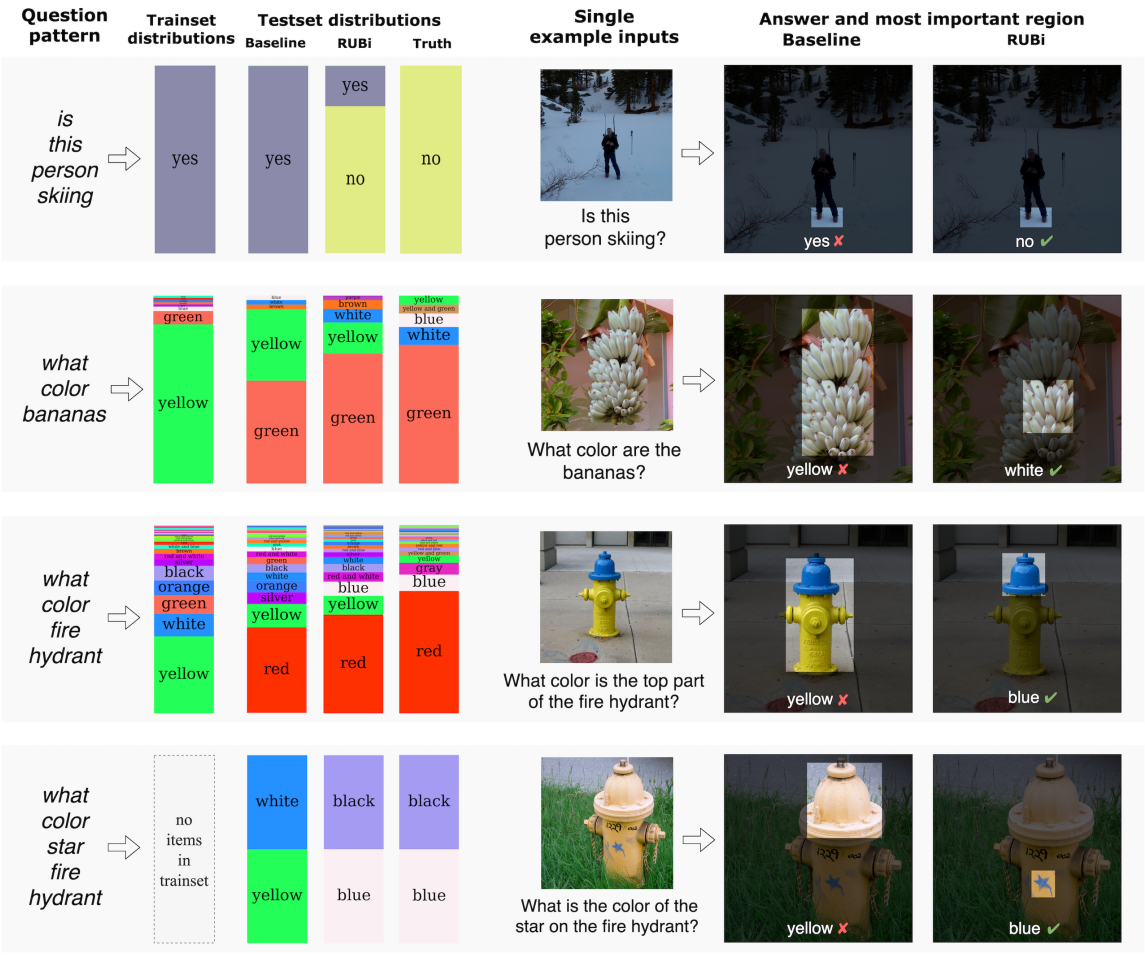


Figure 3.7. – Qualitative comparison between the outputs of RUBi and our baseline on VQA-CP v2 `test`. On the left, we display distributions of answers for the train set, the baseline evaluated on the test set, RUBi on the test set and the ground truth answers from the test set. For each row, we filter questions in a certain way. In the first row, we keep the questions that exactly match the string *is this person skiing*. In the three other rows, we filter questions that respectively include the following words: *what color bananas*, *what color fire hydrant* and *what color star hydrant*. On the right, we display examples that contain the pattern from the left. For each example, we display the answer of our baseline and RUBi, as well as the best scoring region from their attention map.

3.5 Conclusion

In this chapter, we explore unimodal shortcut learning in VQA: models tend to rely mostly on the question modality, sometimes ignoring the image. This is a significant issue for deploying models in the real world: they are not robust to unusual situations and might fail catastrophically. This also means that models do not learn the intended *reasoning mechanism*. To tackle those issues, we propose RUBi, a learning strategy to reduce shortcut learning in VQA. The main VQA model is learned jointly with a question-only branch that captures unwanted statistical regularities from the question modality. This branch influences the base VQA model to prevent the learning of unimodal biases from the question. RUBi is designed to be model agnostic.

We demonstrate the effectiveness of modifying the learning strategy as an inductive bias to reduce question-based shortcut learning in VQA: RUBi improves the performance of baseline models on VQA-CP v2, a dataset specifically designed to account for question biases. Additionally, we see that the RUBi strategy slightly improves the grounding for some models, demonstrating that it might help models learn the intended reasoning mechanism. This shows that learning strategies are an effective way to reduce shortcut learning for the VQA task.

Multiple following works explored other learning strategies. For instance, Kervadec et al. 2020 propose to take account of the semantic structure of the answer distribution: a model answering "pink" instead of "red" is better than answering "basketball". They show this makes the model more robust to biases in the VQA-CP task. Teney et al. 2020a propose to create minimal counterfactual examples, i.e. an example with a slightly different image, as an existing example, with a ground-truth answer that is different from the original example. This encourages the model to use the image modality and not rely solely on the question. In the next chapter, we explore another approach to tackle this same issue: designing the model with architectural priors to prevent shortcut learning.

CHAPTER 4

REDUCING SHORTCUT LEARNING WITH ARCHITECTURAL PRIORS FOR VISUAL COUNTING

Chapter abstract

In this chapter, we focus on the task of answering counting questions, a subset of the Visual Question Answering task. It is also subject to the same kind of biases as the original VQA task, but the output distribution is simpler: it is a single number. This allows us to explore another strategy to reduce shortcut learning: ***architectural priors****. We first introduce two large-scale out-of-distribution datasets: TallyQA-CP and TallyQA-Odd-Even. They are made of training and testing sets that do not follow the same answer distribution to penalize models that have learned biases instead of proper counting mechanisms. We show that state-of-the-art models obtain low performances on our datasets, which means that they have learned biases. Then, we show that using architectural priors in the models can help to reduce shortcut learning: we propose the Spatial Counting Network (SCN), a model which incorporates domain knowledge to encourage learning of the proper counting mechanisms: it outputs a natural number obtained by selecting and counting objects in the image. We show that our model performs better on our datasets that penalize biases. We also report a better ability to select the correct objects to count in the image when trained on a classic dataset.*

The work in this chapter has led to the publication of a workshop paper:

- Corentin Dancette, Remi Cadene, Xinlei Chen, and Matthieu Cord (2021a). "Learning Reasoning Mechanisms for Unbiased Question-based Counting". In: *VQA Workshop, Conference on Computer Vision and Pattern Recognition (CVPR)*.

Contents

4.1 Introduction

As we explain in Chapter 2, shortcut learning is problematic in the context of reasoning tasks like Visual Question Answering (VQA). In Chapter 3, we propose a learning strategy to tackle the issue of unimodal biases. Here, we focus on the task of answering *counting* questions, a subset of VQA, and propose to explore another direction to reduce shortcut learning: using **architectural priors** to constrain the network to learn the correct mechanism and prevent it to learn spurious correlations. We choose the counting task for several reasons:

- First, similarly to VQA, question-based counting requires high-level reasoning abilities and displays similar biases. As illustrated qualitatively in Figure 4.1 (and later quantitatively in our experiments), current models tend to find an easier way out by correlating the output to some spurious patterns in the input and skipping the learning of reasoning mechanisms. For instance, questions starting with *how many wings* can reasonably be answered **2** without looking at the image, allowing models that use this kind of bias to achieve high accuracy on the testing set. However, those models will be easily fooled in the real world. In the largest question-based counting dataset, TallyQA (M. Acharya et al. 2019), we found that the appearance of certain words in the question or objects in the image is highly predictive of the count label. For instance, the presence of the words "cars", "are", and "black" in the

question are associated 94% of the time with the answer **0**. Similarly, the words "legs", and "animal" are associated with the answer **4**. It is critical to create appropriate benchmarks that reflect these failures and to propose approaches to reduce shortcut learning.

- Second, the mechanism of counting is well-defined and has a structure that can be taken into account in models. To properly answer a counting question, one has to first detect each relevant object in the image, based on a complex, sometimes compositional question involving other objects. Then, figure out their relationships for possible de-duplication or filtering, and then accumulate and aggregate the number of objects to count. These mechanisms must be learned using the answer, a single number in the case of counting, as the only supervision. We take advantage of this property to incorporate architectural priors to reduce shortcut learning. Thus, we can evaluate if the proper mechanism has been learned for models that detect and select objects to count.

- Third, counting is a useful task that leads to important practical applications (Lempitsky and Zisserman 2010; Briggs 2009; Onoro-Rubio and López-Sastre 2016). Solving question-based counting would lead to the next generation of counting systems with textual interfaces.

In this chapter, we take a first step towards the development of unbiased models that learn to leverage the underlying mechanisms for multi-modal reasoning tasks. Our contributions are two-fold. We propose two counting datasets meant to evaluate a model and learning strategy's ability to avoid learning biases. Both datasets are built on the idea of *changing distributions*, meaning the training and testing distributions are different. Intuitively, if a model has learned the reasoning mechanisms for counting, it should generalize well despite changing distributions, whereas the models that learn to merely correlate inputs to outputs likely cannot. This method has been used for general VQA. In fact, VQA-CP, presented in Section 2.4.2, was precisely developed by re-organizing the training and testing sets of original VQA v1 and v2. Our first dataset, TallyQA-CP, follows a similar protocol. However, VQA-CP only tackles *question biases* as the re-arrangement was conditioned on different question types. We go a step further by introducing another set, TallyQA-Odd-Even, which by design penalizes models that rely on *any kind of shortcut*, not just question biases. We experimentally verify the feasibility of using our datasets to penalize the use of biases and show that reference VQA models suffer from large performance drops on our datasets, which indicates that they have learned biases.

Additionally, we contribute by introducing a simple and effective model — Spatial Counting Network (SCN) — that avoids learning the biases and instead

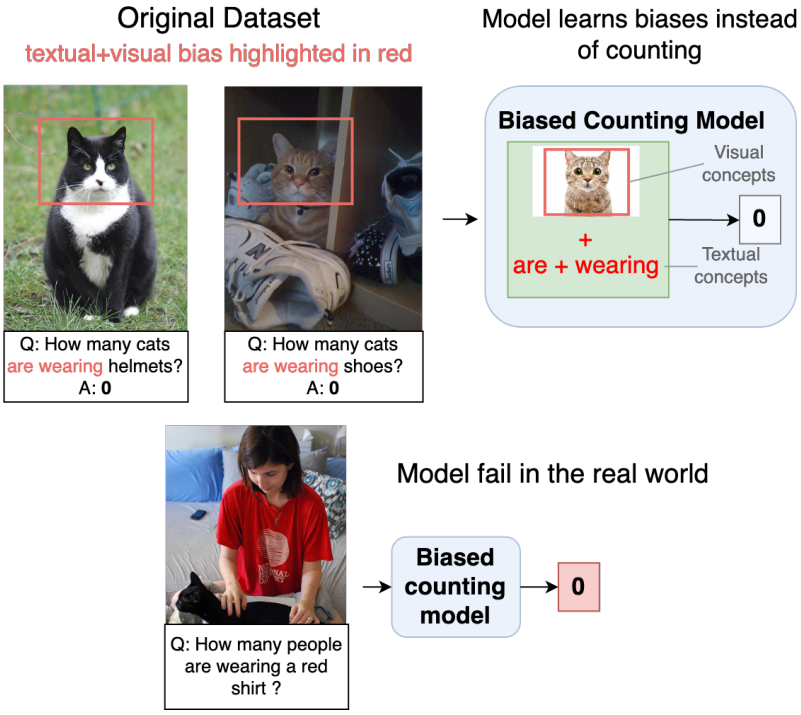


Figure 4.1. – Matching simple patterns from the training set can be enough to answer a large number of counting questions and obtain high accuracy on the testing set. For instance, when the words "are" and "wearing" appear in the question while a head of a cat appears in the image (183 times in the training set), the answer is always "0" in the training and testing sets. In the real world, biased models that rely on such a pattern would fail to provide the correct answer.

learns the proper reasoning mechanisms for counting. It is based on the following design choices: 1) a regression loss instead of a classification loss to account for the answer structures (ordered natural numbers) and strive for a better out-of-distribution generation; 2) a final count based on individual scores to each region with self-attention-based relationship modeling; and 3) entropy regularization to enforce sparse region scores.

In Section 4.2, we give an overview of works related to visual counting. In Section 4.3, we propose two evaluation benchmarks for shortcut learning in visual counting. In Section 4.4, we propose Spatial Counting Network (SCN), our model designed for visual counting, and we evaluate it on our proposed benchmark in Section 4.5.

4.2 Related work

We discussed the general VQA architectures in Section 2.2.2, and shortcut learning in VQA in Section 2.4. In this section, we discuss the related work on visual counting.

Counting has long been a subject of interest in the computer vision community (Lempitsky and Zisserman 2010; Arteta et al. 2016; Onoro-Rubio and López-Sastre 2016; Marsden et al. 2018; Babu Sam et al. 2017; Sindagi and Patel 2018; Chattopadhyay et al. 2017) leveraging annotations such as segmentation maps (Cholakkal et al. 2019), bounding boxes (J. Liu et al. 2018) or localized dots (Lempitsky and Zisserman 2010; Y. Liu et al. 2019). In this section, we discuss more general question-biased counting approaches that require a minimal amount of supervision in the form of a unique answer per question-image pair.

Question-based counting datasets The approaches studying counting questions on real images were first developed on Visual Question Answering (VQA) datasets (Antol et al. 2015b; Goyal et al. 2017b; Krishna et al. 2017b; Kafle and Kanan 2017). They were evaluated on the "how many?" questions (Y. Zhang et al. 2018) or on subsets associated with numerical ground-truth answers such as Count-QA (Chattopadhyay et al. 2017) or HowMany-QA (Trott et al. 2018). Since examples labeled as *number* only account for around 10% of the VQA datasets, a large dataset dedicated to counting questions was introduced: TallyQA (M. Acharya et al. 2019). It is composed of novel simple and complex questions with the addition of examples from previous datasets: VQA v2 (Goyal et al. 2017b), HowMany-QA (Trott et al. 2018), TDIUC (Kafle and Kanan 2017) and Visual Genome (Krishna et al. 2017b). We use TallyQA to build our novel datasets by reorganizing examples between training and testing sets to induce shifts in the distribution of counting labels.

Question-based counting models We consider only models that take a question as input, and not specialized counting models such as (Chattopadhyay et al. 2017; Sindagi and Patel 2018). General VQA models are able to answer counting questions by incorporating various modules that learn a fusion between the image and the question (Malinowski and Fritz 2014b; Antol et al. 2015b; Ben-Younes et al. 2017a; Ben-Younes et al. 2019b; Anderson et al. 2018a). Those equipped with relational and self-attention modules reach better results on counting questions (Santoro et al. 2017; Perez et al. 2018; Drew Arad Hudson and D. Manning 2018; Cadene et al. 2019a). We hypothesize that these modules help to avoid counting duplicated object regions more than once, an important challenge in counting. Then, dedicated counting models were developed. The state-of-the-art, RCN (M.

Acharya et al. 2019) is based on relational networks (Santoro et al. 2017), while Counter (Y. Zhang et al. 2018) builds a graph representation and performs hand-crafted operations to select objects and remove duplicates. Instead, our model is composed of the recent self-attention module (Vaswani et al. 2017). Additionally, an important characteristic of counting is that the answers are natural numbers. RCN and Counter take advantage of that by using a classification loss. ILRC (Trott et al. 2018) uses a different approach: it learns a hard selection of image regions using reinforcement learning. Instead, our model learns a soft selection in an end-to-end fashion and is trained with a regression loss with an entropy regularization term to enforce the output of natural numbers. Our design choices guarantee a certain level of interpretability and allow the model to output counting values that have never been encountered in the training set.

4.3 Novel out-of-distribution datasets

4.3.1 Methodology for creating our evaluation benchmarks

In this section, we describe a methodology to create evaluation benchmarks for visual counting. We design them to penalize models that over-rely on any kind of data biases without the need for external annotations or human supervision. We use them to select models that have learned a more robust counting mechanism instead of biases. We introduce two datasets by changing the distribution of count labels of the training and testing sets of TallyQA (M. Acharya et al. 2019), the recent and biggest question-based visual counting dataset. Its original training set contains 130K real images from COCO (T.-Y. Lin et al. 2014b) and Visual Genome (Krishna et al. 2017b). Each image is associated with questions and count labels for a total of ∼250K samples. Answering counting questions requires abilities to detect relationships between objects, and their attributes, perform spatial reasoning, and more. In Section 4.3.2, we give additional statistics about our datasets. Then, in Section 4.3.3, we benchmark existing counting models and show that they are subject to biases.

TallyQA-CP Inspired by the VQA-CP dataset, detailed in Section 2.4.2, we build a new version of TallyQA (M. Acharya et al. 2019) to penalize models that over-rely on the question-related biases. In this dataset, we condition the final count label distribution on the question modality. We construct a new training set and testing set by first extracting the main concept to be counted from each question (e.g. in "how many tables are green", the concept will be "tables"). The concept serves a similar purpose as the question type in VQA-CP (A. Agrawal

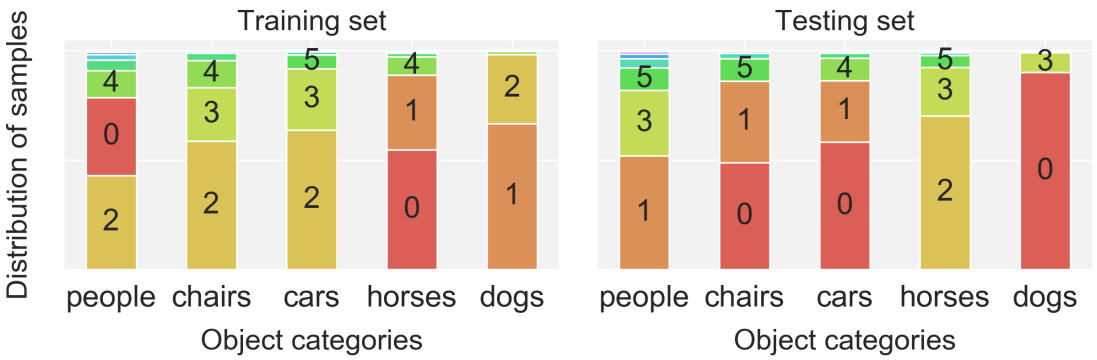


Figure 4.2. – Shift in the distribution of samples between the training and testing sets for the 5 most common objects in our **TallyQA-CP** dataset. Models that over-rely on question biases are penalized when evaluated on the testing set.

et al. 2018a): it conditions the answer distribution differently between the training and the testing set. More formally, if we note the answer set A and the concept of questions C, then our goal is that $\forall c \in C, P_{train}(A|c) \neq P_{test}(A|c)$. A model relying too much on this main concept to answer the question (for example answering **2** each time the concept is *wings*) would be penalized on the testing set.

Here we describe how we find the concept in a given question. The main heuristic consists in using the position of the word in the question. In most cases, the concept to be counted is the third word of the question, as most questions start with "How many <main concept>...". The second heuristic consists in selecting the fourth word when the third is a color. For instance, the concept will be "cars" in "How many blue cars are in the image?". The third heuristic consists in selecting the fifth word when the third and fourth words are "of the" or "of those". For instance, the concept will be "cars" in "How many of the cars are green ?" We manually verified that these heuristics ensure picking the correct concept in most cases.

For each concept, we calculate its associated answer distribution and apply a greedy strategy to split all questions into a new training and testing set: For a concept c, we assign its related question-image-answer samples containing the most common answer randomly to either the training or the testing set. We then assign the samples containing the second most common answer to the other set. We continue alternating training and testing set until all samples have been assigned. We display the distributions for the five most common concepts in Figure 4.2. In Section 4.3.3, we experimentally verify that TallyQA-CP penalizes question biases by evaluating a question-only model: as expected, it is almost unable to provide the correct answer.

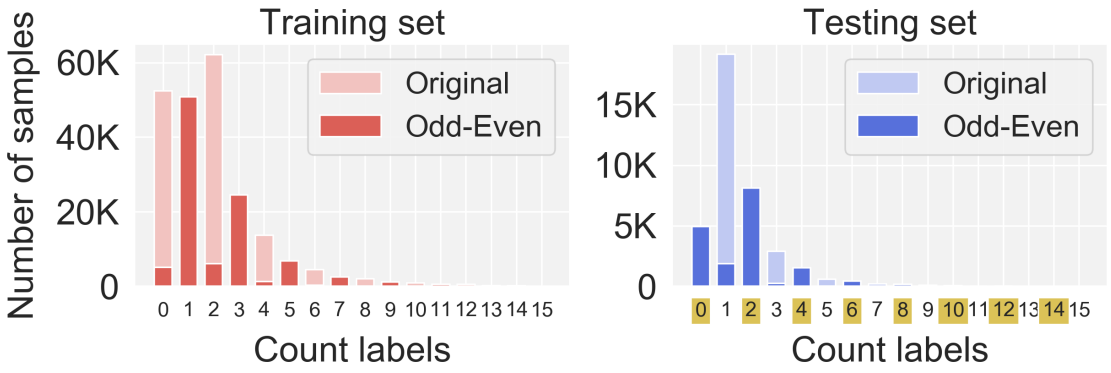


Figure 4.3. – Shift in number of samples between the training and testing sets of the original TallyQA dataset and our **TallyQA-Odd-Even** dataset. Models that over-rely on any kind of data biases are penalized when evaluated on the even count labels (in yellow).

TallyQA-Odd-Even A characteristic of our proposed TallyQA-CP is that it mostly penalizes the use of question-related biases. Instead, we introduce the Odd-Even version that penalizes, **by construction**, the use of any kind of superficial shortcuts. To do so, we modify the count label distribution without any conditioning on the input (question or image) in order to also target image-related shortcuts and multimodal shortcuts. We generate the unbalanced TallyQA-Odd-Even dataset by removing 90% of the samples associated with an even count label from the TallyQA training set and 90% of the samples associated with an odd label from the testing set. We display in Figure 4.3 the resulting number of samples per count label. The 90% proportion was chosen because it introduces a large shift in the distribution of count labels while allowing classification models to learn from every possible count label. Choosing 100% would result in a zero-shot dataset on which existing classification models could not be tested. We show in Section 4.3.2 that the 90% proportion generates a tiny shift in the distribution of images and questions which ensures that we only evaluate the impact of a shift in count labels. In Section 4.3.3, we experimentally verify that TallyQA-Odd-Even equally penalizes question and image biases by evaluating question-only and image-only models. They reach similar low scores. In Section 4.5, we report additional results on various shifts between 70% and 100%.

Validation sets As raised by Teney et al. 2020b, most works that evaluate models on out-of-distribution datasets such as VQA-CP (A. Agrawal et al. 2018a) do not use a validation set to early stop training or select hyperparameters. This bad practice encourages adaptive over-fitting (Dwork et al. 2015) on the testing set distribution. We address this common issue by holding out 10% of the training sets as validation sets so that they follow the same distribution. It is expected that

biased models perform well on validation sets, but have lower scores on testing sets.

4.3.2 Statistics about our datasets

Training, validation and testing sets statistics We additionally define TallyQA-Even-Odd, which is similar to the TallyQA-Odd-Even dataset, but with the reverse protocol for creating the two splits. For all our datasets, the validation set is made of 10% of the training set data, therefore follows the same distribution. More specifically, for TallyQA-Odd-Even and TallyQA-Even-Odd datasets, their validation set is built by holding 10% of the images out of the training set before applying the same ablation strategy on both sets (e.g. removing odd examples). For TallyQA-CP, the validation set is built after the resampling of examples into the training and testing set.

In Table 4.1, we display the number of odd and even triplets in each set of TallyQA-Odd-Even where 90% of triplets have been removed ($p = 90\%$), and other datasets where $p = \{0, 50, 100\}$. In Table A.1 of the Appendix A, we report the same numbers with the TallyQA-Even-Odd dataset. In Table 4.2, we display the number of triplets in each set for the TallyQA-CP dataset.

	Training set		Validation set		Testing set	
p%	Odd	Even	Odd	Even	Odd	Even
0 %	87,289	137,102	9,635	15,292	23,138	15,451
50 %	87,289	68,549	9,635	7,644	11,565	15,451
90 %	87,289	13,707	9,635	1,525	2,328	15,451
100%	87,289	0	9,635	0	0	15,451

Table 4.1. – Number of *image-question-count* triplets for each set generated by our Odd-Even-p% strategy when applied on the TallyQA dataset (Odd-Even-0% leads to the the original TallyQA distribution, Odd-Even-90% leads our TallyQA-Odd-Even dataset, mainly used in this study). Numbers of triplets for intermediate values of p can be obtained with linear interpolation.

Shift in the distribution of questions and visual concepts We compute the distributions of words from the questions and visual concepts in the images in various TallyQA-Odd-Even-p% training sets and compare them to the original distributions of TallyQA. To compute the word distribution, we proceed as follows. We first remove the common words *how, many, can, you, scene, picture, pictured,*

Training Set	Validation Set	Testing Set
137,080	15,231	135,596

Table 4.2. – Number of triplets for our TallyQA-CP dataset.

image, photo, there, are, seen, see, visible, shown, this, in, the, on, be, of, a, to to only keep those that are associated to specific concepts in the images. We then compare the distributions using the Bhattacharyya coefficient (Bhattacharyya 1946) – a similarity metric that reaches 0 when there is no overlap between distributions, and 1 when both are the same. Similarly, we compute visual concept distributions by using the categories assigned to every bounding box extracted from our pre-trained object detector (Anderson et al. 2018a) and compare the distributions using the Bhattacharyya coefficient. In Tables 4.3 and 4.4, we see that all similarity measurements are close to 1 which confirms that our protocol leads to a small shift in the distribution of words and visual concepts from TallyQA original distribution.

p%	Words Similarity	Visual similarity
0 %	1.0	1.0
50 %	0.997	0.9999
90 %	0.986	0.9996
100 %	0.976	0.9995

Table 4.3. – Bhattacharyya coefficients (Bhattacharyya 1946). Words and visual concepts similarity between each of our generated training sets using our Odd-Even-p% strategy and the original TallyQA training set.

Words Similarity	Visual similarity
0.962	1.00

Table 4.4. – Bhattacharyya coefficients (Bhattacharyya 1946). Words and visual concepts similarity between the training and the testing sets of our TallyQA-CP dataset. The shift in distribution is very small.

4.3.3 Evaluating counting models against our benchmarks

Experimental setup We compare state-of-the-art counting models and strong baselines on our proposed datasets. We do not evaluate the best Transformer-based VQA models (J. Lu et al. 2019; Tan and Bansal 2019) since they have been pre-trained on images and questions included in our testing sets. Also, we do not compare against counting models that are not designed to take the question as input (Chattopadhyay et al. 2017; Sindagi and Patel 2018). The current state-of-the-art on TallyQA is RCN (M. Acharya et al. 2019), a classification model based on relation networks (Santoro et al. 2017). Our **Random** $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ baselines are random classifiers that follow respectively the training and the testing set answer distributions. We also test bias-reduction baselines: a uniform sampling of answers during training **(RCN with Sampling)**, and our Reducing Unimodal Biases (RUBi) strategy described in Chapter 3, that reduces question-related biases. Models that over-rely on biases are expected to perform well on the validation sets since they follow the training set distribution but suffer from a large loss in accuracy on the testing sets.

	TallyQA-CP				TallyQA-Odd-Even				TallyQA	
	Testing set		Validation set		Testing set		Validation set		Testing Set	
	Acc.	RMSE	Acc.	RMSE	Acc.	RMSE	Acc.	RMSE	Acc.	RMSE
Random $\mathcal{D}_{\text{train}}$	19.53	2.84	22.13	2.77	10.26	2.81	32.35	2.89	–	–
Random $\mathcal{D}_{\text{test}}$	20.40	2.89	19.78	2.81	**30.68**	2.61	10.21	2.75	–	–
Q-Only	0.63	2.23	66.12	1.86	16.92	1.91	54.46	2.07	42.38	1.74
I-Only	21.55	2.24	41.99	2.08	9.80	2.06	54.20	2.06	38.14	1.70
Q+I	1.68	1.97	73.23	1.49	20.86	1.80	62.35	1.69	52.32	1.49
MUTAN	1.91	1.96	74.08	1.42	24.99	1.67	67.12	1.51	53.51	1.54
Counter	0.64	2.08	71.34	1.66	19.89	1.83	59.98	1.86	62.58	1.34
RCN	2.00	1.76	77.66	1.30	28.40	**1.61**	70.06	1.34	**65.49**	**1.26**
RCN w/ Sampling	5.58	1.82	76.34	1.37	27.10	1.63	65.44	1.44	53.78	1.58
RCN + RUBi	**31.04**	**1.56**	68.11	1.22	25.35	1.71	68.28	1.48	59.83	1.35

Table 4.5. – Benchmark of question-based visual counting models on our TallyQA-CP and TallyQA-Odd-Even datasets. We report the accuracy and the RMSE scores on the testing and validation sets. RCN + Sampling stands for RCN with a uniform sampling strategy. We also report scores on the original TallyQA (M. Acharya et al. 2019). Models are: Q-Only, I-Only, Q+I (M. Acharya et al. 2019), MUTAN (Ben-Younes et al. 2017a), Counter (Y. Zhang et al. 2018), RCN (M. Acharya et al. 2019), RUBi (Chapter 3).

Counting models are biased As shown in Table 4.5, all models suffer from a large drop in accuracy, compared to their scores on the validation set, and on the original version of TallyQA (M. Acharya et al. 2019).

First, TallyQA-CP penalizes strongly the question-only model: it reaches an accuracy close to zero on the testing set. This confirms that this dataset penalizes models that rely on question shortcuts. The image-only model, on the contrary, is less penalized and even beats most of the previous state-of-the-art models such as RCN. On our TallyQA-Odd-Even, we can observe a different trend: the two unimodal baselines have closer scores, 16.92 and 9.80, with the image-only now having the lowest score. This confirms that this dataset penalizes all kinds of shortcuts.

The previous state-of-the-art model RCN has an overall accuracy of 65.49% on TallyQA and reaches an even high score on the validation sets of both our benchmarks. However, it only gets 2% accuracy on the TallyQA-CP testing set, and 28.4% on TallyQA-Odd-Even, suffering from a huge loss in accuracy from the validation sets. We observe similar trends for MUTAN (Ben-Younes et al. 2017a) and Counter (Y. Zhang et al. 2018), two commonly reported VQA models.

Additionally, the bias-reduction methods (uniform sampling and RUBi) have a positive impact on TallyQA-CP, which is expected, especially for RUBi, since it targets specifically question-related biases. On the contrary, both methods degrade performances on our TallyQA-Odd-Even testing set. Finally, we can notice **most of the models**, in both benchmarks, are worse than the **Random** $\mathcal{D}_{\text{test}}$ classifier that follows the testing set distribution. This highlights the fact that, while state-of-the-art counting models reach high accuracy on regular datasets, they are in fact incapable of counting in situations that do not match closely their training distribution and instead rely mostly on biases.

4.4 Spatial Counting Network

We now describe our model, Spatial Counting Network (SCN). It contains inductive biases to encourage the learning of the counting mechanism, and avoid learning biases. Our model uses multi-modal fusion and self-attention to assign counting scores to individual image regions, which allows the final accumulated count number to be spatially grounded. To help generalization to modified count distributions, we use a regression loss to train our model, as opposed to a classification loss (Y. Zhang et al. 2018; M. Acharya et al. 2019) and use entropy regularization to encourage the counting of natural numbers, as opposed to making discrete decisions trained with reinforcement learning (Trott et al. 2018). Impor-

tantly, we do not incorporate knowledge about the testing set distributions such as sampling or weighting triplets based on their count labels.

Overview. An overview of our model is shown in Figure 4.4. We mostly reuse the formalism defined in Section 2.2.2, with some slight changes. Formally, given a dataset $\mathcal{D}$ consisting of n triplet samples (v, q, c) with $v \in \mathcal{V}$ an image, $q \in \mathcal{Q}$ a natural language question and $c \in \mathbb{N}$ a count label corresponding to the number of instances in the image, the goal is to learn a mapping $f{:}\mathcal{V}\times\mathcal{Q}\rightarrow\mathbb{N}$ with learnable parameters θ. Our model builds such a mapping by first encoding both inputs and fusing them, which we detail next.

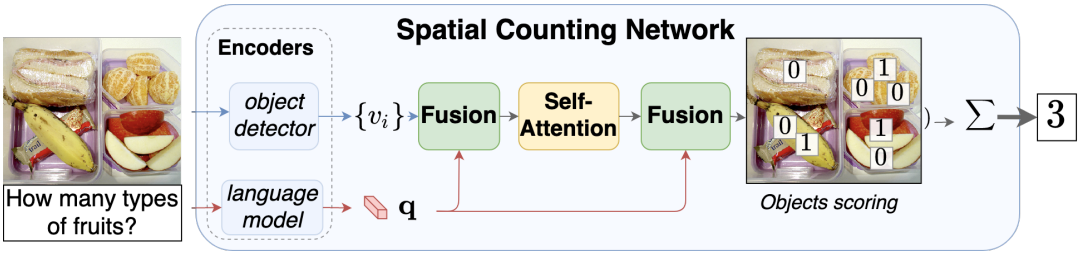


Figure 4.4. – **Spatial Counting Network**. It takes an image and a counting question as inputs and outputs a count label. Each of the detected objects is processed according to the question and their neighborhood until a counting score is obtained. The score indicates the presence (*e.g.* $\approx$1) or absence (*e.g.* $\approx$0) of a corresponding instance. The final count prediction is produced by summing up all scores.

Encoders and multi-modal fusion As shown in Figure 4.4, we use two encoders to produce vectorized representations for image v and question q. For image v, a pre-trained object detector (Anderson et al. 2018a) transforms the raw pixels to a set of n_v spatially located vectors, with each vector $\mathbf{v}_i \in \mathbb{R}^{d_v}$ encoding the semantic content of a region (or bounding box) within the image. We project coordinates of each region into vectors of d_v dimensions and sum them to their associated $\mathbf{v}_i$. For q, we use skip-thought vectors (Kiros et al. 2015a) to obtain its representation $\mathbf{q} \in \mathbb{R}^{d_q}$. We then merge each $\mathbf{v}_i$ with $\mathbf{q}$ using MLB, a multi-modal fusion module from J.-H. Kim et al. 2017, resulting in a new set of vectors $\{\mathbf{m}_i\}_{i\in\{1,\ldots,n_v\}}$ ready for relationship modeling and spatial counting, to be discussed below.

Self-attention. Since the set of bounding boxes used in encoding images can overlap, one core challenge for correct counting is to *de-duplicate* boxes (Y. Zhang et al. 2018; Trott et al. 2018) that are assigned to the same instance. Additionally, questions can require relational reasoning between objects. We address this by modeling general relationships among $\{\mathbf{m}_i\}$ using self-attention (Vaswani et al.

2017), letting the model learn the required mechanisms. Specifically, a single-head attention layer with a residual connection is applied on $\{\mathbf{m}_i\}$, yielding (for each region i) a contextualized representation $\{\mathbf{m}'_i\}$.

Spatial aggregation. After relationship modeling, the resulting $\{\mathbf{m}'_i\}$ vectors are then again merged with the question representation $\mathbf{q}$ using an MLB bilinear fusion and produce a counting score s_i for each region via sigmoid activation. Finally, the global count output $\hat{c} = \sum_i s_i$ is a simple summation of all the individual counting scores. We name our model *Spatial Counting Network*, because each and every count is explicitly grounded to a spatial region and allows for easy interpretation and visualization.

While the above-described model encapsulates general components like multimodal fusion and relationship modeling for visual counting, we would like to highlight **two design choices** that are important for improving its generalization, described next.

Regression, not classification. First, unlike many reference counting models (Y. Zhang et al. 2018; M. Acharya et al. 2019) and general VQA models, including large-scale pre-trained vision-and-language models (J. Lu et al. 2019; Tan and Bansal 2019) that treat count numbers as classification labels, we state they should be interpreted as actual numbers and we directly train the model to regress the final output $\hat{c}$ to the ground truth count label c. We choose the standard Mean Squared Error (MSE) as the loss:

$$\mathcal{L}_{\text{MSE}}(\theta; \mathcal{D}) = \frac{1}{n} \sum_{(v,q,c)\in\mathcal{D}} (\hat{c} - c)^2. \tag{4.1}$$

During testing, we round the fractional value $\hat{c}$ to its nearest integer to complete the mapping $f(v, q)$ to count labels. This loss is suited to counting, as it takes advantage of the natural order of the count labels. It also allows our model to output count labels that were not seen during training, which is beneficial when the testing set follows a different distribution of count labels.

Entropy regularization. Second, although regression is a natural choice for number-related tasks, directly applying it to visual counting can be disadvantageous, because it attempts to model the entire output counting range (*i.e.* $\hat{c}$ can be any real values between 0 and N) and does not take advantage of the fact that *all the count labels are integers*. One way to fix this is to select regions one by one, with discrete decisions, and train the model through reinforcement learning (Trott et al. 2018). However, the resulting objective function is hard to optimize directly.

Here we propose an alternative solution by simply imposing a binary entropy regularization term per region:

$$\mathcal{L}_H = -\frac{1}{n} \sum_{(v,q,c)\in\mathcal{D}} \left[\frac{1}{n_v} \sum_{i=1}^{n_v} s_i \log(s_i) + (1 - s_i) \log(1 - s_i) \right], \tag{4.2}$$

which essentially encourages each sigmoid output s_i to be close to 0 or 1. Intuitively, it means for each region, there is either one whole object or none – it won't be fractional (e.g. 0.5). This regularization not only enforces the final count $\hat{c}$ to be close to integers (since $\hat{c}$ is produced by summing up scores that are close to 0 or 1), but also benefits *grounding* the final count in the image (since it significantly reduces the chance of multiple overlapping regions being assigned some fractional value and summing up to be an integer count), which in turn helps generalization.

Our final training loss is a combination of MSE and entropy regularization: $\mathcal{L} = \mathcal{L}_{\text{MSE}} + \mathcal{L}_H$.

4.5 Experiments on SCN

4.5.1 Implementation details

Our SCN model We use the common Faster R-CNN (Ren et al. 2015) pre-trained by Anderson et al. 2018a to extract object features from the image, and the common GRU language model pre-trained by Kiros et al. 2015a to extract language features from the question. To keep a similar number of parameters with the state-of-the-art RCN model (M. Acharya et al. 2019), we use hidden dimensions of 1500 for the multimodal embeddings m_i, 500 for the self-attention, 768 for both bilinear fusions, and use only one self-attention head. We train our model for 30 epochs with the Adam optimizer (Kingma and Ba 2015) and a learning rate of 2.e-5 which is decayed by 0.25 every 2 epochs, starting at epoch 15. The learning rate schedule was tuned on the validation accuracy of the TallyQA-Odd-Even set. Importantly, for all other experiments, we use the same hyperparameters. We early stop training based on the highest accuracy computed on the validation set.

RCN We follow the implementation and hyperparameters described in (M. Acharya et al. 2019).

Training details All of our results are the average over 3 runs with different seeds. We report small standard deviations. Our SCN model takes 10 hours to

train on the original TallyQA and TallyQA-CP datasets. It takes about 6 hours to train on the TallyQA-Odd-Even dataset because it is composed of fewer triplets. We train our model on a single Titan X Pascal 12GB GPU.

4.5.2 Results

	TallyQA-CP				TallyQA-Odd-Even				
	Testing set		Validation set		Testing set		Validation set		
	Acc.	RMSE	Acc.	RMSE	Acc.	RMSE	Acc.	RMSE	#Param
Random $\mathcal{D}_{\mathbf{train}}$	19.53	2.84	22.13	2.77	10.26	2.81	32.35	2.89	–
Random $\mathcal{D}_{\mathbf{test}}$	20.40	2.89	19.78	2.81	30.68	2.61	10.21	2.75	–
RCN	2.00	1.76	**77.66**	1.30	28.4	1.61	**70.06**	1.34	47 M
RCN with $\mathcal{L}_{\text{MSE}}$	14.99	1.60	71.92	1.16	31.44	1.51	63.02	**1.26**	47 M
SCN (ours)	**34.79**	**1.46**	63.81	1.14	**40.87**	1.50	54.04	1.29	52 M
SCN w/o $\mathcal{L}_{\mathcal{H}}$	26.88	1.47	66.56	**1.11**	39.54	**1.48**	55.62	**1.26**	52 M

Table 4.6. – Results on TallyQA-CP and TallyQA-Odd-Even. We report the accuracy and the RMSE scores. SCN without $\mathcal{L}_{\mathcal{H}}$ stands for SCN without entropy regularization.

Main results In Table 4.6, we compare our model against the previous state-of-the-art approach RCN (M. Acharya et al. 2019). Scores for SCN are averaged over three runs, with a variance of 0.4 for accuracy and 0.01 for RMSE. On our TallyQA-CP dataset, we report the best accuracy of 34.79% for our SCN on the testing set, which corresponds to a +32.79 gain in accuracy points over RCN. On the TallyQA-Odd-Even dataset, our model reaches the best accuracy of 40.87%, with gains of +12.47 points over RCN. As expected, on both benchmarks we report lower performance than RCN on the validation set. Biased models such as RCN reach higher performances on in-distribution data by exploiting biases but fail on out-of-distribution data. Importantly, we also show that SCN does not overfit the testing set distribution: the validation accuracy is still higher than the testing set accuracy.

Impact of regression loss A notable difference between our model and state-of-the-art models such as RCN and Counter is that they are trained using classification instead of regression. For fair comparisons, we isolate the contribution of this design choice by introducing **RCN with $\mathcal{L}_{\text{MSE}}$**, which is a modified RCN that outputs a real number before rounding and is trained using the MSE loss. We train **RCN with $\mathcal{L}_{\text{MSE}}$** by changing the output dimension of the last linear

layer from 15 to 1. This allows us to train the model with an MSE regression loss instead of a classification loss. We use the same hyperparameters as RCN.

In Table 4.6, we report a gain of +12.99 points and +3.04 points over the regular RCN model on the TallyQA-CP test and TallyQA-Odd-Even test respectively. We conclude that **RCN with $\mathcal{L}_{\text{MSE}}$** is less sensible to biases. These good performances suggest that regression models are a better design choice to avoid learning biases. However, other design choices allow our model to reach further gains with +19.8 and +9.43 accuracy points against **RCN with $\mathcal{L}_{\text{MSE}}$**.

Impact of Entropy regularization. We also perform an ablation study of our SCN by training it without the entropy regularization (**SCN without $\mathcal{L}_{\mathcal{H}}$** in Table 4.6). We report an important effect on TallyQA-CP, with +7.91 points on its testing set. It shows that entropy regularization helps to generalize. Interestingly, it has very little impact on TallyQA-Odd-Even.

Difference in accuracy per count label. Gains in accuracy could be due to different patterns such as an important gain on only one count label or small gains on all of them. We study this in Figure 4.5, where we display a fine-grained comparison between our model and RCN according to their accuracy per count label. Interestingly compared to RCN, we report a higher accuracy on even count labels which are less represented in the training set and a lower accuracy on odd count labels which are more represented in the training set. We also report much smaller differences in accuracy between adjacent count labels, compared with RCN. For instance, we report a loss of -29.56 accuracy points between labels 1 and 2 compared to -85.15 with RCN. Overall, there is much less variation in our model between even and odd count labels. These results suggest that our design choices are useful to learn a proper mechanism of counting which helps to generalize to a different distribution of count labels.

Difference in accuracy on various shifts in distribution We create new TallyQA-Odd-Even variants by changing this removal proportion that was initially defined as 90% in Section 4.3. We vary the proportion from 70% to 100%, which controls the amount of biases that can be learned. On the extreme side, TallyQA-Odd-Even-100% generates a training set with no even count labels and a testing set with no odd count labels (i.e., a zero-shot setting). We also introduce the Even-Odd dataset, where the training and testing sets mostly contain even and odd count labels respectively, and also vary the removal proportion from 70% to 100%. We note those datasets Odd-Even-p% and Even-Odd-p%. In Figure 4.6, we compare the accuracy of our model against the state-of-the-art model RCN for visual counting, and its version with regression, RCN with $\mathcal{L}_{\text{MSE}}$, on the Odd-Even-p% and Even-

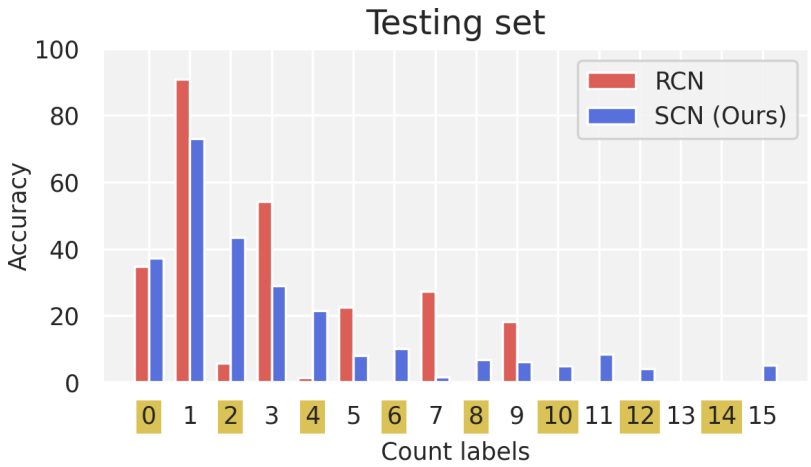


Figure 4.5. – Accuracy per count labels of our model and RCN on TallyQA-Odd-Even. Our model reaches higher accuracies on even labels (in yellow). These count labels are meant to penalize models that over-rely on biases.

Odd-p% datasets. We show that our model reaches significant and consistent gains. As expected, we report larger gains over RCN ranging from +12.78 accuracy points to +34.52 on datasets that possess the most important shift in distributions (e.g. p $>$ 80). We see similar gains over RCN with $\mathcal{L}_{\text{MSE}}$. Interestingly, our model is able to answer in the zero-shot setting ($p = 100\%$), reaching 26.87% and 34.52% accuracy for Odd-Even and Even-Odd respectively, while RCN has 0% accuracy. We perform similar experiments on a modified version of our TallyQA-CP dataset, where 10% of the samples are exchanged between training and testing to induce a lower shift in distributions. In this setup, it is expected that biased models such as RCN reach better performances than on the original TallyQA-CP. We report the best accuracy of 50.10% accuracy on the testing set for our SCN, which corresponds to a +9.9 gain in accuracy points over RCN.

More balanced TallyQA-CP version In Table 4.7, we show results on a modified version of our TallyQA-CP dataset where 10% of the examples from the training and testing are moved to the opposite set. We call this more balanced dataset TallyQA-CP-10%. Models are expected to perform better on it than on our main TallyQA-CP dataset. Our SCN model still reaches the best accuracy of 50.10% on the testing set. We also report better results when adding the entropy regularization to our SCN, and when using the regression loss $\mathcal{L}_{\text{MSE}}$ on RCN.

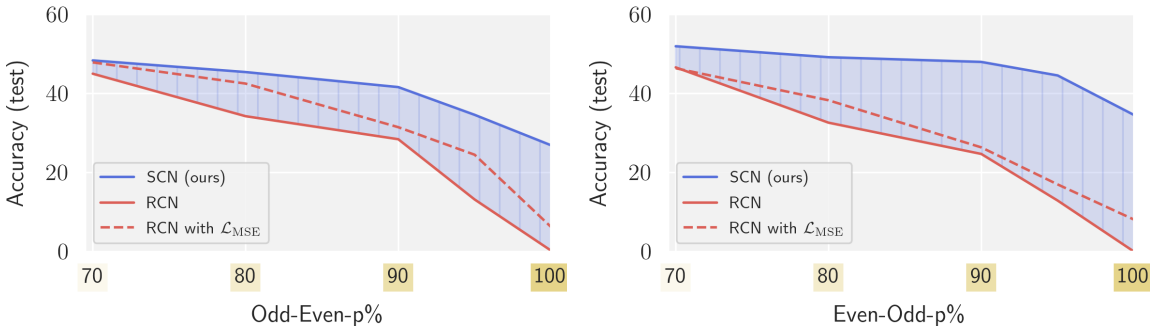


Figure 4.6. – Comparison between our model, RCN and its regression variant on various versions of TallyQA using our Odd-Even-p% and Even-Odd-p% datasets. p% controls the shift in distributions between the training and testing sets (with the original distribution when p = 0). Models that over-rely on biases (*e.g.* original RCN) are strongly penalized when p% is high (yellow gradient).

	Testing set		Validation set	
	Acc.	RMSE	Acc.	RMSE
Question Only	12.67	2.12	58.84	1.98
I-Only	24.99	2.18	37.73	2.16
Q+I	26.89	1.73	66.87	1.55
MUTAN (Ben-Younes et al. 2017a)	27.52	1.69	68.00	1.50
Counter (Y. Zhang et al. 2018)	21.33	1.83	65.91	1.68
RCN	40.21	1.46	**71.52**	1.30
RCN with $\mathcal{L}_{\text{MSE}}$	43.99	1.34	65.99	**1.2**
SCN without $\mathcal{L}_H$	48.16	1.28	61.42	**1.2**
SCN	**50.10**	**1.30**	60.15	1.23

Table 4.7. – Results on a more balanced TallyQA-CP dataset where 10% of examples have been moved between the training and testing sets.

4.5.3 Study of the grounding ability

COCO-Grounding. Similarly to the work done in IRLC (Trott et al. 2018), we use the grounding ability as a proxy to evaluate the proper counting mechanism and to assess the interpretability of models. To this end, we introduce the COCO-Grounding dataset that, contrary to previous works, allows us to compare models that use different visual features than ours. The grounding ability can be evaluated in a similar way to object detection models. To this end, we specifically design a dataset named *COCO-Grounding*. We create questions automatically from COCO images based on the provided annotations and save for each image-question pair

	COCO-Grounding	TallyQA	
	AP@.50	Acc.	RMSE
SCN (ours)	**10.90**	55.54	1.25
SCN without $\mathcal{L}_{\mathcal{H}}$	7.63	57.07	1.24
SCN without Self-att.	9.10	54.38	1.39
Counter* (Y. Zhang et al. 2018)	6.44	60.58	1.37
Counter (Y. Zhang et al. 2018)	–	62.58	1.34
RCN (M. Acharya et al. 2019)	not evaluable	64.41	1.28

Table 4.8. – Grounding ability of models trained on original TallyQA dataset. AP@.50 on COCO-Grounding is a classic metric for object detection. Low AP@.50 values are expected because these models were not trained using the bounding boxes class annotations. Counter* (Y. Zhang et al. 2018) was retrained by us.

the bounding boxes of the objects to be counted. Our dataset is composed of the 4459 images from MSCOCO (T.-Y. Lin et al. 2014b) that can not be found in Visual Genome (Krishna et al. 2017b) and importantly not in the TallyQA training set. Each MSCOCO image is annotated with bounding boxes around objects associated with a category among 80 classes of objects. We use these classes to automatically generate simple questions about a given image using the "How many {class}?" pattern. The answer to a question is the number obtained by counting the bounding boxes associated with the given {class}. We also generate questions associated with the count label **0** by sampling a random class among 80 that is not present on the image. We generate an equal number of 734 *image-question-count* triplets associated with the count label 0, 1 and 2, and generate all possible triplets for higher count labels (with a maximum label of 15) to reach a total number of 3311 triplets over 2139 images.

Evaluation metrics Similarly to object detection models, our model can output bounding box predictions. Therefore, we use the Average Precision (AP), a standard metric in object detection tasks (Everingham et al. 2015; T.-Y. Lin et al. 2014b). It allows us to evaluate the ability of our model to detect the correct instances of objects to count in the image. We use the AP@.50 metric, also used in the COCO (T.-Y. Lin et al. 2014b) and PASCAL-VOC (Everingham et al. 2015) challenges.

Results In Table 4.8, we compare our SCN against Counter (Y. Zhang et al. 2018) on the mean average precision AP@.50, an object detection metric, with an IoU threshold of 0.5. Both models have been trained on the original TallyQA dataset.

We report the best performances on grounding and a gain of +4.5 points over Counter. As expected, we report a lower accuracy on the TallyQA testing set since models that over-rely on biases are not penalized. We do not compare against RCN (M. Acharya et al. 2019), because it does not internally associate counting numbers to regions of the image. We also highlight the importance of entropy regularization (+3.3 points) and self-attention mechanism (+1.9) on grounding. This justifies our choices in architecture and regularization.

4.5.4 Qualitative results

In Figures 4.7, 4.7 and 4.9, we display representative examples of outputs of our model with (on the left) and without (on the right) entropy regularization. We display bolded red bounding boxes around objects when their associated count value c_i is close to 1. In Figure 4.7, we display the bounding box scores for the question 'How many people are in the picture?'. Both model predicts the same answer after rounding, but we see an important difference in the bounding box scores. With entropy regularization, SCN is able to select the four bounding boxes corresponding to people in the image. On the other hand, our model without entropy fails to distinguish duplicates and associates fractional values to multiple regions. In Figure 4.8, We compare both models on the questions 'How many giraffes are shown?' and 'How many zebras are shown?'. We observe similar observations for both of those questions: SCN with entropy regularization assigns high scores to the regions containing the objects to be counted, and the model without this regularization fails to do so. Additionally, the model without entropy regularization answers incorrectly to the first question: it predicts 2.71, rounded to 3, while the correct answer is 2.

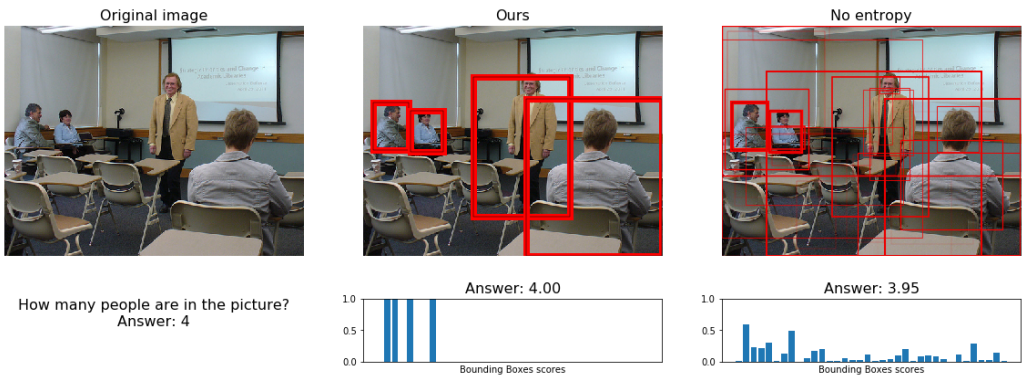


Figure 4.7. – Qualitative comparison of bounding box scores for our SCN with and without entropy regularization. Both models are correct, but our model with entropy regularization selects the correct regions.

In Figure 4.9, we display two complex questions on the same image, and show that our SCN model is able to select the correct object (*people*) according to an attribute (*is he playing tennis or football?*), and output a correct count (*1* or *0*).

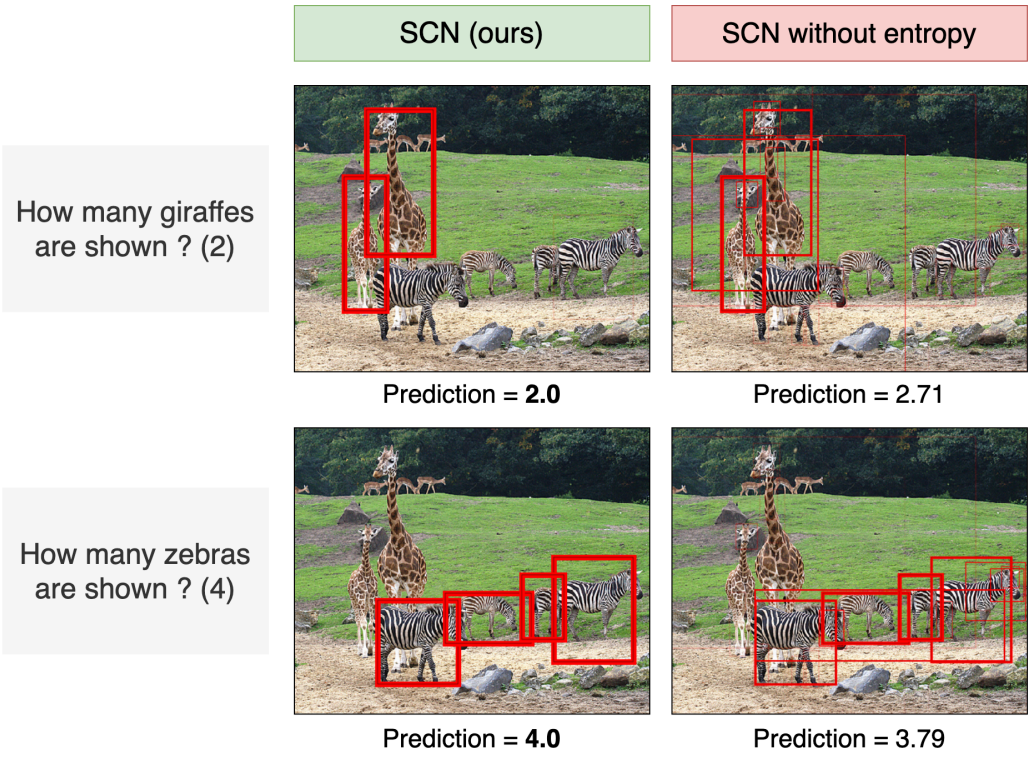


Figure 4.8. – Qualitative comparison between our model with and without entropy regularization. Red bounding boxes are shown with bolded borders when their associated c_i is close to 1.

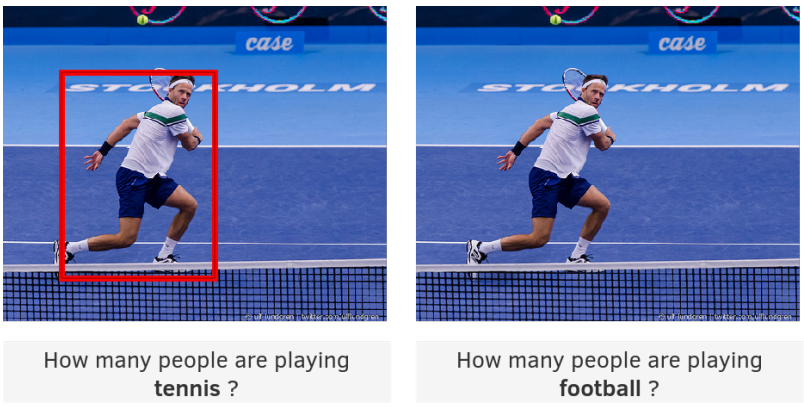


Figure 4.9. – Regions selected by our SCN model for two complex questions on the same image. SCN answers are respectively 1 and 0.

4.6 Conclusion

In this chapter, we explore a second direction in reducing shortcut learning: using **architectural priors** to constrain the network to learn the correct mechanism and prevent it to learn spurious correlations. We focus on the task of visual counting, as the output distribution is simpler than the original VQA, and the answering mechanism is more constrained. First, we introduce two out-of-distribution datasets to penalize models that have learned dataset biases. The first, TallyQA-CP, has a distribution shift of answers conditioned to the main concept in the question, similar to VQA-CP. It tests the model's reliance on question-based shortcuts. The second, TallyQA-Odd-Even, has a distribution shift of answers conditioned to the parity of the answer. Contrarily to TallyQA-CP, it tests the model's reliance on all superficial correlations between the input and the answer. On our datasets, we show that reference models suffer from large performance loss which indicates that they have learned biases. We then introduce the Spatial Counting Network (SCN), a model that encompasses architectural priors to encourage the learning of the correct counting mechanisms. We validate the interest of each design choice and showed that our model is better at selecting the correct objects to count and less prone to learn biases. We also note that our proposed model is more *explainable by design* than previous approaches, as it is easy to interpret the reasoning process of the model: each region gets assigned a score. This makes it easier for a user to trust the model's predictions.

Although the task of visual counting is fairly constrained, this is a step towards deep neural networks that learn to reason. Our work is an example of an interpretable model on top of black-box modules that can be used to learn more complex reasoning mechanisms. We believe that this is a promising direction for future research.

CHAPTER 5

DETECTING MULTIMODAL SHORTCUTS FOR VQA

Chapter abstract

In Chapter 3, as in most of the Visual Question Answering (VQA) literature, the most studied shortcuts are those coming from the question modality. This makes models rely, for some examples, solely on the question, without considering the image information. However, this covers only a small part of all potential shortcuts that can be exploited by models. In this chapter, we go a step further and explore the existence of multimodal shortcuts that involve both questions and images. We identify potential shortcuts in the popular VQA v2 training set by mining shallow predictive rules such as co-occurrences of words and visual elements. We then introduce VQA-CounterExamples (VQA-CE), an evaluation protocol based on our subset of CounterExamples i.e. image-question-answer triplets where our rules lead to incorrect answers. We use this new evaluation in a large-scale study of existing approaches for VQA. We demonstrate that even reference models perform poorly and that existing techniques to reduce biases are largely ineffective in this context. Our findings suggest that past work on question-based biases in VQA has only addressed one facet of a complex issue.

The work in this chapter has led to the publication of this conference paper:

- Corentin Dancette, Remi Cadene, Damien Teney, and Matthieu Cord (2021b). "Beyond Question-Based Biases: Assessing Multimodal Shortcut Learning in Visual Question Answering". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Contents

5.1 Introduction

In Chapter 3, we mostly studied the issue of VQA shortcuts coming from the question modality: they were superficial statistical patterns in the training data that allow predicting correct answers by using mostly the textual information, without deploying the desirable behavior. This is also the case in most of the literature related to biases in the VQA task. But shortcuts might be more subtle and involve both textual and visual elements. For instance, training questions containing *What sport* are strongly associated with the answer *tennis* when they co-occur with a racket in the image (see Figure 5.1). This seems to be a valid answering strategy, but there are situations where this pattern will fail. Some examples can be found in the validation set, such as *What sport field is in the background ?*, that lead to a different answer (*soccer*) despite a racket being present in the image. Because of such exceptions, a model that strongly relies on simple co-occurrences will fail on unusual questions and scenes. Most previous work and existing evaluation protocols are limited to text-based shortcuts. This chapter studies multimodal biases and their impact on VQA models.

Our work introduces VQA-CounterExamples (VQA-CE), an evaluation protocol for multimodal shortcuts. It is easy to reproduce and can be used on any model trained on VQA v2, without requiring retraining. We first start with a rule-mining-based method to discover superficial statistical patterns in a given VQA dataset that could be the cause of shortcut learning. We discover a collection of co-occurrences of textual and visual elements that are strongly predictive of

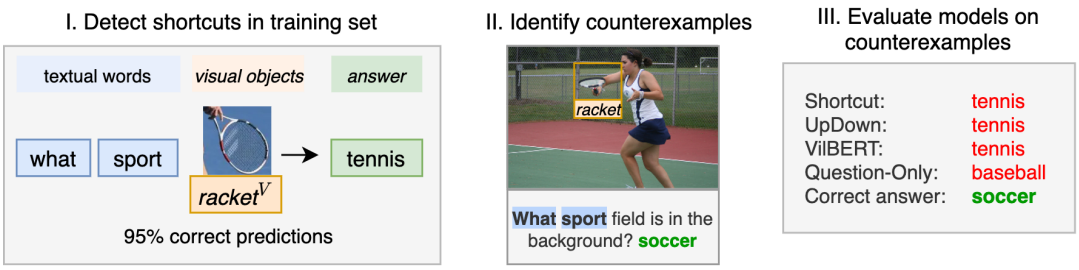


Figure 5.1. – Overview of this work. We first mine simple predictive rules in the training data such as what + sport + racket^V $\rightarrow$ **tennis**. We then search for counterexamples in the validation set that identify some rules as undesirable statistical shortcuts. Finally, we use the counterexamples as a new challenging test set and evaluate existing VQA models like UpDown (Anderson et al. 2018a) and VilBERT (J. Lu et al. 2019).

certain answers in the training data and often transfer to the validation set. For instance, we discover a rule that relies on the appearance of the words "what", "they", "playing" together with the object "controller" in the image to always predict the correct answer "wii". We consider this rule to be a shortcut since it could fail on arbitrary images with other controllers, as it happens in the real world. Thus, our method can be used to reflect biases of the datasets that can potentially be learned by VQA models. We go one step further and identify counterexamples in the validation set where the shortcuts produce an incorrect answer. These counterexamples form a new challenging evaluation set for our VQA-CE evaluation protocol. This benchmark addresses some of the shortcomings of VQA-CP: First, it evaluates models trained on the original VQA v2 training set and does not require retraining. Second, it evaluates against real shortcuts, instead of artificially-created correlation. Finally, we propose a clear evaluation setup with an in-distribution validation set to avoid test set overfitting. We find that the accuracy of existing VQA models is significantly degraded on this data. More importantly, we find that most current approaches for reducing biases and shortcuts are ineffective in this context. They often reduce the average accuracy over the full evaluation set without significant improvement on our set of counterexamples. Finally, we analyze models to find which shortcuts they exploit by comparing their predictions to the shortcut predictions.

In Section 5.3, we propose **a method to discover shortcuts** which rely on the appearance of words in the question and visual elements in the image to predict the correct answer. By applying it to the widely-used VQA v2 training set, we find a high number of multimodal shortcuts that are predictive on the validation set. Then, in Section 5.4, we introduce **the VQA-CE evaluation protocol** to assess the VQA models' reliance on these shortcuts. By running a large-scale evaluation

of recent VQA approaches, we find that reference VQA models exploit these shortcuts and that bias-reduction methods are ineffective in this context.

5.2 Related Work

We review existing approaches to discovering potential statistical shortcuts and assess their use by learned models.

Detecting cases of shortcut learning The general methods to detect shortcut learning are explained in Section 2.3 and 2.4.

The closest approach to the work in this chapter, Manjunatha et al. 2019b, uses the Apriori algorithm on VQA v2 to extract predictive rules that combine the appearance of words and visual contents. However, these rules are specific to the attention maps and predictions of the VQA model from Kazemi and Elqursh 2017. They are extracted on the validation set and are mainly used for qualitative purposes. Our approach also relies on the Apriori algorithm but extracts rules directly on the training set, independently of any model, and the predictive capacity of the rules is evaluated on the validation set. We then propose an evaluation benchmark based on those shortcuts.

Evaluating VQA models' reliance on shortcuts We discuss extensively evaluation benchmarks in Section 2.4.2. Once a class of shortcuts has been identified, a way to evaluate models' robustness is to build external out-of-distribution evaluation datasets on which using these shortcuts leads to a wrong prediction.

The main dataset used for the evaluation of biases is VQA-CP, presented in Chapter 2. Our proposed evaluation has a few differences from VQA-CP. First, it does not require retraining the model: this enables us to evaluate any model trained on VQA v2, instead of evaluating a given training procedure. Second, we focus on multimodal shortcuts, instead of text-based shortcuts. We follow guidelines from (D'Amour et al. 2020; Teney et al. 2020b) for a better evaluation of the use of shortcuts.

The closest work to ours is the GQA-OOD (Kervadec et al. 2021) dataset: they extract from the GQA (Drew A Hudson and Manning 2019) validation and testing set example with rare answers, conditioned on the type of question. It enables the evaluation of models without retraining on a separate training set.

Frequent itemset mining Frequent itemset mining techniques have been used extensively for database analysis (R. Agrawal, Srikant, et al. 1994; Uno et al. 2003).

They have also been used more recently for sequence prediction (Bourrand et al. 2021) or computer vision tasks (Quack et al. 2007; Yuan et al. 2007; Fernando et al. 2012). For example, Fernando et al. 2012 propose to mine *Frequent Local Histograms* for image classification tasks. In this chapter, we propose to use frequent itemset mining to find superficial decision rules for the VQA task.

5.3 Detecting multimodal shortcuts in VQA

5.3.1 Our shortcut detection method

We introduce our method to detect shortcuts relying on textual and visual input. Our approach consists in building a dataset of input-output variables and applying a rule-mining algorithm. The code for our method is available online [1]. We consider the VQA formulation specified in Section 2.2.2: we have a training set $\mathcal{D}_{train}$ made of n triplets $(v_i, q_i, a_i)_{i\in[1,n]}$ with $v_i \in \mathcal{V}$ an image, $q_i \in \mathcal{Q}$ a question in natural language and $a_i \in \mathcal{A}$ an answer. VQA is usually cast as a problem of learning a multimodal function $f : \mathcal{V} \times \mathcal{Q} \rightarrow \mathcal{A}$ that produces accurate predictions on $\mathcal{D}_{test}$ of unseen triplets.

Mining predictive rules on a training set Our goal is to detect shortcuts that a VQA model f might use to provide an answer without deploying the desired behavior. To this end, we limit ourselves to a class of shortcuts that we hypothesize to be often leveraged by VQA models. We display in Figure 5.2 our rule-mining process. These shortcuts are short predictive association rules $A \rightarrow C$ that associate an **antecedent** A to a **consequent** C. Our antecedents are composed of words of the question and salient objects in the image (or image patch), while our consequents are just answers. For instance, the rule {what, color, plant} $\rightarrow$ {**green**} provides the answer "green" when the question contains the words "what", "color" and "plant". These shallow rules are by construction shortcuts. They are predictive on the validation set but do not reflect the complex behavior that needs to be learned to solve the VQA task. For instance, they do not rely on the order of words, nor the position and relationships of visual contents in the image. They lack the context that is required to properly answer the question. Moreover, even rules that seem correct often have counterexamples in the dataset, i.e. examples that are matched by the antecedent but the consequent provides the wrong answer. We later use these counterexamples in our evaluation procedure.

1. https://github.com/cdancette/detect-shortcuts

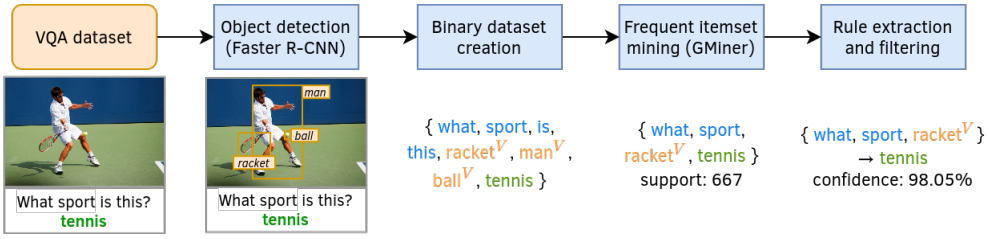


Figure 5.2. – Pipeline of the proposed method to detect potential shortcuts in a VQA training set. We detect and label objects in images with a Faster R-CNN model. We then summarize each VQA example with binary indicators representing words in the question, answer, and labels of detected objects. Finally, a rule mining algorithm identifies frequent co-occurrences and extracts a set of simple predictive rules.

Binary dataset creation To detect these rules, we first encode all question-image-answer triplets of $\mathcal{D}_{train}$ as binary vectors. Each dimension accounts for the presence or absence of **(a)** a word in the question, **(b)** an object^V in the image, represented by its textual detection label from a Faster R-CNN model (Anderson et al. 2018a), **(c)** an **answer**. The number of dimensions of each binary vector is the sum of the size of the dictionary of words (e.g. ~13,000 words in VQA v2), the number of detection labels of distinct objects in all images (e.g. 1,600 object labels), and the number of possible answers in the training set (e.g. 3,000 answers). We additionally report results with ground truth instead of detected labels in Appendix B, Table B.1.

Frequent itemset mining On our binary dataset, we apply the GMiner algorithm (Chon et al. 2018) to efficiently find frequent *itemsets*. An itemset is a set of tokens $\mathcal{I} = \{i_1, .., i_n\}$ that appear very frequently together in the dataset. The **support** of the itemset is its number of occurrences. For example, the itemset {what, color, plant, **green**} might be very common in the dataset and have high support. GMiner takes one parameter, the minimum support. We include an additional parameter, which is the maximum length for an itemset. We detail how we select parameters at the end of this section.

Rules extraction and filtering The next step is to extract rules from the frequent itemsets. First, we filter out the itemsets that do not contain an answer token, as they cannot be converted to rules. For the others that do contain an answer a, we remove it from the itemset to create the antecedent $\mathcal{X}$ ($\mathcal{X} = \mathcal{I} \setminus a$). The rule is then $\mathcal{X} \Rightarrow a$. The **support** s of the rule is the number of occurrences of $\mathcal{X}$ in the dataset. The **confidence** c of the rule is the frequency of correct answers among examples that have $\mathcal{X}$. We then proceed to filter rules. We apply the following three steps:

(a) We remove the rules with confidence on the training set lower than 30% ($c < 0.3$).

(b) If some rules have the same antecedent but different answers, then we keep the rule with the highest confidence and remove the others. For instance, given the rules {is, there} ⇒ **yes** and {is, there} ⇒ **no** with a respective confidence of 70% and 30%, we only keep the first one with the answer **yes**.

(c) if a rules r_1's antecedent is a superset of another rule r_2's antecedent if both have the same answer, and r_1 has equal or lower confidence than r_2, then we remove r_1. For instance, given the rules {is, there} ⇒ **yes** and {is, there, cat} ⇒ **yes** with a respective confidence of 70% and 60%, we only keep the first one without the word *cat*.

We consider the remaining rules as shortcuts. Note that rules with a confidence of 100% could be considered *correct* and not shortcuts, but these rules will not influence our evaluation protocol, detailed in Section 5.4.

5.3.2 Analysis of shortcuts on natural data

We analyze the shortcuts that our approach can detect on the VQA v2 dataset. We extract ensembles of rules with different combinations of minimum support and confidence. Each time, we aggregate them into a classifier that we evaluate on the validation set. We detail how to build this kind of classifier in Section 5.4.3. We select the support and confidence leading to the best overall accuracy. It corresponds to a minimum support of $2.1 \cdot 10^{-5}$ (about ∼8 examples in training set), and a minimum confidence of 0.3. Once these shortcuts have been detected, we assess their number and type (purely textual, purely visual, or multimodal). We also verify that they can be used to find counterexamples that cannot be accurately answered using shortcuts. Finally, we evaluate their confidence on the validation set. In the next section, we leverage these counterexamples with our VQA-CE evaluation protocol to assess models' reliance on shortcuts.

Words-only and objects-only shortcuts First, we show that our approach is able to detect shortcuts that are purely textual or visual. In the first row of Figure 5.3, we display a shortcut detected on VQA v2 that only accounts for the appearance of words in the question. It predicts the answer "white" when the words "what", "color", "is", "snow" appear at any position in the question. In the training set, these words appear in 95 examples and 90.62% of them have the "white" answer. This shortcut is highly predictive on the validation set and gets 95.65% of correct answers over 92 examples. We also display an example in which exploiting the shortcut leads to the correct answer and a counterexample in which the shortcut

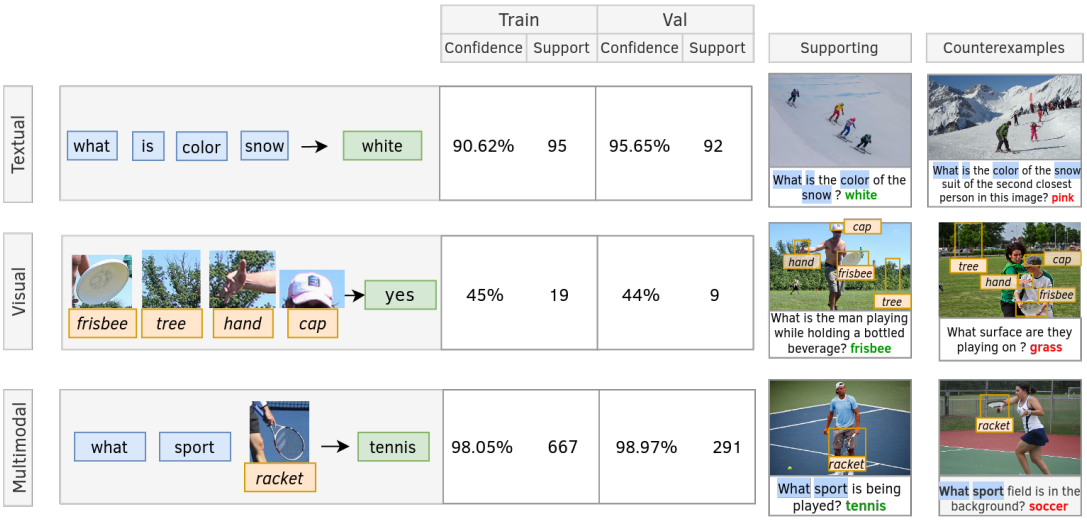


Figure 5.3. – Examples of shortcuts found in the VQA v2 dataset. The confidence is the accuracy obtained by applying the shortcut on all examples matching by its *antecedent*. The support is the number of matching examples. More supporting examples and counterexamples are shown in Figure 5.6.

fails because the question was about "the color of the snow suit" which is "pink". In the second row, we show a shortcut that only accounts for the appearance of visual objects. It predicts "yes" when a "frisbee", a "tree", a "hand" and a "cap" appear in the image. However, this kind of shortcut is usually less predictive since they cannot exploit the question-type information which is highly correlated with certain answers, i.e. "what color" is usually answered by a color.

Multimodal shortcuts Then, we show that our approach is able to detect multimodal shortcuts. They account for the appearance of both words **and** visual objects^V. In the third row of Figure 5.3, we display a multimodal shortcut that predicts "tennis" when the words what, sport and a racket^V appear. It is a common pattern with a confidence of 98.05% based on a support of 667 examples in the training set. It is also highly predictive on the validation set with 98.97% confidence and 291 support. At first sight, it is counter-intuitive that this simple rule is a shortcut but answering complex questions is not about detecting frequent words and objects in images that correlate with an answer. In fact, this shortcut is associated with counterexamples where it fails to answer accurately. Here, the sport that can be played in the background is not tennis but soccer.

Number of shortcuts and statistics per type Here we show that our approach can be used to detect a high number of multimodal shortcuts. Overall, it detects ~1.12M shortcuts on the VQA v2 training set. As illustrated in Figure 5.4, since

there are ~413K examples, it is often the case that several shortcuts can be applied to the same example. This is the main reason behind the high number of shortcuts. For instance, the antecedent {animals, what, giraffe^V} overlaps with {animals, these, what, giraffe^V}. Among all the shortcuts that our method can detect, only ~50k are textual, ~77k are visual and ~1M are multimodal. In other words, ~90% are multimodal. In addition to being more numerous, they are also more predictive. For instance, the most confident shortcut that matches an example, highlighted in green in Figure 5.4, is multimodal 91.80% of the time. Finally, ~3K examples are not matched by any shortcut antecedents. They have unusual question words or visual content. We later do not take them into account in our VQA-CE evaluation protocol. We display some representative examples in Section 5.4.2

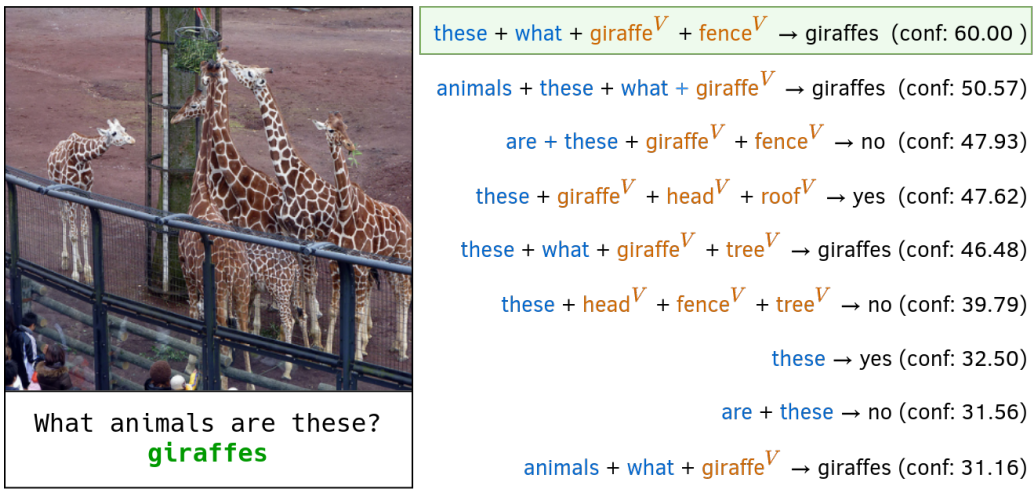


Figure 5.4. – Multiple shortcuts can often be exploited to find the correct answer in any given example. The confidence is the percentage of accurate answers among examples that are matched by the shortcut *antecedent*. The shortcut of highest confidence (in green) is multimodal for ~92% of examples.

Confidence distribution on training and unseen data Here we show that shortcuts detected on the VQA v2 training set transfer to the validation set. In Figure 5.5, we display the confidence distribution of these shortcuts. As told earlier, we only consider shortcuts that reach a confidence greater than 0.3 on the training set. The number of shortcuts decreases when confidence increases. It is expected to find fewer shortcuts with higher levels of confidence due to the collection procedure of VQA v2 which focused on reducing the amount of data biases and shortcuts. We evaluate on the validation set the same shortcuts detected on the training set and also display the confidence distribution. We show that our shortcuts are predictive on both training data, and unseen data that follows the training set distribution. The number of shortcuts that reach a confidence

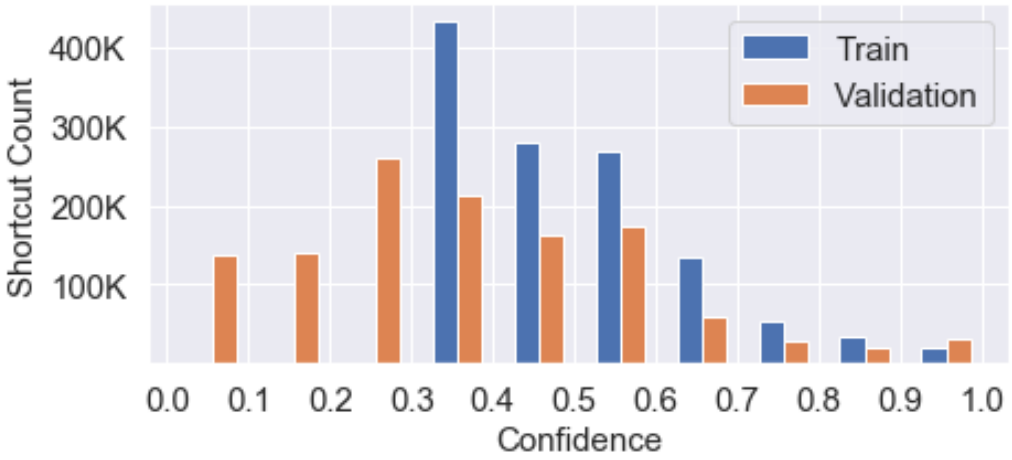


Figure 5.5. – Histogram of shortcuts binned per confidence on the VQA v2 training and validation sets. Our shortcuts are detected on the training set and selected to have a confidence above 30%. Even though their confidence could be expected to be lower on the validation set, it still is above 30% for a large number of them, indicating that the selection transfers well to the validation set.

between 0.9 and 1.0 is even higher on the validation set than on the training set. The confidences are overall slightly lower on the validation set, but a large number of them are still above 0.3, indicating that they generalize to new examples from the same distribution. The great majority of shortcuts, which obtain a confidence lower than 1.0, allow finding examples that contradict them by leading to the wrong answers. We manually verified by looking at these examples that only a minority are wrongly annotated or ambiguous, most of them are counterexamples. These counterexamples are the core of our approach to assess the VQA model's reliance on shortcuts.

5.3.3 Identifying most exploited shortcuts

We introduce a method to identify shortcuts that may be exploited by a given model. On the validation set, we calculate for each shortcut a correlation coefficient between its answer and the predictions of a VQA model. Importantly, a 100% correlation coefficient indicates that the model may exploit the shortcut: both always provide the same answers, even on counterexamples on which using the shortcuts leads to the wrong answer.

In Table 5.1, we report shortcuts that obtain the highest correlation coefficient with UpDown (Anderson et al. 2018a) and VilBERT (J. Lu et al. 2019). Overall, these shortcuts have high confidence and support, which means that they are

common in the dataset and predictive. Most importantly, they are multimodal. As a consequence, these shortcuts obtain low correlations with Question-Only (Goyal et al. 2017b). On the contrary, they obtain a 100% correlation coefficient with ViLBERT and UpDown. For instance, the second shortcut provides the answer **skateboarding** for the appearance of sport, this, what in the question and a skateboard^V in the image. It is a common pattern with a support of 31 examples in the validation set. It gets a correlation of 0% because Question-Only mostly answers baseball for these examples. Its confidence of 87.1% indicates that 4 counterexamples can be found where the shortcut provides the wrong answer. To be correctly answered, they require more than a simple prediction based on the appearance of words and salient visual contents. These results suggest that VQA models tend to exploit multimodal shortcuts. It shows the importance of taking them into account in an evaluation protocol for VQA.

5.3.4 Rules with supporting examples and counterexamples

In Figure 5.6, we display some counterexamples to some rules displayed in Table 5.1. Some of those examples are "true" counterexamples, where the input does match the rule's antecedent, but the answer is different. For instance, in the first example of the first rule, the question is actually about the clothes and not the sport, and the man is dressed in a basketball outfit. On the contrary, some examples are there due to incorrect object detection: in the second example of the first rule, the object detection module detected a skateboard instead of a scooter. Thus, the example is incorrectly matched.

5.4 Evaluation: Assessing models' reliance on shortcuts

The classic evaluation protocol in VQA consists in calculating the average accuracy over all the examples. Instead, we introduce the VQA-CounterExamples (VQA-CE) evaluation protocol that additionally calculates the average accuracy over a specific subset of the validation set. This subset is made of counterexamples that cannot be answered by exploiting shortcuts. Models that do exploit shortcuts are expected to get a lower accuracy. It is how we assess the use of shortcuts. Importantly, our protocol does not require retraining as was the case with the previous protocol. We first detail the subsets creation procedure at the core of our VQA-CE protocol. We then run extensive experiments to assess the use of shortcuts on many VQA models and bias-reduction methods.

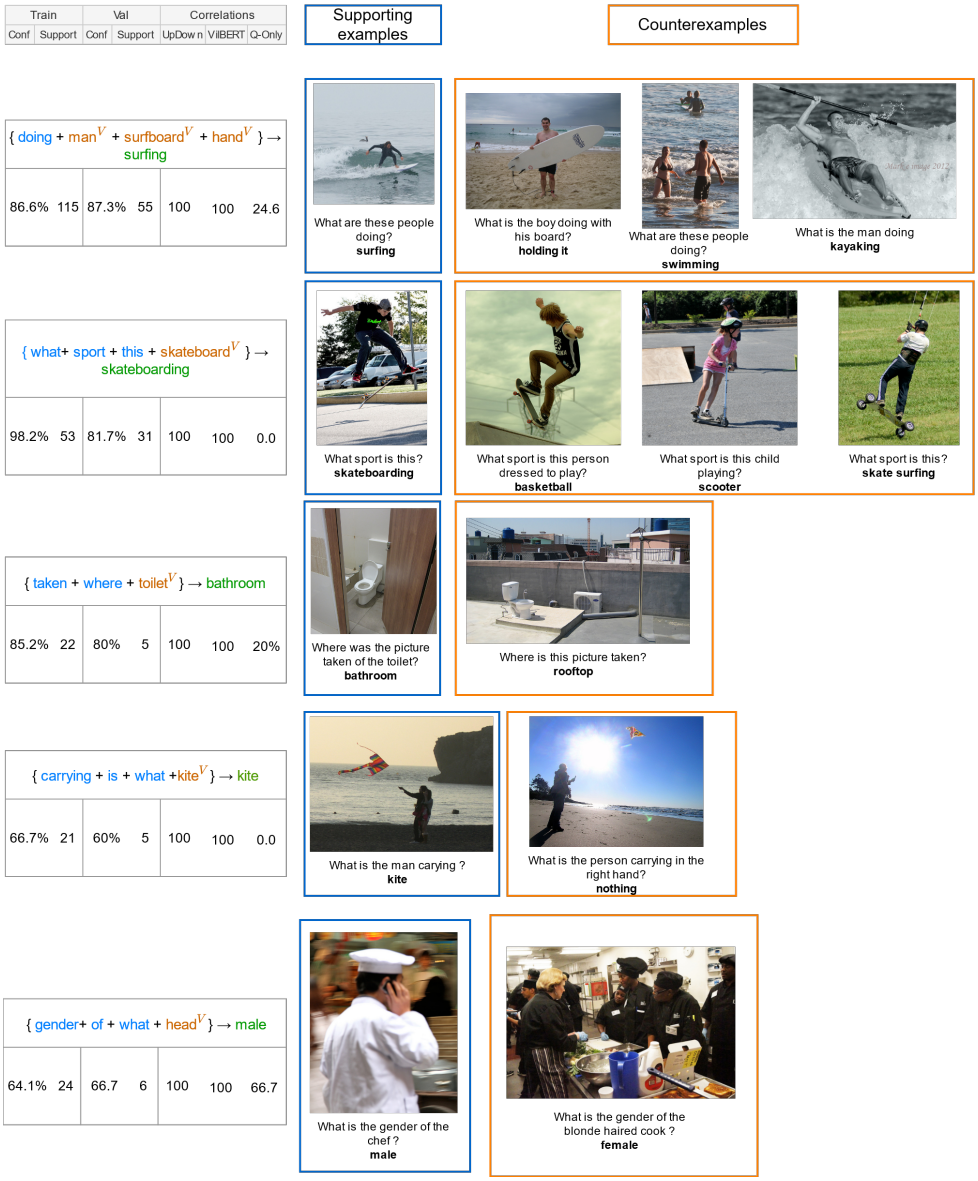


Figure 5.6. – Shortcuts that are highly correlated with VQA models' predictions. We display their antecedent made of words from the question and objects^V from the image, and their **answer**. Their support, i.e. number of examples matched by the antecedent, and confidence, i.e. percentage of correct answers among them, have been calculated on the VQA v2 training and validation sets. We report the correlation coefficients of their predictions with those of three VQA models: Up-Down [3] which uses an object detector, VilBERT [31] that has been pre-trained on a large dataset, and Q-only [21] that only uses the question. We also display some supporting examples, in blue, and counterexamples, in orange.

Rule (antecedent → consequent)	Train	Val	Correlations (Val)		
	Conf. (Sup.)	Conf. (Sup.)	UpDown	VilBERT	Question-Only
doing + man^V + surfboard^V + hand^V → **surfing**	86.6 (115)	87.3 (55)	100.0	100.0	23.6
sport + this + what + skateboard^V → **skateboarding**	98.2 (53)	87.1 (31)	100.0	100.0	0.0
holding + this + what + racket^V → **tennis racket**	75.0 (26)	33.3 (3)	100.0	100.0	33.3
played + shorts^V + racket^V + leg^V → **tennis**	100.0 (29)	80.0 (5)	100.0	100.0	40.0
playing + they + what + controller^V → **wii**	100.0 (30)	88.9 (9)	100.0	100.0	66.7
picture + where + beach^V + people^V → **beach**	100.0 (21)	90.0 (10)	100.0	100.0	90.0
taken + where + toilet^V → **bathroom**	85.2 (22)	80.0 (5)	100.0	100.0	20.0
eating + what + pizza^V + arm^V → **pizza**	81.5 (21)	66.7 (6)	100.0	100.0	66.7
carrying + is + what + kite^V → **kite**	66.7 (21)	60.0 (5)	100.0	100.0	0.0
gender + of + what + head^V → **male**	64.1 (24)	66.7 (6)	100.0	100.0	66.7
position + helmet^V + bat^V + dirt^V → **batter**	61.8 (20)	71.4 (7)	100.0	100.0	0.0

Table 5.1. – Instances of shortcuts that are highly correlated with VQA models' predictions. We display their antecedent made of words from the question and objects^V from the image, and their **answer**. Their support, i.e. number of examples matched by the antecedent, and confidence, i.e. percentage of correct answers among them, have been calculated on the VQA v2 training and validation sets. We report the correlation coefficients of their predictions with those of three VQA models: UpDown (Anderson et al. 2018a) that uses an object detector, VilBERT (J. Lu et al. 2019) that has been pre-trained on a large dataset, and Q-only (Goyal et al. 2017b) that only uses the question. We show some counterexamples in Figure 5.6.

5.4.1 Our VQA-CE evaluation protocol

Subsets creation using shortcuts By leveraging the shortcuts that we have detected before, we build the **Counterexamples** subset of the VQA v2 validation set. This subset is made of 63,298 examples on which all shortcuts provide the incorrect answer. As a consequence, VQA models that exploit these shortcuts to predict will not be able to get accurate answers on this kind of example. They will be penalized and obtain a lower accuracy on this subset. On the contrary, we build the non-overlapping **Easy** subset. It is made of 147,681 examples of which at least one shortcut provides the correct answer. On this subset, VQA models that exploit shortcuts can reach high accuracy. Finally, 3,375 examples are not matched by any shortcut's antecedent. Since these examples do not belong to any of our two subsets, we do not consider them in our analysis. As we show later in Section 5.4.2, they have unusual questions and images.

Distribution of examples Here, we show how the split between our two subsets Counterexamples and Easy affects the distribution of examples. In Figure 5.7, we show that the original distribution of answers is similar to the Easy distribution but dissimilar to the Counterexamples distribution. Highlighted in blue, we display the five most common answers from the Easy distribution. They can be found at the same positions in the original distribution, the two major answers being "yes" and "no". It is not the case in the Counterexamples subset where these answers appear less frequently. Nonetheless, they are still in the top 30 answers which shows that our subsets creation is not a trivial splitting between frequent and rare answers. Similarly, the five most common answers from the Counterexamples subset, highlighted in orange, can be found in the Easy and All subset. Next, we report similar observations for the questions and answer-type distributions.

Distribution of examples per question-type In Figure 5.8, we display the distribution of examples per question type, and their split between the Easy and the Counterexamples split. We show that examples of a question type that can be answered by *yes* or *no*, such as *is*, *are*, *does*, *do*, mostly belong to the Easy subset. Examples of a question-type beginning by *what*, *where* or *why* mostly belong to the Counterexamples subset. These examples need to be answered using a richer vocabulary than *yes* or *no*. Examples of a question-type beginning by *how* belong to both subsets.

Distribution of examples per answer type In Figure 5.9, we display the distribution of examples in our two subsets per answer type. We see that most yes-no

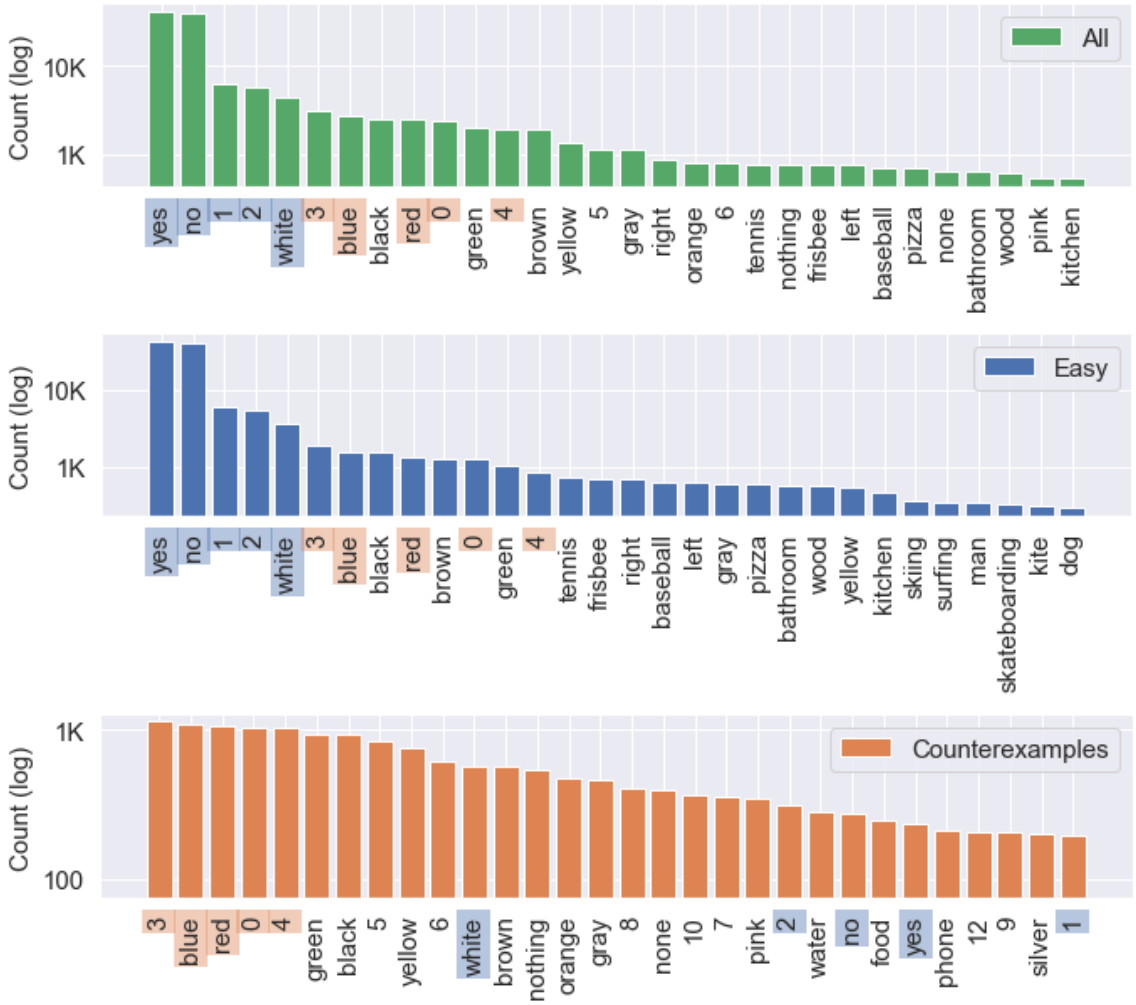


Figure 5.7. – Number of examples per answer (30 most frequent ones) in the complete validation set, our Counterexamples subset, and our Easy subset. Answers highlighted in blue and orange are the top 5 answers for the Easy and Counterexamples subsets respectively.

questions are going in the Easy subset, as they are correctly predicted by some rules. On the contrary, for the two other answer types, examples are more evenly distributed between the Easy and Counterexamples subsets.

5.4.2 Examples that are not matched by any rule

In Figure 5.10, we display some representative examples that are neither in the Easy subset nor the Counterexamples subset. These examples are not matched by any antecedent of our rules. Their input might be unusual. We do not add these examples to our Counterexamples subset, as they do not contradict the shortcuts

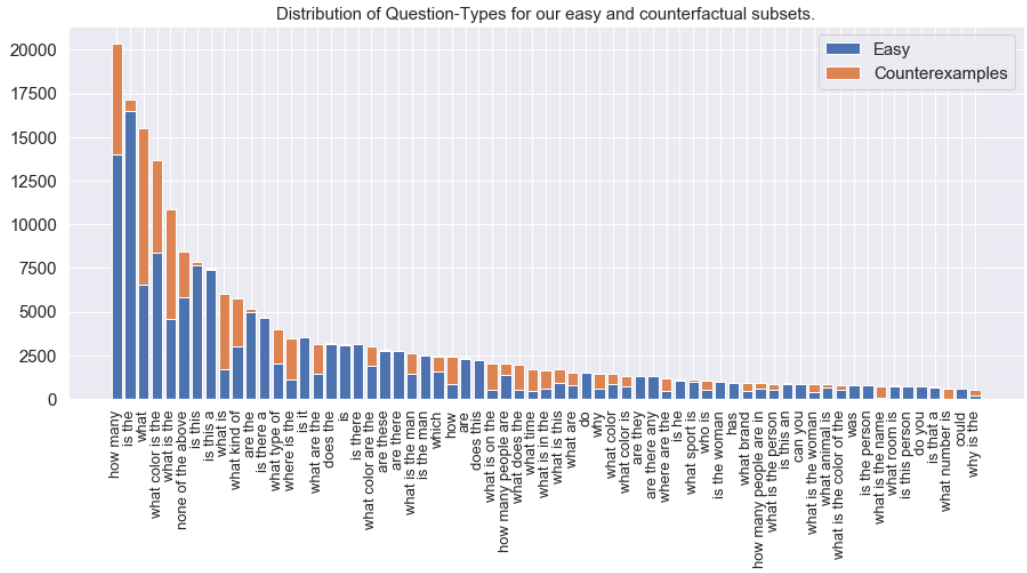


Figure 5.8. – Distribution of the number of examples per question type. Examples associated with our Counterexamples subset are matched by some shortcuts, but no shortcut leads to the correct answer. Examples associated with our Easy subset are matched by at least one shortcut that leads to the correct answer.

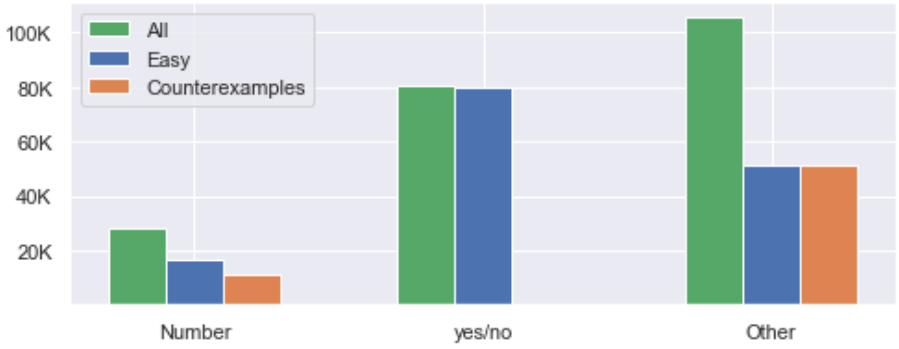


Figure 5.9. – Number of examples per answer type. "All" corresponds to all the examples from the VQA v2 validation set. Among them, examples associated with our "Counterexamples" subset are matched by some shortcuts, but none of these shortcuts leads to the correct answer. Inversely, examples associated with our Easy subset are matched by at least one shortcut that leads to the correct answer.

we found. We discard them entirely from our analysis. There consists of about 3K of examples.

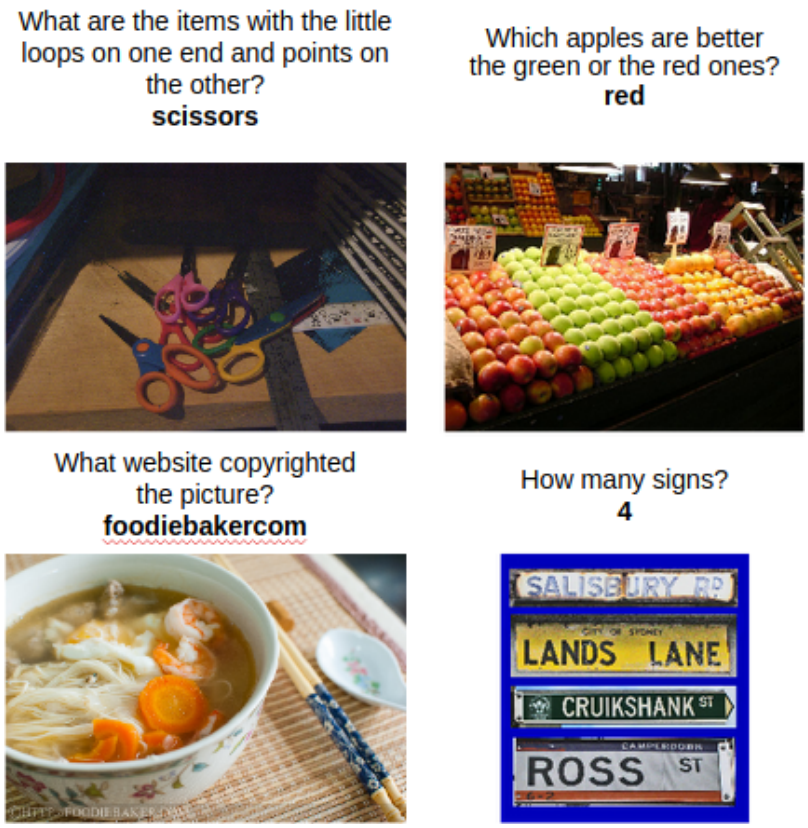


Figure 5.10. – Representative instances of image-question-answer examples that are not matched by any of our shortcuts. These examples have unusual questions, images or answers.

5.4.3 Main results

In Table 5.2, we report results of some baselines, common VQA models, and the latest bias-reduction methods following our VQA-CE evaluation protocol. Models that exploit shortcuts are expected to get a lower accuracy on the Counterexamples compared to their overall accuracy. All models have been trained on the VQA v2 training set and evaluated on the VQA v2 validation set. We detail them and discuss our findings in the next paragraphs.

Baselines The Question-Only and Image-Only baselines are deep models that only use one modality. They are often used to assess the amount of unimodal shortcuts that a deep model can capture. We report extreme drops in accuracy on our Counterexamples subset compared to the overall accuracy, with a loss of 32.53 points and 22.12 points respectively. This shows that most of the questions that are easily answerable by only using the question, or the image, are filtered out of our Counterexamples subset.

	Approaches	Overall	**Counterexamples**	**Easy**	VQA-CP v2
	Number of examples	*214,354*	*63,298*	*147,681*	
Baselines	Shortcuts	42.26	0.00	61.13	22.64
	Image-Only	23.70	1.58	33.58	19.31
	Question-Only	44.12	11.59	58.61	15.95
VQA models	SAN – *grid features*	55.61	26.64	68.45	24.96
	UpDown	63.52 (+0.00)	33.91 (+0.00)	76.69 (+0.00)	39.74
	BLOCK	63.89	32.91	77.65	38.69
	VilBERT – *pretrained*†	67.77	39.24	80.50	–
	UpDown is used as a base architecture for bias-reduction methods				
Bias-reduction methods	RUBi	61.88 (-1,64)	32.25 (-1,66)	75.03 (-1.66)	44.23
	LMH + RMFE	60.96 (-2.56)	33.14 (-0.77)	73.32 (-3.37)	54.55
	ESR	62.96 (-0.56)	33.26 (-0.65)	76.18 (-0.51)	48.50
	LMH	61.15 (-2.37)	34.26 (+0.35)	73.12 (-3.57)	52.05
	LfF	63.57 (+0.05)	34.27 (+0.36)	76.60 (-0.09)	39.49
	LMH+CSS	53.55 (-9.97)	34.36 (+0.45)	62.08 (-14.61)	58.95
	RandImg	63.34 (-0.18)	34.41 (+0.50)	76.21 (-0.48)	55.37

Table 5.2. – Results of our VQA-CE evaluation protocol. We report accuracies on VQA v2 full validation set and on our two subsets: **Counterexamples** and **Easy** examples. We re-implemented all models and bias-reduction methods. †VilBERT is pre-trained on Conceptual Caption and fine-tuned on VQA v2 training set. Scores in (green) and (red) are relative to UpDown. We also report accuracies on VQA-CP v2 (A. Agrawal et al. 2018a) which focus on question biases and come with a different training set and testing set. VilBERT was not evaluated for VQA-CP as it was pre-trained on balanced datasets. We evaluate SAN (Yang et al. 2016), UpDown (Anderson et al. 2018a), BLOCK (Ben-Younes et al. 2019b), VilBERT (J. Lu et al. 2019), RUBi (Cadene et al. 2019c), LMH + RMFE (Gat et al. 2020), ESR (Shrestha et al. 2020), LMH (C. Clark et al. 2019), LFF (Nam et al. 2020), LMH+CSS (L. Chen et al. 2020), RandImg (Teney et al. 2020b)

Aggregating shortcuts to create a classifier In order to evaluate our shortcuts as a whole, we aggregate them to build a VQA classifier. As shown in the preceding section, each training example is associated with shortcuts that can be used to find the correct answer. Among these correct shortcuts, we select the highest-confidence one for each example. This leaves us with 115,718 unique shortcuts. In order to predict an answer for an unseen example, we take the most predicted answer for all its matching shortcuts weighted by the confidence of the shortcuts. For the examples that are not matched by any shortcut, we output "yes", the most common answer. Our shortcut-based classifier reaches an overall accuracy of 42.26%, close to the 44.12% of the deep question-only baseline. Interestingly, both use different classes of shortcuts. Ours is mostly based on shallow multimodal

shortcuts, not just shortcuts from the question. Since we use the same shortcuts to create our subsets, the shortcut-based classifier reaches a score of 0% on the Counterexamples. On the VQA-CP testing set, our classifier reaches 22.44% accuracy. It highlights the difference with our counterexamples subset: VQA-CP does penalize some shortcuts, but there are still some that can be exploited.

VQA models learn shortcuts We compare different types of VQA models: SAN (Yang et al. 2016) represents the image as a grid of smaller patches and uses a stacked attention mechanism over these patches, instead, UpDown (Anderson et al. 2018a) represents the image as a set of objects detected with Faster-RCNN and uses a simpler attention mechanism over them, BLOCK (Ben-Younes et al. 2019b) also relies on the object representations but uses a more complex attention mechanism based on a bilinear fusion, VilBERT (J. Lu et al. 2019) also relies on the object representations but uses a transformer-based model that has been pretrained on the Conceptual Caption dataset (Sharma et al. 2018). First, they suffer from a loss of ~29 accuracy points on the counterexamples compared to their overall accuracy. This suggests that, despite their differences in modeling, they all exploit shortcuts. Note that comparable losses are reported on VQA-CP v2 (A. Agrawal et al. 2018a) which especially focuses on shortcuts based on question types. Second, our evaluation protocol can be used to compare two models that get similar overall accuracies: UpDown and BLOCK which gets +0.37 points over UpDown. We can explain that this gain is due to a superior accuracy on the Easy subset with +0.96 and report a loss of -1.00 points on the Counterexamples. These results suggest that the bilinear fusion of BLOCK better captures shortcuts. On the contrary, VilBERT gets a better accuracy on our both subsets. It might be explained by the advantages of pretraining on external data.

Bias-reduction methods do not work well on natural multimodal shortcuts Our evaluation protocol can also be used to assess the efficiency of common bias-reduction methods. We use publicly available codebases when available, or our own implementation. All methods have been developed on the VQA-CP v2 dataset. It introduces new training and evaluation splits of VQA v2 that follow different distributions conditioned on the question type. All the studied methods have been applied to UpDown and reached gains ranging from +5 to +20 accuracy points on the VQA-CP testing set. We evaluate them in the more realistic context of the original VQA v2 dataset. We show that their effect on our Counterexamples subset is very small. More specifically, some methods such as our previous work RUBi (Chapter 3), LMH+RMFE (Gat et al. 2020), and ESR (Shrestha et al. 2020) have a negative effect on all subsets. Some methods such as LMH (C. Clark et al. 2019) and LMH+CSS (L. Chen et al. 2020) slightly improve the accuracy on counterexamples but strongly decrease the accuracy on the Easy subset, and

consequently decrease the overall accuracy. As reported in Teney et al. 2020b, most of those methods rely on knowledge about the VQA-CP testing distribution (inversion of the answer distribution conditioned on the question), which no longer applies in our VQA v2 evaluation setting. Finally, we found two methods, Learning From Failure (LfF) (Nam et al. 2020) and RandImg (Teney et al. 2020b) that slightly improve the accuracy on the Counterexamples subset with gains of +0.36 and +0.50, while having a very small impact on the overall accuracy, even reaching small gains in the best case of LfF. It should be noted that LfF is more general than others since it was not designed for the VQA-CP context. Overall, all effects are much smaller compared to their effectiveness on VQA-CP. This suggests that those bias-reduction methods might exploit the distribution shift between VQA-CP training and evaluation splits. They are efficient in this setting but do not work as well to reduce *naturally-occurring* shortcuts in VQA.

Additional Experiments We share additional experiments in Appendix C. We explore two variants of our evaluation benchmark: First, in Section B.1, we show results using the ground-truth visual labels instead of the detected objects. Second, in Section B.2, we share results on the VQA v1 dataset. Both experiments show similar results as the ones presented in this Chapter.

5.5 Conclusion

As we explained in Chapter 2, most of the literature related to biases and shortcuts in VQA focuses on language priors. In this chapter, we explore the existence of multimodal shortcuts in the VQA v2 dataset. We introduce a method that discovers multimodal shortcuts in VQA datasets. It gives novel insights into the nature of shortcuts in VQA: there are many multimodal shortcuts that could be exploited by models to achieve high accuracy. We find many shortcuts that correlate with predictions of VQA models, suggesting that they are exploiting these superficial decision rules. Using those shortcuts, we introduce an evaluation benchmark to assess whether a given model exploits those: it consists in evaluating models on shortcut *counterexamples*. If a model does exploit shortcuts, it will perform poorly on those counterexamples. We find that most reference VQA models suffer from a significant loss of accuracy in this setting, whether they are simple task-specific models like SAN (Yang et al. 2016) or large pre-trained transformers like VilBERT (J. Lu et al. 2019). We also evaluate existing bias-reduction methods, including our previous work RUBi from Chapter 3. We find that even the most general purpose of these methods does not significantly reduce the use of multimodal shortcuts. This shows the need for new shortcut-reduction methods.

CHAPTER 6

RELIABILITY FOR VISUAL QUESTION ANSWERING

Chapter abstract

Despite significant improvements in Visual Question Answering (VQA), the ability of models to assess their own correctness remains under-explored. Recent work has shown that VQA models, out-of-the-box, can be very bad at abstaining from answering when they are wrong. The option to abstain, also called Selective Prediction, is highly relevant when deploying systems to users who must trust the system's output (e.g., VQA assistants for users with visual impairments). For such scenarios, abstention can be even more important as users may provide out-of-distribution (OOD) or adversarial, inputs that make incorrect answers more likely. In this work, we explore Selective VQA in both in-distribution (ID) and OOD scenarios, where models are presented with mixtures of ID and OOD examples. The goal is to maximize the number of questions answered while minimizing the risk of error on those questions. We propose a simple yet effective Learning from Your Peers (LYP) approach for training multimodal selection functions for making abstention decisions. Our approach uses predictions from models trained on distinct subsets of the training data as targets for optimizing a Selective VQA model. It does not require additional manual labels or held-out data and provides a signal for identifying examples that are easy/difficult to generalize to. In our extensive evaluations, we show this benefits several different models across different architectures and scales.

The work in this chapter has led to the publication of this conference paper:

- Corentin Dancette, Spencer Whitehead, Rishabh Maheshwary, Ramakrishna Vedantam, Stefan Scherer, Xinlei Chen, Matthieu Cord, and Marcus Rohrbach (2023). "Improving Selective VQA by learning from your peers". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Contents

6.1 Introduction

Recent successes of deep learning models for multimodal tasks have created the potential for many exciting real-world applications that require a large degree of reliability, such as assisting users with visual impairments (Gurari et al. 2018; Sidorov et al. 2020). However, with these novels, high-stakes applications come responsibilities towards the users, especially regarding the problem setups and the general approach to evaluating model performance. Moreover, as we saw in previous chapters, multimodal models are often not robust to out-of-distribution (OOD) inputs, in particular when learning spurious correlations. A prevalent cause of such incorrect predictions in real-world settings is distribution shifts (DeGrave et al. 2021; Mårtensson et al. 2020; Geirhos et al. 2020), where the test environment may differ from the training environment and models could encounter a wide variety of input examples at test time that may not satisfy the independent and identically distributed assumption often made by practitioners when developing models. This is especially true in open-ended tasks like Visual Question Answering (VQA) where models may receive adversarial, out-of-distribution (OOD) inputs that are difficult to answer correctly. Moreover, we

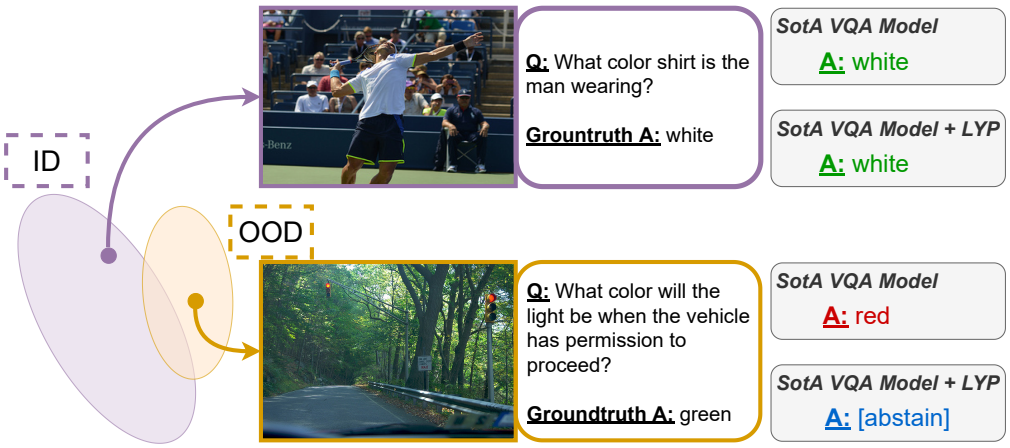


Figure 6.1. – VQA Models are able to answer straightforward ID questions, as in the top example where a state-of-the-art model (P. Wang et al. 2022) with and without our Learning from Your Peers (LYP) approach answers correctly. However, difficult OOD examples can arise, like the bottom example. With LYP, the model is able to abstain from answering to avoid outputting the incorrect answer, whereas the existing model is overconfident and outputs the answer anyways.

showed in previous Chapters 3,4 and 5 that VQA models were sensible to the learning of biases or shortcuts from their training data. Those models are likely to be overconfident and wrong on OOD examples that do not follow the training shortcuts. Another issue with OOD is examples that require knowledge or skills that the model did not learn during its training. For example, in Fig. 6.1, a model is asked a question that requires unknown background knowledge.

One particularly important consideration when developing models for real-world applications is *reliability*, i.e., the ability of the model to avoid making errors when facing uncertainty.

One way to approach reliability is to frame the problem as a selective prediction task (Chow 1957; El-Yaniv and Wiener 2010; Whitehead et al. 2022b). In selective prediction, models are able to either output an answer or abstain from answering (i.e., effectively saying "*I don't know*") based on the model's confidence/uncertainty in order to avoid making incorrect predictions. While the ability to answer open-ended questions has been a point of focus in VQA, having a model correctly answer all questions, ID and OOD, is likely unattainable (Geiger et al. 2019; Kamath et al. 2020). Therefore, framing this problem as a selective prediction task provides an avenue to handle such OOD examples more gracefully as the model can abstain from answering on many of these inputs, while still attempting to answer as many questions as possible. Doing this requires models to recognize OOD examples for abstention decisions (OOD detection)

and generalize to OOD examples (OOD generalization) in order to make predictions on examples that the model will get right. However, previous evaluations for selective prediction in VQA (Whitehead et al. 2022b) have been done on ID data, where the images and questions all come from the VQA v2 dataset (Goyal et al. 2017c). In NLP, there are some efforts on selective prediction with OOD examples (Kamath et al. 2020; Varshney et al. 2022), although they tend to not address practical considerations, such as assuming access to OOD data or threshold generalization. More broadly, selective prediction and OOD generalization have largely been studied as independent problems in the literature (Tran et al. 2022).

In this chapter, we explore selective prediction for VQA with distribution shifts, where we present models with mixtures of both ID and OOD examples, and measure the ability of different approaches to optimize answering as many questions as possible while maintaining a low risk of error (or high accuracy) on those questions. We perform experiments on VQA v2 (Goyal et al. 2017c) as our ID data and AdVQA (Sheng et al. 2021), an adversarially-collected VQA dataset, as our OOD data. We share more details about AdVQA in Section 6.2.

We evaluate several state-of-the-art approaches to this problem and find that existing models' softmax probabilities are generally poor confidence estimates for abstention decisions on OOD data, leading models to answer $<3\%$ of questions to achieve 1% risk of error in some settings. Further, we show that training a selection function (Whitehead et al. 2022b) improves performance ID and OOD, but integrating features from OOD detection methods as well as augmenting with known-OOD data (i.e., OOD data different from the unknown target distribution) does not improve beyond simply training this selection function on ID data. However, we observe that existing methods for training multimodal selection functions can require a held-out dataset in order to be effective.

Therefore, we propose a Learning from Your Peers (LYP) approach that removes the need for held-out data while also allowing both the VQA model and selection function to learn from the additional data that would have been held out. By using predictions on the training data from models that have not seen these examples, our approach provides a signal for which examples in the training data can be generalized to for a given model class, and which are too hard and should be abstained on. This allows us to train both the main VQA model and the selector on more data, boosting their performance.

In Section 6.2, we review prior work on selective prediction in VQA and distribution shifts in VQA. In Section 6.3, we propose an evaluation benchmark for selective prediction in VQA with distribution shifts, and evaluate reference methods. Then, in Section 6.4, we propose a method to use more efficiently the available data to train a better confidence model. Finally, in Section 6.5, we present our experiments and results.

6.2 Related Work

The main VQA architectures are discussed in Chapter 2, Section 2.2. In this chapter, we use the OFA model, which is described in Section 2.2.2.4. Next, we describe other works on out-of-distribution and selective prediction.

Out-of-distribution VQA We discussed in Chapter 2, Section 2.22 the VQA-CP benchmark, which proposes training and testing sets with a controlled distribution shift to control the learning of shortcuts.

AdVQA (Sheng et al. 2021) and A-VQA (L. Li et al. 2021) are recently introduced VQA benchmarks that comprise adversarial questions using human and model-in-the-loop procedures to generate adversarial examples. They propose evaluation datasets, which can be used with models trained on the original VQA datasets. We use AdVQA in this chapter, and show its construction process in Figure 6.2. It is collected in an adversarial manner, with humans in the loop: the human annotator is asked to write a question that might be hard for a model to answer, and then gets the output of a reference VQA model. If the model gets it correctly, the user can refine its question until the model gets it wrong. Other annotators then validate the examples or provide multiple ground-truth answers. This way, most questions are very hard to answer, even for state-of-the-art VQA models. This gives us a good benchmark to study the robustness of our models: they should abstain from answering most of those questions.

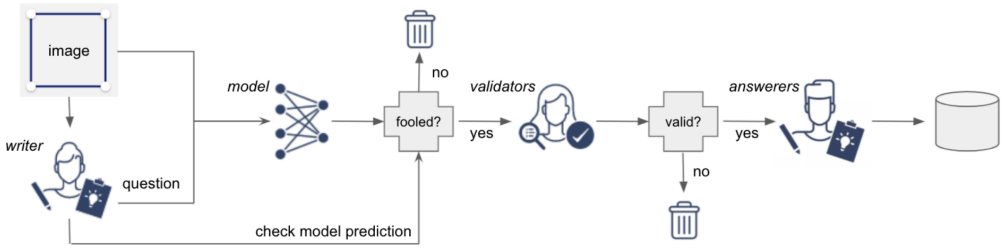


Figure 6.2. – Diagram of the data collection process for AdVQA. First, *writers* are asked to create questions that will fool a VQA model. *Validators* then double-check if the model was fooled. Finally, *answerers* provide ground-truth answers. Image from Sheng et al. 2021.

Other datasets require different abilities, such as TextVQA (Singh et al. 2019) which contains questions requiring reading text in the image, or OK-VQA (Marino et al. 2019) which requires outside knowledge. Recently, A. Agrawal et al. 2022 performed a cross-dataset evaluation to show that VQA models exhibit poor out-

of-distribution generalization. They used the GQA, VizWiz and Visual Genome datasets, presented in Chapter 2, Section 2.2.

Selective prediction & Reliability Recently, Whitehead et al. 2022b explored Selective Prediction for ID VQA. They experiment with different selectors on top of the base VQA model for improving their reliability on the VQA task. Varshney et al. 2022 investigates selective prediction approaches across several NLP tasks in ID, OOD, and adversarial settings. Specifically, they trained a selector (Multi-Layer Perceptron (MLP)) on top of the base model on a held-out split and used the selector's confidence scores to either answer or abstain from answering and improved risk, and coverage metrics compared to MaxProb. Corbière et al. 2019 studies failure prediction in Deep Neural Network (DNN) by training a confidence model on top to provide confidence measures for the model prediction.

OOD Selective Prediction. Geifman and El-Yaniv 2019 proposed SelectiveNet that incorporates a selection head on the top of the base model, which is optimized with a selective loss to reject samples that the model is uncertain about. Kamath et al. 2020 trains a calibrator on top of an existing NLP model to generalize to unknown OOD data at test time. Specifically, it trains the calibrator on a mixture of some held-out ID data and 'known' OOD data. The final model is used for the evaluation of the unknown OOD data at test time.

OOD Detection. Earlier works (Hendrycks and Gimpel 2017) relied on the maximum class probability (MaxProb) to detect OOD samples. Liang et al. 2018 proposed ODIN that combines temperature scaling and image perturbation to achieve better separation in softmax scores for OOD and IID images. Another line of work used distance-based scores (K. Lee et al. 2018) or energy scores (W. Liu et al. 2020; Z. Lin et al. 2021; Haoran Wang et al. 2021) to detect OOD. Haoqi Wang et al. 2022 introduced VIM that detects OOD samples by fusing the logits and feature information obtained from the model. Tian et al. 2014; Bergman et al. 2020; Sun et al. 2022a compute nearest-neighbor distances in the feature dimension to detect OOD data.

Image OOD Detection & reliability. Ovadia et al. 2019 investigates the effect of dataset distribution shift on accuracy and calibration. Lakshminarayanan et al. 2017 uses deep ensembles to quantify uncertainty estimates of classification models. Gawlikowski et al. 2021; Abdar et al. 2021 extensive review of uncertainty estimation methods in deep learning literature.

6.3 Selective VQA with ID and OOD Data

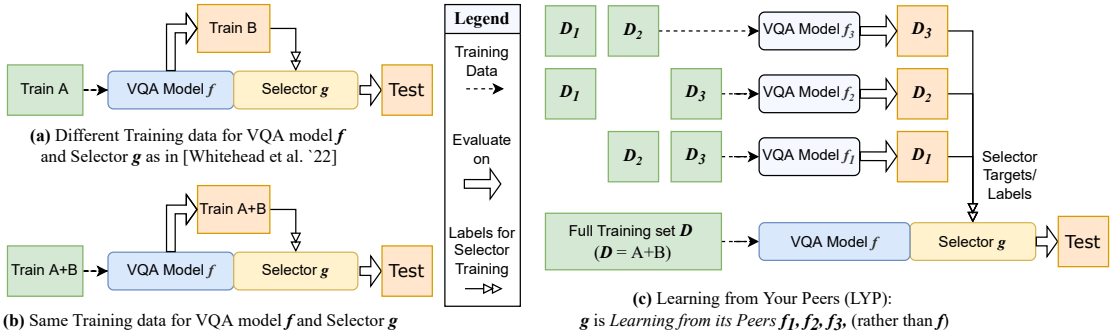
In this section, we discuss the problem formulation of Selective VQA(Section 6.3.1), how we evaluate in the ID (in-distribution) scenario (Section 6.3.2) and in the mixed ID+OOD (out-of-distribution) scenarios (Section 6.3.3).

6.3.1 Problem formulation

As explained in Section 2.2.2, the primary setting for VQA is to learn a function $f : \mathcal{Q}, \mathcal{V} \mapsto \mathcal{A}$ to predict an answer $a \in \mathcal{A}$ to a question $q \in \mathcal{Q}$ about a given image $v \in \mathcal{V}$. However, when exposing models to the real world they might encounter hard questions, OOD data points, or even adversarial questions by users and we cannot expect that models are able to answer all questions in these scenarios correctly. Therefore, we instead would like to identify inputs $x = (v, q) \in \mathcal{X}$ that models cannot correctly answer and abstain in those cases. This is the setting of Selective Prediction (El-Yaniv and Wiener 2010), which has also recently been studied for ID VQA (Whitehead et al. 2022b) and OOD text-only question answering (Kamath et al. 2020). In this chapter, we advocate for this selective prediction setting for ID and OOD scenarios. We closely follow the formalism introduced in Whitehead et al. 2022b for VQA. Specifically, the output space is extended to allow for an abstention option (denoted by $\emptyset$): $h : \mathcal{X} \mapsto \mathcal{A} \cup \{\emptyset\}$. Such a *Selective Model* h can be realized by decomposing h into two functions, a VQA model f and selection function $g : \mathcal{X} \mapsto \{0, 1\}$

$$h(x) = (f, g)(x) = \begin{cases} f(x) & \text{if } g(x) = 1, \\ \emptyset & \text{if } g(x) = 0. \end{cases} \tag{6.1}$$

For a given image-question pair $x = (v, q)$, the Selective VQA model h only predicts an answer from the VQA model f if the selection function g decides that an answer should be given. Otherwise, the Selective VQA model h abstains. The selection function g can be formulated based on a function $g' : \mathcal{X} \mapsto \mathbb{R}$ that scores the correctness of the model's prediction $f(x)$. Then, for a given γ, the model outputs the answer $f(x)$ if $g'(x) \geq \gamma$ and abstains otherwise. Ideally, g' should yield higher values if $f(x)$ is correct and lower if it is incorrect. However, as we show in the experiments this is a hard task.



(a) Different Training data for VQA model ***f*** and Selector ***g*** as in [Whitehead et al. `22]

(b) Same Training data for VQA model ***f*** and Selector ***g***

(c) Learning from Your Peers (LYP): ***g*** is *Learning from its Peers* f_1, f_2, f_3, (rather than ***f***)

Figure 6.3. – Comparison between Selector g training procedures. (a) shows the one in Whitehead et al. 2022b, (c) shows our LYP. See Section 6.4 for details.

6.3.2 Evaluation

Beyond accuracy, we evaluate using the metrics designed for models with abstention options following Whitehead et al. 2022b:

Risk and coverage. For a dataset D, model f, and a selection function g, *coverage* is the proportion of answered questions:

$$\mathcal{C}(g) = \frac{1}{|D|} \sum_{x \in D} g(x),$$

while *risk* is the average error on the covered subset

$$\mathcal{R}(f, g) = \frac{\sum_{(x_i, y_i) \in D} (1 - \mathrm{Acc}(f(x_i), y_i)) \cdot g(x_i)}{\mathcal{C}(g)},$$

where Acc is VQA accuracy (Antol et al. 2015b) and y_i is the corresponding ground truth answer. We measure the maximum coverage at a specific risk tolerance, denoted ($\mathcal{C}@\mathcal{R}$), by determining the largest consecutive subset of questions that can be answered with at most $\mathcal{R}$ risk. Further, we also compute the Area Under the Curve (AUC) for the risk-coverage curve (Kamath et al. 2020) for a summary of performance across different coverage levels. The AUC is computed by integrating the risk-coverage curve. Note that here, we aim for a low AUC, as for a given coverage, we prefer a model which minimizes the risk.

Effective Reliability Φ_c. This metric was introduced in Whitehead et al. 2022b to better compare methods on the test set for a threshold selected on a validation set. This is especially important for OOD, as thresholds for a certain risk level

don't generalize to the test scenario. Φ_c is a cost-based metric and jointly measures the reliability and effectiveness of selective models in a single metric. It assigns a cost of c to every wrong answer that the model outputs (i.e., does not abstain on):

$$\Phi_c(x) = \begin{cases} Acc(x) & \text{if } g(x) = 1 \text{ and } Acc(x) > 0, \\ -c & \text{if } g(x) = 1 \text{ and } Acc(x) = 0, \\ 0 & \text{if } g(x) = 0. \end{cases} \tag{6.2}$$

The total score is $\Phi_c = \frac{1}{|D|} \sum_{x \in D} \Phi_c(x)$, a mean over all samples x. To compute this metric, we set the threshold γ on a validation set to maximize Φ_c. Then, we use this threshold for abstention decisions on the test set.

6.3.3 Evaluating with Mixed ID+OOD data

As previously mentioned, we want to explore the setting where models encounter mixtures of ID and OOD data. More formally, we assume we are given $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ that are drawn from different distributions. In our setting, to simulate a setting closer to a real-world use case, the test data is sampled from a mixture of ID and OOD data. Similar to Kamath et al. 2020, we assume that our training data is drawn from P_{src} while our testing data is drawn from P_{tgt}, where $P_{\text{tgt}} = \alpha P_{\text{src}} + (1 - \alpha) P_{\text{unk}}$. Here, P_{unk} is an unknown distribution different from P_{src} from which we obtain our OOD examples. We obtain different mixtures of data by varying α and evaluate models across these using the metrics discussed in Section 6.3.2. Different from prior work in NLP (Kamath et al. 2020), we assume we *do not* have access to known OOD data for training, meaning all models must be trained and thresholds must be chosen on ID data. However, we do compare our method with this setting in our experiments.

6.4 LYP: Learning from Your Peers

Prior work has established training a selection function (or Selector) g to predict the correctness of the outputs of a model f (Geifman and El-Yaniv 2019; Kamath et al. 2020; Whitehead et al. 2022b) as a method for selective prediction. As in Whitehead et al. 2022b, our Selector g learns to predict the VQA Accuracy of f. One option is to train f on one part of the training data (Train A) and g on a different, typically smaller, part (Train B), as shown in Fig. 6.3(a). Having separate training data for g can be crucial: if f overfits the training data, then training g on that same data will lead g to a solution that doesn't generalize well

(e.g., always answering). We show some of these drawbacks in our experiments with observations similar to findings on stacked generalization (Wolpert 1992). However, withholding data from training f could reduce the overall performance of f, as it does not allow f to learn from this data. Likewise, g is unable to learn from the training data for f. This motivates training both f and g on the same data, e.g., as done in Geifman and El-Yaniv 2019 (shown in Fig. 6.3(b)).

We propose a simple yet effective approach, called Learning from Your Peers (LYP), for training g that allows both f and g to utilize all the available training data. Inspired by work on collective outliers (Karamcheti et al. 2021) and improving worst group performance (E. Z. Liu et al. 2021), our approach aims to identify examples in the training data that are difficult to generalize to, for a given architecture and learning procedure. In particular, we want to provide more signal to g about which examples in the training data may not be generalizable and likely should be abstained on, despite the VQA model's potential ability to fit these examples during training.

Shown in Fig. 6.3(c), we first partition our full training set $\mathcal{D}$ into N disjoint subsets ($\mathcal{D}$ = Train A + Train B). For our VQA setting, we create our partitions by ensuring no images overlap between them. Next, we train N different models on combinations of the subsets in leave-one-out manner: we create a training set $\mathcal{D}_n^* = \mathcal{D} \setminus \mathcal{D}_n$ and train a VQA model f_n on $\mathcal{D}_n^*$. Once we have trained f_n, we use it to make predictions on $\mathcal{D}_n$, which it has not seen during its training. We use the ground truth annotations for $\mathcal{D}_n$ to obtain VQA accuracy for each prediction, which we treat as a label for the correctness of each prediction. After performing this operation for $n = 1, ..., N$, we can union the partitions to obtain an updated training set $\mathcal{D}^{\text{sel}}$ that additionally has correctness labels for each example $(x_i^{(n)}, y_i^{(n)}, f_n(x_i^{(n)}), \xi_i^{(n)})$ for $(x_i^{(n)}, y_i^{(n)}) \in \mathcal{D}_n$, where $\xi_i^{(n)} = \text{Acc}(f_n(x_i^{(n)}), y_i^{(n)})$.

We train our VQA model f on all of $\mathcal{D}$ and then, with the obtained correctness labels, we train our Selector g on top of f using the full $\mathcal{D}^{\text{sel}}$ dataset. For training g, we optimize it using a binary cross-entropy objective with the correctness labels as the target. Note the source of our targets is not the model f itself but, rather, the subset models $\{f_n\}_{n=1}^N$ (i.e., the *peers* of f). The idea behind this is that if a model trained on the remainder of the training data $\mathcal{D}_n^*$ cannot generalize to an example in $\mathcal{D}_n$, then that may be a challenging example that g should choose to abstain on as the model f is unlikely to generalize reliably to such an example at test time, even if it has fit it during training. Essentially, these correctness labels may provide a signal for which examples are difficult and might require abstention *more generally* rather than concerning a specific model. Moreover, as we show in our experiments, this allows both f and g to learn from the entire training data, giving a boost in both overall accuracy as well as abstention performance.

6.5 Experiments

In this section, we evaluate our approach LYP in an in-distribution scenario and a realistic mixed setting, where we encounter in-distribution and out-of-distribution examples at test time.

6.5.1 Experimental Setups

Models We operate on two different VQA architectures: **CLIP-ViL** (Shen et al. 2021) and **OFA** (P. Wang et al. 2022). CLIP-ViL is an ensemble of MCAN (Yu et al. 2019) and MoVie (Nguyen et al. 2021) with a CLIP (Radford et al. 2021) image encoder, and **OFA** is a recent transformer encoder-decoder model that performs multiple tasks and achieves state-of-the-art performance on VQA v2. CLIP-ViL represents a strong VQA model that treats VQA as a classification task over a large set of answers (Teney et al. 2018), while OFA represents a large-scale pre-trained model that treats VQA as a generative task [1]. For OFA, we explore 2 different sizes of the model: OFA-Base and OFA-Large.

Selection Functions. We explore **MaxProb** (Geifman and El-Yaniv 2017; C. Guo et al. 2017; Hendrycks and Gimpel 2017; Kamath et al. 2020; Whitehead et al. 2022b) as a baseline as it is a natural comparison to the VQA model out-of-the-box since the confidence scores are simply the output probabilities of the model. We also evaluate the **Selector** developed by Whitehead et al. 2022b as it attains the strongest performance for selective VQA. Selector is a MLP that takes in a combination of image, question, multimodal and answer representations from the VQA model in order to predict a confidence score.

For CLIP-ViL, We use the implementations for MaxProb and Selector provided by Whitehead et al. 2022b. [2]. We follow the given hyperparameters and training procedure. We refer readers to Whitehead et al. 2022b for details.

OFA first processes the image using a convolutional network (He et al. 2016) to obtain a set of visual representations $\tilde{V}$. Likewise, the question is tokenized and converted to a sequence of question token embeddings $\tilde{Q}$. Then, the visual features are flattened into a sequence and concatenated with the question token embedding sequence. This entire sequence is given as input to an encoder-decoder

1. While OFA is a generative model, it uses a trie-based decoding method for VQA that restricts the generated sequences to an answer vocabulary, as opposed to open-ended generation (P. Wang et al. 2022).
2. https://github.com/facebookresearch/reliable_vqa

transformer model (Vaswani et al. 2017) to predict the answers. The encoder produces multimodal representations of the image tokens $\{v_i\}_{i=1}^{|\tilde{V}|}$ and question tokens $\{q_j\}_{j=1}^{|\tilde{Q}|}$. The encoded tokens are used as input to cross-attention layers in the transformer decoder at each decoding step. The decoder generates output token representations $\{o_l\}_{l=1}^{L}$ for an answer of L tokens. These token representations can be fed to a linear layer to give the output logits over the token vocabulary. We use beam search to decode the answers.

We fine-tune OFA from the pre-trained checkpoints provided by the authors of P. Wang et al. 2022.[3] We follow the hyperparameters from the original paper for fine-tuning. In the following, we detail the setup for the selection functions:

MaxProb. Since OFA is a sequence-to-sequence model that generates answers token-by-token, for the MaxProb baseline, we use the joint probability of each answer token as the confidence value, similar to common decoding algorithms like beam search.

Selector. We largely replicate the same Selector architecture and training as Whitehead et al. 2022b (i.e., two-layer MLP), but with some slight differences. We remove the non-linear projection (or one-layer MLP) for each input representation. We also use slightly different input representations: First, we max-pool the encoder image (v_i) and question (q_i) token representations to obtain a single representation for each set of representations. Then, we extract the probability of the predicted answer p, which is the joint probability of each answer token. Finally, we extract the first output token embedding o_1 that is used to predict the first answer token. We concatenate these representations and feed this as input to the Selector.

Training Selector with OFA. We report the training parameters in Table 6.1. We first train the VQA model as discussed above, freeze the VQA model, and then train Selector on top of this frozen model. We train for a maximum number of 32 epochs and perform early-stopping on the Val split (Table 6.2) using the AUC metric. We keep the dropout in the main model during the selector training, as we found this improved performance of the selector.

3. https://github.com/OFA-Sys/OFA

	OFA-Base	OFA-Large
Batch Size	256	512
Learning Rate	1e-4	4e-4
LR warmup	no	no
LR-decay (linear)	-1e-10/step	-1e-10/step
Optimizer	Adam	Adam
Optimizer Beta	(0.9,0.999)	(0.9,0.999)
Gradient clipping	1.0	1.0
Selector Dropout	0.1	0.1
Main model dropout	0.1	0.1
Image size	480	480

Table 6.1. – Hyperparameters for Selector Training on top of OFA

6.5.2 Dataset Splits

Data. In our experiments, we require both ID and OOD data that have annotations available for evaluation. Therefore, we utilize the splits of the VQA v2 dataset (Goyal et al. 2017c) made available by (Whitehead et al. 2022b) as our ID data. The entire VQA v2 train set (call it split **A**) is used for training VQA models (f). Meanwhile, the VQA v2 validation set is split into 3 parts: 86k examples (40%) for training selection functions g (call it split **B**); 22k examples (10%) for validating models; 106k examples (50%) as a test split for evaluating full selective models $h = (f, g)$. LYP does not require different sets for training f and g, so we train them both with the combination of A and B (**A+B**). For OOD data, we use AdVQA (Sheng et al. 2021), which is an adversarial dataset constructed by asking human annotators to create questions that are difficult to answer for existing VQA models trained on VQA v2. The images in AdVQA and VQA v2 overlap with each other, so we only use images from AdVQA that appear in our test split. While AdVQA is not OOD in terms of the images, one can still consider this as adversarial, OOD since the questions are designed to fall outside the training distribution of VQA v2. This is similar to other OOD VQA datasets like VQA-CP (A. Agrawal et al. 2018a), VQA-CE (Dancette et al. 2021b), or other VQA generalization benchmarks (A. Agrawal et al. 2022; Whitehead et al. 2021). However, for our setting, we create different mixtures of VQA v2 and AdVQA to serve as our evaluation data, where each mixture contains a different percentage of ID/OOD data.

In-Distribution Splits We follow Whitehead et al. 2022b and use the splits provided in the official implementation. We detail the splits again in Table 6.2. No images (or question-answer annotations) are shared between splits.

Split	Usage	Source	%src	#I	#Q
Train A	Train f,g	VQA v2 train	100%	82,783	443,757
Train B	Train f, g	VQA v2 val	40%	16,202	86,138
Val	Validate f, g	VQA v2 val	10%	4,050	21,878
Test	Test $h = (f, g)$	VQA v2 val	50%	20,252	106,338

Table 6.2. – Size of the splits of VQA v2 from Whitehead et al. 2022b. Note, the "Usage" is the setting for the full model (A+B). Some models are trained on subsets (e.g., just A) as specified in the corresponding tables.

ID/OOD Mixtures We use AdVQA as our source of OOD data. As discussed, AdVQA is an adversarial dataset where human annotators intentionally ask questions that state-of-the-art models trained on VQA v2 answer incorrectly. The images in AdVQA come from T.-Y. Lin et al. 2014c, as do VQA v2. However, we consider this as OOD since the questions are adversarial in nature and contain distribution shifts meant to induce errors for models trained on VQA v2.

In our work, we create mixtures of ID/OOD examples for our evaluations. To form our mixtures, we first discard all AdVQA images that overlap with the A+B train set. This leaves 5,008 AdVQA examples. For each setting, we randomly sample examples from the ID Test split (Table 6.2) to create the desired OOD proportion: 45K for 10% OOD, 10K for 33% OOd, 5K for 50% OOD and 2.5K for 66% OOD.

6.5.3 In-Distribution Experiments

We first experiment with only in-distribution data to compare with prior work. Discussed in Section 6.3.1, we evaluate using maximum coverage at different risk levels ($\mathcal{C}@\mathcal{R}$), AUC for the risk-coverage curve, and effective reliability at different costs (Φ_c). We also present accuracy to give an idea of the question-answering performance of each model.

ID performance consistently improves with LYP. Table 6.3 shows that across all model architectures, the top scores are achieved by utilizing LYP. For instance, we see improvements in $\mathcal{C}@1\%$ over both MaxProb (A+B) and Selector (B) with OFA-Large of 16.31% and 2.06%, respectively. Likewise, Φ_{100} increases with LYP by 12.49 and 1.2 over MaxProb (A+B) and Selector (B), respectively, for OFA-Large. Moreover, the improvements are sustained at higher risk levels and lower costs. These same observations hold across each model we experiment with on ID data.

VQA Model f		Selection function g			Acc ↑	$\mathcal{C}@\mathcal{R}$ in % ↑			AUC ↓	Φ_1	Φ_{10}	Φ_{100}
Name	Training	Name	Training	Targets		$\mathcal{C}@1\%$	$\mathcal{C}@5\%$	$\mathcal{C}@10\%$				
CLIP-ViL	A	MaxProb	-	-	69.98	4.97	33.79	53.62	10.92	54.67	21.40	1.32
		Selector	B	Self	69.98	15.79	37.79	55.65	10.21	55.44	25.82	8.74
	A+B	MaxProb	-	-	70.72	5.54	34.84	55.04	10.49	55.93	22.81	2.59
		Selector	A+B	Self	70.72	6.45	34.26	56.07	10.48	56.07	22.99	2.39
		Selector	A+B	LYP	70.72	**18.40**	**38.65**	**57.40**	**9.76**	**56.53**	**26.45**	**9.74**
OFA Base	A	MaxProb	-	-	74.87	3.45	45.60	66.61	7.99	62.52	30.57	6.81
		Selector	B	Self	74.87	24.58	49.35	68.61	7.39	62.80	34.54	13.49
	A+B	MaxProb	-	-	75.18	14.88	46.15	67.51	7.79	63.04	30.13	7.29
		Selector	A+B	Self	75.18	26.64	50.80	69.56	7.10	63.66	34.92	12.92
		Selector	A+B	LYP	75.18	**27.71**	**51.64**	**70.20**	**6.98**	**63.88**	**36.29**	**16.30**
OFA Large	A	MaxProb	-	-	77.53	20.57	53.99	75.18	6.42	66.68	36.12	8.21
		Selector	B	Self	77.53	30.86	58.05	76.65	5.81	67.34	41.43	17.58
	A+B	MaxProb	-	-	77.79	16.31	53.83	75.27	6.43	66.96	36.06	6.29
		Selector	A+B	Self	77.79	31.47	58.80	77.14	5.69	67.82	41.43	16.08
		Selector	A+B	LYP	77.79	**32.92**	**59.43**	**77.52**	**5.60**	**68.02**	**42.83**	**18.78**

Table 6.3. – Risk-coverage metrics and effective reliability on ID data (i.e., VQA v2 test split (Whitehead et al. 2022b)). Scores for OFA-Large with Selector are averaged over 3 trials.

Lastly, we see that all Selector models improve beyond all MaxProb models on every metric for this ID data, as shown in Whitehead et al. 2022b.

LYP helps VQA models and Selector learn from the same data. We observe that training Selector and CLIP-ViL on the same data (A+B) performs poorly, achieving $\mathcal{C}@\mathcal{R}$ and Φ_c similar to its MaxProb counterpart. Conversely, the OFA models and Selector can be trained on the same data and reap the benefits of training on more data. We hypothesize this is due to the overfitting issue: CLIP-ViL has a training accuracy of 87.40% whereas, e.g., OFA-Base has a training accuracy of 82.92% while also having higher accuracy on the test split. However, we see that when using LYP, CLIP-ViL and Selector can be trained on the same data and improve beyond the model of Whitehead et al. 2022b by, e.g., 2.61% $\mathcal{C}@1\%$. Furthermore, although training on the same data can be done for the OFA models and Selector, it does not perform quite as well as when LYP is used. For example, with OFA-Base, training both the VQA model and Selector on A+B has $\mathcal{C}@1\%$ of 26.64% compared to 24.58% when the VQA model is trained on A and Selector is trained on B. Meanwhile, using LYP with OFA-Base attains 27.71% $\mathcal{C}@1\%$. These results suggest that LYP can help ID performance regardless of overfitting on the training data.

VQA Model f		Selection function g			Acc ↑	$\mathcal{C}@\mathcal{R}$ in % ↑			AUC ↓	Φ_1	Φ_{10}	Φ_{100}
Name	Training set	Name	Training Set	Targets		1%	5%	10%				
CLIP-ViL	A	MaxProb	-	-	66.35	0.00	24.16	43.53	13.55	49.12	14.39	-4.64
		Selector	B	Self	66.35	12.69	31.12	46.96	12.47	50.36	20.15	5.22
	A+B	MaxProb	-	-	67.12	2.60	26.13	45.25	12.97	50.49	16.59	-0.93
		Selector	A+B	Self	67.12	2.97	26.70	46.19	12.80	50.89	18.19	-0.65
		Selector	A+B	LYP	67.12	**15.22**	**32.58**	**49.18**	**11.90**	**51.43**	**22.09**	**7.12**
OFA Base	A	MaxProb	-	-	71.59	0.01	36.07	56.49	10.10	57.49	23.15	-0.34
		Selector	B	Self	71.59	19.00	42.61	59.82	9.15	58.13	28.09	4.79
	A+B	MaxProb	-	-	72.00	1.74	37.02	57.57	9.78	58.11	22.09	0.53
		Selector	A+B	Self	72.00	19.72	42.70	60.84	8.90	58.90	28.05	2.88
		Selector	A+B	LYP	72.00	**21.58**	**44.09**	**61.69**	**8.74**	**59.11**	**28.79**	**10.88**
OFA Large	A	MaxProb	-	-	74.56	4.76	44.53	66.06	8.21	61.90	28.20	0.21
		Selector	B	Self	74.56	23.53	50.17	68.76	7.33	62.96	34.43	9.88
	A+B	MaxProb	-	-	74.79	1.30	43.70	65.95	8.26	62.24	27.09	-2.46
		Selector	A+B	Self	74.79	22.68	50.27	69.27	7.32	63.03	33.50	4.92
		Selector	A+B	LYP	74.79	**25.38**	**51.07**	**69.74**	**7.17**	**63.41**	**34.85**	**10.34**

Table 6.4. – Mixed ID/OOD scenario, composed of 90% VQA v2 and 10% AdVQA examples.

6.5.4 OOD Evaluation

For our OOD evaluations, we build mixed datasets comprised of 10%, 33%, 50%, and 66% OOD examples. All mixtures contain 5K examples from AdVQA as OOD examples, and the rest are randomly sampled from our ID test split. We report the results for all models on the 10% OOD dataset in Table 6.4. Results for other splits can be found in Appendix B.

MaxProb can be overconfident on OOD data. Across all models, we see that MaxProb has <3% $\mathcal{C}@1\%$ and its Φ_c scores become negative. These results suggest that MaxProb may be overconfident on OOD examples, on which the model is more likely to be incorrect. While improving the VQA accuracy of the model improves MaxProb performance, training a Selector remains the most effective approach and consistently.

LYP maintains improvements over other methods in the 90%/10% setting. Similar to the pure ID setting, LYP continues to outperform other methods on the 90%/10% mixed setting. Although, we see decreases in all metrics across each of the different methods, demonstrating the challenge of this task even with just 10% OOD data.

The more OOD data, the more challenging. We show the AUC for our models on various mixtures of ID/OOD data in Fig. 6.4. Overall, our LYP method consis-

tently improves AUC over the Selector baseline from Whitehead et al. 2022b, for the three models (note lower is better for AUC).

We display two other metrics, $\mathcal{C}@5\%$ and Φ_{100} in Figs. 6.5a and 6.5b. For both plots, we show each of the three models with a Selector trained with LYP versus the Selector from Whitehead et al. 2022b. We see in Fig. 6.5a that the improvements are slightly less consistent: For CLIP-ViL and OFA-Base, LYP consistently improves the scores over a baseline on all mixture levels. For OFA-Large, we see that LYP only improves the results on in-distribution and low-OOD levels. On higher-OOD levels, the baseline performs slightly better than LYP. This might be explainable by the fact that OFA-Large is trained on a much larger dataset, and thus is more robust out-of-the-box OOD data, thus making LYP less effective. We make a similar observation for the Φ_{100} metric in Fig. 6.5b: LYP outperforms the baseline for CLIP-ViL on all mixed settings but does not improve consistently performance for OFA-Base and OFA-Large on high-OOD levels. This shows that more work is needed to help generalize to such OOD data.

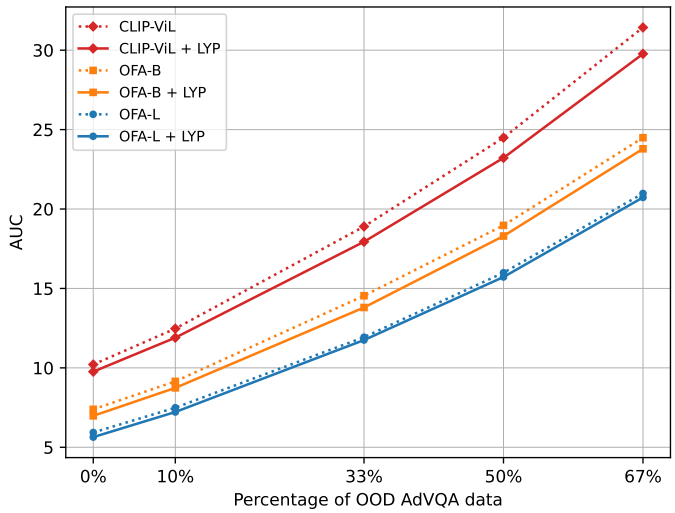
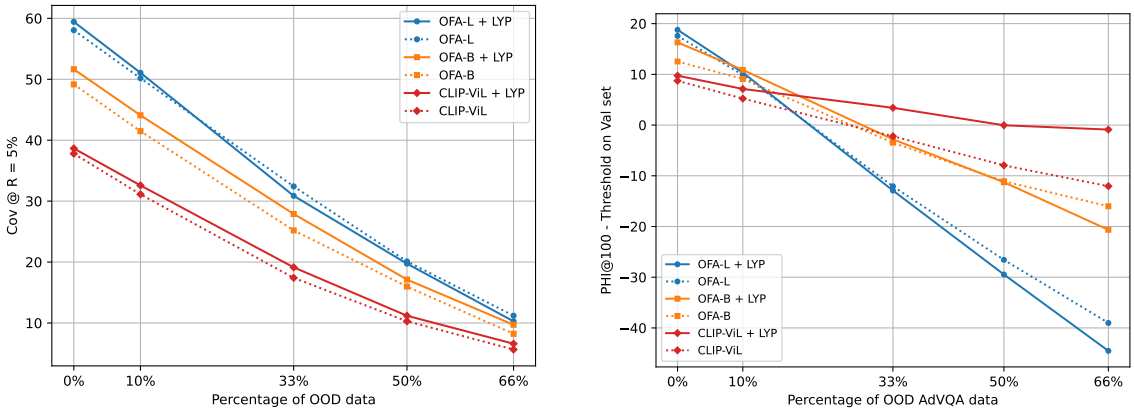


Figure 6.4. – AUC for various mixtures of VQA v2 + AdVQA. Note: lower is better for AUC

Additional OOD results In Tables C.4 to C.6 of the Appendix B, we present the results for our experiments on the remaining OOD mixtures of VQA v2 and AdVQA (Sheng et al. 2021). While we on average see that the Selector models and Selector + LYP perform better than the corresponding baselines models out-of-the-box (MaxProb), all models degrade dramatically if there is a high percentage of OOD data in the test mixture, especially for low risk ($\mathcal{C}@1\%$) or high cost of error (Φ_{100}). Especially if we look at the realistic scenario where the threshold is chosen on the validation set and used at test time (as for Φ_{100}), we notice that the scores



(a) $\mathcal{C}@5\%$ (b) Φ_{100}

Figure 6.5. – $\mathcal{C}@5\%$ and Φ_{100} for various mixtures of VQA v2 + AdVQA. OFA-L stands for Large, OFA-B for Base. All models with Selector.

of all methods drop below 0 with 33.3% or more OOD data. This can be seen in the last column of Tables C.4 to C.6 or in Figure 6.5b. These results demonstrate that these thresholds can be overconfident on OOD examples, which leads to poor abstention decisions such that the cost of the models' incorrect outputs outweighs the gains of the correct ones. Future work is needed to improve such OOD generalization and recognize samples that cannot reliably be answered in this challenging setup, which this work provides a new and interesting test setup for.

6.5.5 Threshold Generalization

In this section, we investigate threshold generalization. All previous tables reported numbers on "maximum coverage" at risk $\mathcal{R}$. This metric is irrespective of the threshold chosen as it solves for the coverage that satisfies a given risk level. In a real-world setting, the threshold would need to be fixed once using a validation set and then used at test time. We already evaluate this setting of evaluating the optimal threshold on the validation set for the cost-based metric Φ_c in Section 6.5.4. In contrast to Φ_c, which allows comparing a single number, for risk and coverage, choosing a threshold on a validation set leads to changes in coverage *and* risk, making it difficult to compare two methods. Still, in this section, we evaluate how the threshold generalizes to ID and OOD settings.

Our method improves risk generalization over out-of-the-box MaxProb. In Fig. 6.6, we show the test risk on various ID/OOD mixtures with a threshold set on the ID validation split of VQA v2 for a target risk of 1%. We see that LYP (solid

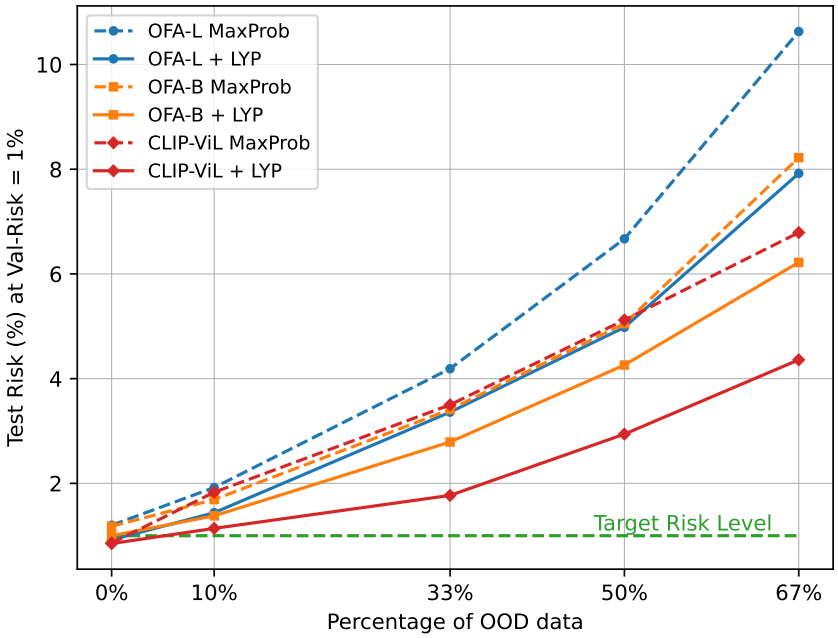


Figure 6.6. – Risk at various percentages of OOD when the threshold is optimized on the validation set for maximum coverage, with a target risk level of 1%.

line) consistently improves the generalization of risk over the MaxProb baseline: the curves corresponding to LYP are closer to the 1% target risk level compared to MaxProb.

Risk generalization is limited for OOD data. While we observe reasonable good risk generalization for ID, the generalization is really limited for larger percentages of OOD data.

CLIP-ViL is the best model for risk generalization. We see that all variants of CLIP-ViL outperform their corresponding methods on OFA-B and OFA-L. Note that the associated coverages are lower for the same risk level, thus CLIP-ViL is not the best method overall. This is somewhat surprising, as Kadavath et al. 2022b found that larger language models were better calibrated on NLP tasks.

Full results are available in Table C.7 and C.8 for our in-distribution testing set and our mixed setting with 90% of VQA v2 and 10% of AdVQA examples.

6.5.6 Further analysis

How many models/splits are needed for LYP? We run an ablation on OFA-Base to show the impact of the number splits N we make of the training dataset $\mathcal{D}$. This impacts the total training time, as we need to train a model for each split $\mathcal{D}_n^*$. We see in Table 6.5 that the number of models has a very small impact on the final results. This suggests that the overhead in training time can be reduced significantly while maintaining strong performance.

N	$\mathcal{C}$@1%	$\mathcal{C}$@5%	$\mathcal{C}$@10%	AUC
10	**27.71**	**51.64**	**70.20**	**6.98**
2	27.64	51.24	70.12	7.01

Table 6.5. – Varying the number of splits N for LYP. Results are reported on our ID VQA v2 test split for OFA Base, trained on A+B, with a selector trained on A+B.

Effect of training data size. We show in Table 6.6 that the amount of data used for the Selector training is an important factor for its performance. Note that the Train B set has 86K examples, which is ∼15% of the full Train A+B. The additional data, labeled with LYP, helps Selector generalize better to test examples.

% of A+B	$\mathcal{C}$@1%	$\mathcal{C}$@5%	$\mathcal{C}$@10%	AUC
100	**27.71**	**51.64**	**70.20**	**6.98**
75	27.48	51.11	70.26	7.01
50	26.84	51.04	70.04	7.06
25	26.03	50.15	69.65	7.16
10	23.30	47.97	68.03	7.44
5	22.62	46.10	66.10	7.71

Table 6.6. – Varying the amount of training data for the Selector. Results are reported on the ID VQA v2 test split. The model is OFA Base, trained on A+B, with a selector trained on a subset of A+B. Scores are labeled by 10 models following our LYP method.

Impact of scaling on Selective prediction. We show in Table 6.7 the results for three OFA variants: OFA Medium, OFA Base, and OFA Large. Those models have respectively 93M, 180M, and 470M parameters. We see that larger models, in addition to having a much higher accuracy on the testing set, have much better reliability when paired with a trained selector.

Model	Method	Acc.	$\mathcal{C}$@1%	$\mathcal{C}$@5%	$\mathcal{C}$@10%	AUC
Medium	MaxProb	71.30	5.08	37.56	56.85	9.95
Base	MaxProb	74.70	15.82	45.67	66.11	7.96
Large	MaxProb	77.79	6.12	53.57	75.04	6.49
Medium	Ours	71.30	19.69	41.28	59.60	9.17
Base	Ours	74.70	27.71	51.64	70.20	6.98
Large	Ours	77.79	32.54	58.98	77.34	5.64

Table 6.7. – Scaling results for three variants of OFA: Medium (93M parameters) Base (180M parameters) and Large (470M parameters) on our VQA v2 test subset.

6.5.7 Evaluation of other baseline methods

Additionally, we evaluate the performance of multiple other baseline methods to improve reliability in the out-of-distribution (OOD) setting. We report those results in Appendix C. First, inspired by Fisch et al. 2022, we train Selector with out-of-distribution detection scores computed with KNN (Sun et al. 2022b) or SSD (Sehwag et al. 2021) as added features. We share the results in Section C.2. Second, as discussed in Section 6.3.3, we also try training Selector on the B set, along with some known OOD datasets similar to Kamath et al. 2020. We show these results in Section C.3.

We show that both of those baselines, which are effective in image classification tasks, fail here to improve the reliability of VQA models.

6.6 Qualitative examples

Figs. 6.7 to 6.9 show qualitative results comparing the OFA-Large + LYP and OFA-Large + MaxProb, on the AdVQA dataset. In all cases, the OFA-Large model f is trained on A+B. For all examples, the abstention threshold is set on the in-distribution validation set to get maximum coverage at 5% risk. Fig. 6.7 shows examples where the VQA model (OFA-Large) is incorrect. Thus, the correct behavior is to abstain. But the MaxProb model does not abstain using the provided threshold, instead, it answers incorrectly. On the contrary, our model OFA-L + LYP abstains. Fig. 6.8 shows examples where the OFA-L model is correct: the best behavior is to answer. The MaxProb model abstains, while our method answers correctly. Fig. 6.9 shows two kinds of failure cases of our models. In the first line, OFA-L + LYP incorrectly abstains, as the VQA model was correct. In the second line, our model incorrectly answer instead of abstaining, as the answer provided by the model was incorrect.

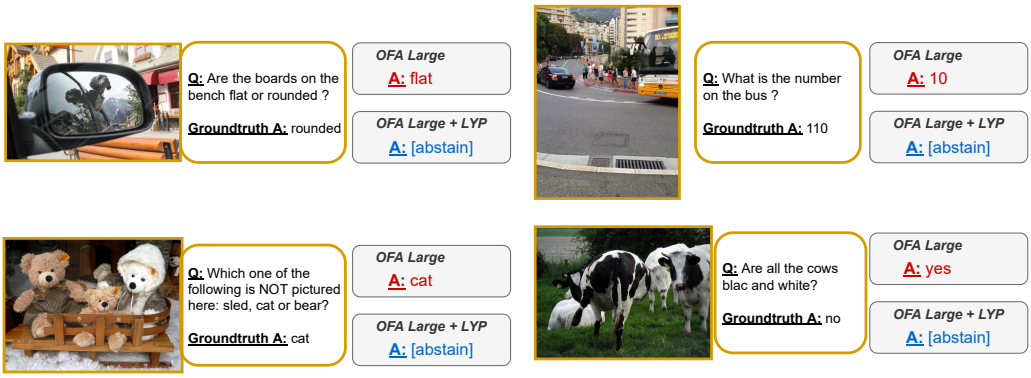


Figure 6.7. – Qualitative examples for OFA-Large on AdVQA: on those examples, the baseline (MaxProb) answers incorrectly the answer, and our model with LYP abstains.

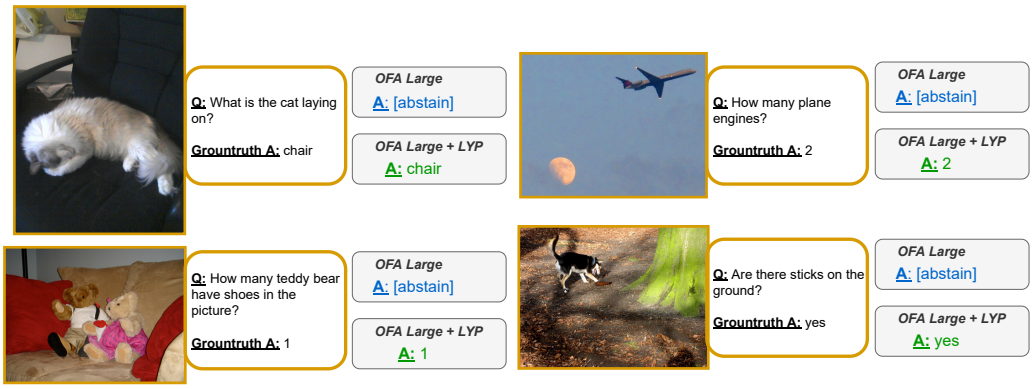


Figure 6.8. – Qualitative examples on AdVQA: on those examples, the baseline model abstains but had predicted the correct answer. OFA-L + LYP does not abstain.

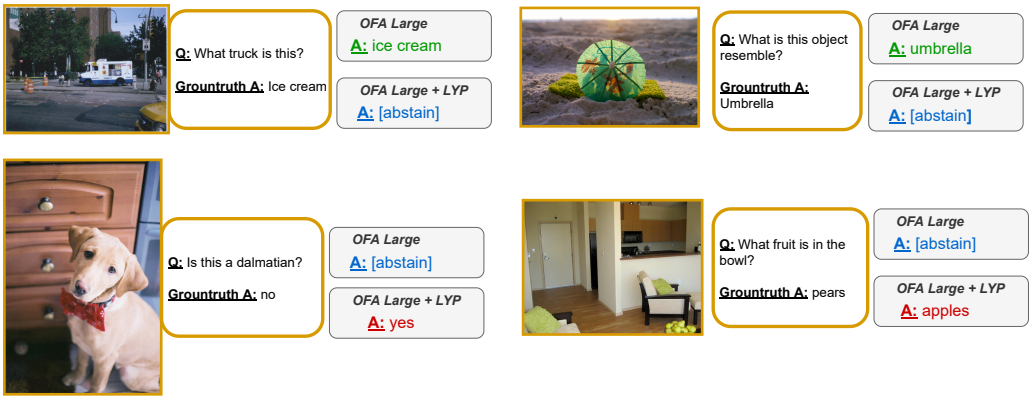


Figure 6.9. – Failure cases: on the first two examples, the baseline predicts the correct answer, and OFA-L + LYP abstains. On the second line, the baseline abstains from answering an incorrect answer, while OFA-L + LYP still answers.

6.7 Conclusion

In this chapter, we explore Selective Visual Question Answering in the realistic, but challenging mixed ID+OOD scenario, where a model is exposed to samples from both the training distribution and also out-of-distribution (OOD) samples. We find that out-of-the-box, state-of-the-art VQA models like OFA largely fail on this task at a low risk of error (e.g., 1%). When training a multimodal selector, models significantly improve their reliability, matching observations in the in-distribution (ID) scenario. However, a limitation of the selector training is that it requires splitting the training data between the VQA model training and the Selector training to avoid over-fitting on the training data. We address this with our approach Learning from Your Peers (LYP), which allows us to train both the VQA model and the Selector on the full training set. We observe that in the ID scenario as well as the mixed ID/OOD scenario, LYP consistently performs best across all VQA models and Metrics and improves over baselines and prior work. Our best result doubles the $\mathcal{C}@1\%$ over prior work. Overall, all models still have difficulties recognizing when they cannot answer OOD examples correctly and thus decrease in performance when the percentage of OOD increases. Thus, major challenges remain, both for improving the generalizing abilities of VQA models to OOD examples (i.e., answering them correctly) as well as identifying examples that the model cannot answer, whether they are in-distribution or out-of-distribution.

Chapter 7

CONCLUSION

7.1 Summary of Contributions

We first summarize the contributions that we propose in this thesis before discussing directions for future work. The problem we tackle is shortcut learning in Visual Question Answering: the ability of a model to learn superficial statistical regularities from the data, which can lead to answering correctly in most cases, but without learning the correct answering mechanism. This is a dramatic problem for deploying VQA models in the real world, especially in safety-critical applications, like assisting visually impaired people or autonomous driving, where unexpected cases will be encountered.

The first direction we explore is to **reduce shortcut learning** in VQA models, by adding constraints on the model. As mentioned in Geirhos et al. 2020, shortcuts are learned partially due to the *simplicity bias*: models are biased towards a simpler solution, and the inductive biases that have an impact on the hypothesis class are the architecture, the training data, and the learning algorithm and loss.

In this thesis, we show that we can effectively modify those inductive biases in order to reduce shortcut learning: by having an *a-priori* on the source of biases and the type of shortcut, we can design a system that is more robust to those biases. First, in Chapter 3, we explore how the **loss function** can be modified to reduce question-based shortcuts: we train the main model together with a "blind" model, as a kind of *mixture of experts*, and use only the main VQA model for inference. This allows the blind model to learn the shortcuts, and the main model to learn the desired behavior. Then, in Chapter 4, we explore another source of shortcut learning: the **architecture**. We show that by incorporating domain knowledge in the model, we can improve its generalization abilities to out-of-distribution datasets: for the Visual Counting task, a subset of VQA with only counting questions, we design the Spatial Counting Network model, that assigns a spatial distribution to the final answer. We also incorporate an entropy regularization loss: we impact both the architecture and the loss. This allows the model to be more robust to superficial shortcuts and is also explainable by design.

In this direction, multiple works have followed ours, exploring various strategies. For example, Kervadec et al. 2020 proposes an additional loss for VQA models that takes into account the semantic structure of the answer space: if the ground truth answer is "red", models should be less penalized for answering "pink" than "car". Teney et al. 2020a explores a third axis of shortcut learning: the *data* itself. They propose to create counterfactual examples, i.e. "minimally different" examples that are very close to a training example, but with a different answer: more specifically, they modify the image to make the question unanswerable, and force the model to output no answer. This forces the model to learn to use the image modality, and not rely only on the question, thus learning the causal relationship of the task. L. Chen et al. 2020 also proposes to use synthetic counterfactual examples as additional training data, by changing the colors of objects, or removing them using image inpainting, to change the ground truth answer.

The second direction we explore is the **detection of shortcuts** in datasets, and the evaluation of models: in order to effectively reduce shortcut learning, it is critical to know more about the source of shortcuts. We show in Chapter 5 that there are a large amount of multimodal shortcuts as well in the VQA v2 dataset. They are more subtle than the previously studied question-based shortcuts. We first propose a shortcut detection method based on object detection and rule mining algorithms, and an evaluation benchmark, which addresses some of the shortcomings of VQA-CP: our benchmark evaluates models against real VQA v2 shortcuts and does not introduce an artificial shift in distribution to create shortcuts. We show that most shortcut reduction methods proposed for VQA are only effective for question-based shortcuts, but do not seem to reduce multimodal shortcut learning. This paves the way for future work to explore multimodal shortcut reduction methods.

Finally, we explore in Chapter 6 a tangential but related topic: **reliability in VQA models** in the **out-of-distribution context**. As we saw previously, models tend to rely on shortcuts and might fail catastrophically in real-world scenarios. Parallel to fixing those issues, we can try to detect cases where a model will fail, and give the model the ability to abstain from answering those. We find that models of all scales are very unreliable out of the box and fail to estimate their own confidence, especially in out-of-distribution settings. We explore various solutions to tackle this issue and find that the most effective solution is to train a dedicated selector function to estimate the model's confidence. We propose *Learning from your peers*, a strategy to share the model's and selector's training data to maximize their performances given a fixed amount of data. This strategy effectively improves the reliability of VQA models, in both in-distribution and out-of-distribution settings.

Overall, we show that VQA models are very sensitive to shortcuts, and that shortcut learning is a major issue in VQA. We explore various strategies to reduce shortcut learning and show that most of them are only effective for question-based shortcuts, and do not seem to reduce multimodal shortcut learning. We also explore the reliability of VQA models and show how to improve them.

7.2 Perspective for Future Works

Large Vision and Language models The trend in vision and language is now large transformer models, pre-trained on a very large amount of aligned image and text data. VQA, as most of the vision-and-language tasks, is strongly impacted by this trend: It seems clear now that using only datasets of the size of VQA v2 is not enough to reach human performance, given the difficulty of the task. Eventually, a good use of the VQA dataset seems to be as a zero-shot or few-shot downstream task for large pre-trained models: models should be able to answer questions without any dedicated VQA training. Flamingo (Alayrac et al. 2022) shows promising results in this direction. A potential direction is to build multimodal models by adapting large language models to vision-and-language data, such as BLIP (J. Li et al. 2022), Flamingo (Alayrac et al. 2022) or OFA (P. Wang et al. 2022), without having to train huge models from scratch. All those works use the Transformer architecture, which is very effective for vision-and-language tasks.

Additionally, the reliability and calibration of large multimodal models is an active research area. Kadavath et al. 2022a suggests, for Natural Language Processing (NLP) tasks, that large language models might be less susceptible to shortcut learning. This remains underexplored for multimodal learning, and we hope to see more work in this direction. By avoiding training on small-sized datasets like VQA, shortcut learning might be less of an issue: models are not able to use training shortcuts if they are evaluated in a zero-short or few-shot fashion.

Data efficiency Scaling of models and data size seems to be one of the most promising directions to train better models. A very important research direction is training efficiency: how to train the best models with the least amount of data and compute power. Those models are very costly, for instance, the Flamingo that we mentioned earlier is trained for 15 days using 1536 TPU chips, on 185 million images and 182 Gb of text, and this is likely to increase for future models. How to extract the most value from each example is a very important question.

Human Interaction One last perspective is human interaction: those systems seem very strong on our datasets, but there is a need for human-in-the-loop research, where those systems are deployed to real users. Training with humans in the loop to align the model's behavior to human expectations has been shown effective with large language models such as InstructGPT (Ouyang et al. 2022) and ChatGPT (OpenAI 2022). In the multimodal case, humans-in-the-loop could help both for training and evaluation, for example with visually impaired users for VQA. Real questions are much harder than the ones in the VQA dataset, due to the quality of images and the variability in questions (Gurari et al. 2018). There is still a lot of exciting work to do in this direction.

BIBLIOGRAPHY

Abacha, Asma Ben, Sadid A. Hasan, Vivek Datla, Joey Liu, Dina Demner-Fushman, and Henning Müller (2019). "VQA-Med: Overview of the Medical Visual Question Answering Task at ImageCLEF 2019". In: *Conference and Labs of the Evaluation Forum* (cit. on p. 1).

Abdar, Moloud, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. (2021). "A review of uncertainty quantification in deep learning: Techniques, applications and challenges". In: *Information Fusion* 76, pp. 243–297 (cit. on p. 106).

Acharya, Manoj, Kushal Kafle, and Christopher Kanan (2019). "TallyQA: Answering complex counting questions". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)* (cit. on pp. 33, 56, 59, 60, 65, 66, 68–70, 74, 75, 174).

Agrawal, Aishwarya, Dhruv Batra, and Devi Parikh (2016). "Analyzing the behavior of visual question answering models". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (cit. on p. 26).

Agrawal, Aishwarya, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi (2018a). "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 (CVPR)* (cit. on pp. 29–31, 60, 62, 96, 97, 113, 175).

Agrawal, Aishwarya, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi (2018b). "Don't Just Assume; Look and Answer: Overcoming Priors for Visual Question Answering". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 32, 33, 38, 39, 43, 45–47, 173).

Agrawal, Aishwarya, Ivana Kajić, Emanuele Bugliarello, Elnaz Davoodi, Anita Gergely, Phil Blunsom, and Aida Nematzadeh (2022). "Rethinking Evaluation Practices in Visual Question Answering: A Case Study on Out-of-Distribution Generalization". In: *arXiv preprint arXiv:2205.12191* (cit. on pp. 105, 113).

Agrawal, Rakesh, Ramakrishnan Srikant, et al. (1994). "Fast algorithms for mining association rules". In: *Proc. 20th int. conf. very large data bases, VLDB*. Vol. 1215. Citeseer, pp. 487–499 (cit. on p. 82).

Alayrac, Jean-Baptiste, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. (2022). "Flamingo: a visual language model for few-shot learning". In: *arXiv preprint arXiv:2204.14198* (cit. on p. 127).

Alcorn, Michael A, Qi Li, Zhitao Gong, Chengfei Wang, Long Mai, Wei-Shinn Ku, and Anh Nguyen (2019). "Strike (with) a pose: Neural networks are

easily fooled by strange poses of familiar objects". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 24).

Anand, Ankesh, Eugene Belilovsky, Kyle Kastner, Hugo Larochelle, and Aaron Courville (2018). "Blindfold baselines for embodied qa". In: *arXiv preprint arXiv:1811.05013* (cit. on p. 23).

Anderson, Peter, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang (2018a). "Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 18, 39, 43, 59, 64, 67, 69, 81, 84, 88, 91, 96, 97, 156, 157, 170, 175–177).

Anderson, Peter, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang (June 2018b). "Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 43, 45, 46, 173).

Anderson, Peter, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang (2018c). "Bottom-up and top-down attention for image captioning and visual question answering". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6077–6086 (cit. on p. 20).

Andreas, Jacob, Marcus Rohrbach, Trevor Darrell, and Dan Klein (2016). "Neural module networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 17).

Angelov, Plamen P. (2021). "Keynote: Explainable-by-design Deep Learning". In: *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pp. 175–175 (cit. on p. 26).

Antol, Stanislaw, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh (2015a). "VQA: Visual Question Answering". In: *International Conference on Computer Vision (ICCV)* (cit. on pp. 1, 11, 12, 15, 16, 26, 38, 43, 165).

Antol, Stanislaw, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh (2015b). "VQA: Visual Question Answering". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on pp. 59, 108).

Arjovsky, Martin, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz (2019). "Invariant risk minimization". In: *arXiv preprint arXiv:1907.02893* (cit. on pp. 25, 26).

Arpit, Devansh, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. (2017). "A closer look at memorization in deep networks".

In: *International conference on machine learning*. PMLR, pp. 233–242 (cit. on p. 22).

Arteta, Carlos, Victor Lempitsky, and Andrew Zisserman (2016). "Counting in the wild". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on p. 59).

Babu Sam, Deepak, Shiv Surya, and R. Venkatesh Babu (2017). "Switching Convolutional Neural Network for Crowd Counting". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 59).

Barbu, Andrei, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz (2019). "ObjectNet: A large-scale bias-controlled dataset for pushing the limits of object recognition models". In: *Advances in Neural Information Processing Systems (NeurIPS)* (cit. on p. 24).

Ben-Younes, Hedi, Remi Cadene, Nicolas Thome, and Matthieu Cord (2017a). "MUTAN: Multimodal Tucker Fusion for Visual Question Answering". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on pp. 16, 59, 65, 66, 73, 174).

Ben-Younes, Hedi, Remi Cadene, Nicolas Thome, and Matthieu Cord (2019a). "BLOCK: Bilinear Superdiagonal Fusion for Visual Question Answering and Visual Relationship Detection". In: *Proceedings of the 33st Conference on Artificial Intelligence (AAAI)*. URL: http://remicadene.com/pdfs/paper_aaai2019.pdf (cit. on p. 43).

Ben-Younes, Hedi, Remi Cadene, Nicolas Thome, and Matthieu Cord (2019b). "BLOCK: Bilinear Superdiagonal Fusion for Visual Question Answering and Visual Relationship Detection". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)* (cit. on pp. 59, 96, 97, 156, 157, 175–177).

Ben-Younes, Hedi, Rémi Cadene, Matthieu Cord, and Nicolas Thome (2017b). "Mutan: Multimodal tucker fusion for visual question answering". In: *ICCV* (cit. on p. 20).

Bergman, Liron, Niv Cohen, and Yedid Hoshen (2020). "Deep nearest neighbor anomaly detection". In: *arXiv preprint arXiv:2002.10445* (cit. on p. 106).

Bhattacharyya, Anil (1946). "On a measure of divergence between two multinomial populations". In: *Sankhyā: the indian journal of statistics*, pp. 401–406 (cit. on pp. 64, 174).

Boser, Bernhard E., I. Ramadass Subramanian, and Vladimir Naumovich Vapnik (1992). "A training algorithm for optimal margin classifiers". In: *Annual Conference Computational Learning Theory* (cit. on p. 8).

Bottou, Léon et al. (1998). "Online learning and stochastic approximations". In: *On-line learning in neural networks* 17.9, p. 142 (cit. on p. 7).

Bourrand, Erwan, Luis Galarraga, Esther Galbrun, Elisa Fromont, and Alexandre Termier (2021). "Discovering Useful Compact Sets of Sequential Rules in

a Long Sequence". In: *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, pp. 1295–1299 (cit. on p. 83).

Briggs, C (2009). "Quality counts: new parameters in blood cell counting". In: *International journal of laboratory hematology* 31.3, pp. 277–297 (cit. on p. 57).

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. (2020). "Language models are few-shot learners". In: *Advances in neural information processing systems* 33, pp. 1877–1901 (cit. on pp. 9, 10, 19).

Cadene, Remi, Hedi Ben-Younes, Nicolas Thome, and Matthieu Cord (2019a). "MUREL: Multimodal Relational Reasoning for Visual Question Answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 16, 17, 39, 59).

Cadene, Remi, Hedi Ben-Younes, Nicolas Thome, and Matthieu Cord (2019b). "MUREL: Multimodal Relational Reasoning for Visual Question Answering". In: *IEEE Conference on Computer Vision and Pattern Recognition cvpr*. URL: http://remicadene.com/pdfs/paper_cvpr2019.pdf (cit. on pp. 43–45, 49, 173).

Cadene, Remi, Corentin Dancette, Hedi Ben-Younes, Matthieu Cord, and Devi Parikh (2019c). "RUBi: Reducing Unimodal Biases for Visual Question Answering". In: *Advances in Neural Information Processing Systems (NeurIPS)* (cit. on pp. 4, 33, 96, 156, 157, 175–177).

Cadene*, Remi, Corentin Dancette*, Hedi Ben-Younes, Matthieu Cord, and Devi Parikh (2019). "RUBi: Reducing Unimodal Biases for Visual Question Answering". In: *Advances in Neural Information Processing Systems (NeurIPS)* (cit. on p. 35).

Caron, Mathilde, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin (2021). "Emerging properties in self-supervised vision transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660 (cit. on p. 19).

Cavnar, William B, John M Trenkle, et al. (1994). "N-gram-based text categorization". In: *Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval*. Vol. 161175. Las Vegas, NV (cit. on p. 9).

Chao, Wei-Lun, Hexiang Hu, and Fei Sha (2018). "Being negative but constructively: Lessons learnt from creating better visual question answering datasets". In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)* (cit. on p. 21).

Chapelle, Olivier, Bernhard Scholkopf, and Alexander Zien (2009). "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]". In: *IEEE Transactions on Neural Networks* 20.3, pp. 542–542 (cit. on p. 7).

Chattopadhyay, Prithvijit, Ramakrishna Vedantam, Ramprasaath R Selvaraju, Dhruv Batra, and Devi Parikh (2017). "Counting everyday objects in everyday scenes".

In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 59, 65).

Chen, Long, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang (2020). "Counterfactual samples synthesizing for robust visual question answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 28, 96, 97, 126, 156, 157, 175–177).

Chen, Yen-Chun, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu (2019). "Uniter: Learning universal image-text representations". In: (cit. on p. 19).

Cho, Kyunghyun, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio (Oct. 2014). "On the Properties of Neural Machine Translation: Encoder–Decoder Approaches". In: *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Doha, Qatar: Association for Computational Linguistics, pp. 103–111. URL: https://aclanthology.org/W14-4012 (cit. on p. 9).

Cholakkal, Hisham, Guolei Sun, Fahad Shahbaz Khan, and Ling Shao (2019). "Object counting and instance segmentation with image-level supervision". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 59).

Chon, Kang-Wook, Sang-Hyun Hwang, and Min-Soo Kim (2018). "GMiner: A fast GPU-based frequent itemset mining method for large-scale data". In: *Information Sciences* 439, pp. 19–38 (cit. on p. 84).

Chow, Chi-Keung (1957). "An optimum character recognition system using decision functions". In: *IRE Transactions on Electronic Computers* EC-6.4, pp. 247–254 (cit. on p. 103).

Chowdhery, Aakanksha, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. (2022). "Palm: Scaling language modeling with pathways". In: *arXiv preprint arXiv:2204.02311* (cit. on p. 10).

Clark, Christopher, Mark Yatskar, and Luke Zettlemoyer (2019). "Don't Take the Easy Way Out: Ensemble Based Methods for Avoiding Known Dataset Biases". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4060–4073 (cit. on pp. 96, 97, 156, 157, 175–177).

Clark, Christopher, Mark Yatskar, and Luke Zettlemoyer (2020). "Learning to Model and Ignore Dataset Bias with Mixed Capacity Ensembles". In: *EMNLP* (cit. on p. 26).

Corbière, Charles, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez (2019). "Addressing failure prediction by learning model confidence". In: *Advances in Neural Information Processing Systems* 32 (cit. on p. 106).

D'Amour, Alexander, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. (2020). "Underspecification presents challenges for credibility in modern machine learning". In: *arXiv preprint arXiv:2011.03395* (cit. on p. 82).

Dalal, Navneet and Bill Triggs (2005). "Histograms of oriented gradients for human detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* 1, 886–893 vol. 1 (cit. on p. 8).

Dancette, Corentin, Remi Cadene, Xinlei Chen, and Matthieu Cord (2021a). "Learning Reasoning Mechanisms for Unbiased Question-based Counting". In: *VQA Workshop, Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 4, 33, 55).

Dancette, Corentin, Remi Cadene, Damien Teney, and Matthieu Cord (2021b). "Beyond Question-Based Biases: Assessing Multimodal Shortcut Learning in Visual Question Answering". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on pp. 4, 33, 79, 113).

Dancette, Corentin and Matthieu Cord (2022). "Dynamic Query Selection for Fast Visual Perceiver". In: *CVPR Workshop, Transformers for Vision* (cit. on pp. 5, 34).

Dancette, Corentin, Spencer Whitehead, Rishabh Maheshwary, Ramakrishna Vedantam, Stefan Scherer, Xinlei Chen, Matthieu Cord, and Marcus Rohrbach (2023). "Improving Selective VQA by learning from your peers". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 4, 101).

Das, Abhishek, Harsh Agrawal, C. Lawrence Zitnick, Devi Parikh, and Dhruv Batra (2016). "Human Attention in Visual Question Answering: Do Humans and Deep Networks Look at the Same Regions?" In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (cit. on pp. 43, 48–51, 168, 173).

Das, Abhishek, Harsh Agrawal, Larry Zitnick, Devi Parikh, and Dhruv Batra (2017a). "Human attention in visual question answering: Do humans and deep networks look at the same regions?" In: *Computer Vision and Image Understanding* 163, pp. 90–100 (cit. on pp. 24, 31, 32).

Das, Abhishek, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M.F. Moura, Devi Parikh, and Dhruv Batra (2017b). "Visual Dialog". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 1).

DeGrave, Alex J, Joseph D Janizek, and Su-In Lee (2021). "AI for radiographic COVID-19 detection selects shortcuts over signal". In: *Nature Machine Intelligence* 3.7, pp. 610–619 (cit. on p. 102).

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). "Bert: Pre-training of deep bidirectional transformers for language understanding".

In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)* (cit. on pp. 9, 19).

Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. (2021). "An image is worth 16x16 words: Transformers for image recognition at scale". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on p. 10).

Dwork, Cynthia, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth (2015). "Preserving statistical validity in adaptive data analysis". In: *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pp. 117–126 (cit. on p. 62).

Everingham, Mark, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman (2015). "The pascal visual object classes challenge: A retrospective". In: *International Journal of Computer Vision (IJCV)* (cit. on p. 74).

Fauvel, Kevin, Elisa Fromont, Véronique Masson, Philippe Faverdin, and Alexandre Termier (2022). "XEM: An explainable-by-design ensemble method for multivariate time series classification". In: *Data Mining and Knowledge Discovery* 36.3, pp. 917–957 (cit. on p. 26).

Fedus, William, Barret Zoph, and Noam Shazeer (2021). *Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity* (cit. on p. 10).

Fernando, Basura, Elisa Fromont, and Tinne Tuytelaars (2012). "Effective use of frequent itemset mining for image classification". In: *European conference on computer vision*. Springer, pp. 214–227 (cit. on p. 83).

Fisch, Adam, Tommi Jaakkola, and Regina Barzilay (2022). "Calibrated Selective Classification". In: *arXiv preprint arXiv:2208.12084* (cit. on pp. 121, 160).

Fong, Ruth C and Andrea Vedaldi (2017). "Interpretable explanations of black boxes by meaningful perturbation". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on p. 24).

Fournier, Jérome, Matthieu Cord, and Sylvie Philipp-Foliguet (2001). "RETIN: A Content-Based Image Indexing and Retrieval System". In: *Pattern Analysis & Applications* 4, pp. 153–173 (cit. on p. 8).

Fukui, Akira, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach (2016a). "Multimodal compact bilinear pooling for visual question answering and visual grounding". In: *EMNLP* (cit. on p. 20).

Fukui, Akira, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach (2016b). "Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding." In: *EMNLP*. The Association for Computational Linguistics (cit. on p. 16).

Gat, Itai, Idan Schwartz, Alexander Schwing, and Tamir Hazan (2020). "Removing Bias in Multi-modal Classifiers: Regularization by Maximizing Functional

Entropies". In: *Advances in Neural Information Processing Systems (NeurIPS)* (cit. on pp. 96, 97, 156, 157, 175–177).

Gawlikowski, Jakob, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. (2021). "A survey of uncertainty in deep neural networks". In: *arXiv preprint arXiv:2107.03342* (cit. on p. 106).

Geifman, Yonatan and Ran El-Yaniv (2017). "Selective classification for deep neural networks". In: *Advances in neural information processing systems* 30 (cit. on p. 111).

Geifman, Yonatan and Ran El-Yaniv (2019). "Selectivenet: A deep neural network with an integrated reject option". In: *International Conference on Machine Learning*. PMLR, pp. 2151–2159 (cit. on pp. 106, 109, 110, 159).

Geiger, Atticus, Ignacio Cases, Lauri Karttunen, and Christopher Potts (Nov. 2019). "Posing Fair Generalization Tasks for Natural Language Inference". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 4485–4495. URL: https://aclanthology.org/D19-1456 (cit. on p. 103).

Geirhos, Robert, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann (2020). "Shortcut Learning in Deep Neural Networks". In: *arXiv preprint arXiv:2004.07780* (cit. on pp. 2, 21–23, 26, 27, 102, 125).

Geirhos, Robert, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel (2019). "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on pp. 22, 23).

Geman, Donald, Stuart Geman, Neil Hallonquist, and Laurent Younes (2015). "Visual turing test for computer vision systems". In: *Proceedings of the National Academy of Sciences* 112.12, pp. 3618–3623 (cit. on pp. 1, 11).

Gers, Felix A and Jürgen Schmidhuber (2001). "LSTM recurrent networks learn simple context-free and context-sensitive languages". In: *IEEE Transactions on Neural Networks* 12.6, pp. 1333–1340 (cit. on pp. 9, 15).

Gordon, Jonathan and Benjamin Van Durme (2013). "Reporting bias and knowledge acquisition". In: *Proceedings of the 2013 workshop on Automated knowledge base construction*. ACM (cit. on p. 21).

Goyal, Yash, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh (2017a). "Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering". In: *IEEE Conference on Computer Vision and Pattern Recognition cvpr* (cit. on pp. 1, 11, 20, 27, 28, 38, 43, 167).

Goyal, Yash, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh (2017b). "Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 59, 89, 91, 175).

Goyal, Yash, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh (2017c). "Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6904–6913 (cit. on pp. 104, 113).

Grill, Jean-Bastien, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. (2020). "Bootstrap your own latent-a new approach to self-supervised learning". In: *Advances in neural information processing systems* 33, pp. 21271–21284 (cit. on p. 19).

Gulrajani, Ishaan and David Lopez-Paz (2021). "In Search of Lost Domain Generalization". In: *ICLR* (cit. on p. 25).

Guo, Chuan, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger (2017). "On calibration of modern neural networks". In: *International Conference on Machine Learning*. PMLR, pp. 1321–1330 (cit. on p. 111).

Gupta, Abhinav, Adithyavairavan Murali, Dhiraj Prakashchand Gandhi, and Lerrel Pinto (2018). "Robot learning in homes: Improving generalization and reducing dataset bias". In: *Advances in Neural Information Processing Systems*, pp. 9094–9104 (cit. on p. 23).

Gurari, Danna, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham (2018). "Vizwiz grand challenge: Answering visual questions from blind people". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3608–3617 (cit. on pp. 1, 11, 13, 15, 102, 128, 161).

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). "Unsupervised learning". In: *The elements of statistical learning*. Springer, pp. 485–585 (cit. on p. 7).

He, Kaiming, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick (2022). "Masked autoencoders are scalable vision learners". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009 (cit. on p. 19).

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 111).

Hendricks, Lisa Anne, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach (2018a). "Women Also Snowboard: Overcoming Bias in Captioning Models". In: *ECCV* (cit. on p. 23).

Hendricks, Lisa Anne, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach (2018b). "Women also snowboard: Overcoming bias in captioning models". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pp. 771–787 (cit. on p. 22).

Hendrycks, Dan and Thomas Dietterich (2019). "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations". In: *ICLR* (cit. on p. 24).

Hendrycks, Dan and Kevin Gimpel (2017). "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks". In: *Proceedings of International Conference on Learning Representations* (cit. on pp. 106, 111).

Hu, Ronghang, Jacob Andreas, Trevor Darrell, and Kate Saenko (2018). "Explainable Neural Computation via Stack Neural Module Networks". In: *ECCV* (cit. on p. 17).

Hu, Ronghang, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko (2017). "Learning to reason: End-to-end module networks for visual question answering". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on p. 17).

Hudson, Drew A and Christopher D Manning (2019). "GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 13, 14, 28, 82).

Hudson, Drew Arad and Christopher D. Manning (2018). "Compositional Attention Networks for Machine Reasoning". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on p. 59).

Jabri, Allan, Armand Joulin, and Laurens Van Der Maaten (2016). "Revisiting visual question answering baselines". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*. Springer, pp. 727–739 (cit. on p. 38).

Jia, Sen, Thomas Lansdall-Welfare, and Nello Cristianini (2018a). "Right for the Right Reason: Training Agnostic Networks". In: *Lecture Notes in Computer Science*, pp. 164–174. URL: http://dx.doi.org/10.1007/978-3-030-01768-2_14 (cit. on p. 21).

Jia, Sen, Thomas Lansdall-Welfare, and Nello Cristianini (2018b). "Right for the Right Reason: Training Agnostic Networks". In: *Lecture Notes in Computer Science*, pp. 164–174 (cit. on p. 21).

Jiang, Huaizu, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen (2020). "In Defense of Grid Features for Visual Question Answering". In: *arXiv preprint arXiv:2001.03615* (cit. on p. 20).

Jiang, Yu, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh (2018). "Pythia v0.1: the winning entry to the vqa challenge 2018". In: *arXiv preprint arXiv:1807.09956* (cit. on pp. 18, 20).

Jiaxin Shi Hanwang Zhang, Juanzi Li (2019). "Explainable and Explicit Visual Reasoning over Scene Graphs". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 17).

Johnson, Jeff, Matthijs Douze, and Hervé Jégou (2019). "Billion-scale similarity search with GPUs". In: *IEEE Transactions on Big Data* 7.3, pp. 535–547 (cit. on p. 160).

Johnson, Justin, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick (2017a). "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 1).

Johnson, Justin, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick (2017b). "CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning". In: *IEEE Conference on Computer Vision and Pattern Recognition cvpr* (cit. on pp. 13, 28).

Kadavath, Saurav, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield Dodds, Nova DasSarma, Eli Tran-Johnson, et al. (2022a). "Language Models (Mostly) Know What They Know". In: *arXiv preprint arXiv:2207.05221* (cit. on p. 127).

Kadavath, Saurav, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield Dodds, Nova DasSarma, Eli Tran-Johnson, et al. (2022b). "Language models (mostly) know what they know". In: *arXiv preprint arXiv:2207.05221* (cit. on p. 119).

Kafle, Kushal and Christopher Kanan (2017). "An Analysis of Visual Question Answering Algorithms". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on pp. 14, 59).

Kalimeris, Dimitris, Gal Kaplun, Preetum Nakkiran, Benjamin Edelman, Tristan Yang, Boaz Barak, and Haofeng Zhang (2019). "Sgd on neural networks learns functions of increasing complexity". In: *Advances in neural information processing systems* 32 (cit. on p. 22).

Kamath, Amita, Robin Jia, and Percy Liang (July 2020). "Selective Question Answering under Domain Shift". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 5684–5696. URL: https://aclanthology.org/2020.acl-main.503 (cit. on pp. 103, 104, 106–109, 111, 121, 161).

Karamcheti, Siddharth, Ranjay Krishna, Li Fei-Fei, and Christopher Manning (Aug. 2021). "Mind Your Outliers! Investigating the Negative Impact of Outliers on Active Learning for Visual Question Answering". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, pp. 7265–7281. URL: https://aclanthology.org/2021.acl-long.564 (cit. on p. 110).

Kärkkäinen, Kimmo and Jungseock Joo (2019). "FairFace: Face Attribute Dataset for Balanced Race, Gender, and Age for Bias Measurement and Mitigation". In: *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1547–1557 (cit. on p. 24).

Karpathy, Andrej and Li Fei-Fei (2015). "Deep visual-semantic alignments for generating image descriptions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 1).

Kazemi, Vahid and Ali Elqursh (2017). "Show, ask, attend, and answer: A strong baseline for visual question answering". In: *arXiv preprint arXiv:1704.03162* (cit. on p. 82).

Kervadec, Corentin, Grigory Antipov, Moez Baccouche, and Christian Wolf (2020). "Estimating semantic structure for the VQA answer space". In: *arXiv preprint arXiv:2006.05726* (cit. on pp. 30, 53, 126).

Kervadec, Corentin, Grigory Antipov, Moez Baccouche, and Christian Wolf (2021). "Roses are red, violets are blue... but should vqa expect them to?" In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2776–2785 (cit. on pp. 30, 82).

Kim, Byungju, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim (2019). "Learning not to learn: Training deep neural networks with biased data". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9012–9020 (cit. on p. 25).

Kim, Jin-Hwa, Jaehyun Jun, and Byoung-Tak Zhang (2018). "Bilinear attention networks". In: *Advances in Neural Information Processing Systems*, pp. 1564–1574 (cit. on pp. 20, 45, 173).

Kim, Jin-Hwa, Kyoung-Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang (2017). "Hadamard product for low-rank bilinear pooling". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on pp. 16, 67).

Kingma, Diederik P. and Jimmy Ba (2015). "Adam: A Method for Stochastic Optimization". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on p. 69).

Kiros, Ryan, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler (2015a). "Skip-thought Vectors". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on pp. 67, 69).

Kiros, Ryan, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler (2015b). "Skip-thought vectors". In: *Advances in neural information processing systems*, pp. 3294–3302 (cit. on pp. 1, 43).

Krishna, Ranjay, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. (2017a). "Visual genome: Connecting language and vision using crowdsourced dense

image annotations". In: *International journal of computer vision* 123.1, pp. 32–73 (cit. on pp. 13, 19).

Krishna, Ranjay, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. (2017b). "Visual genome: Connecting language and vision using crowdsourced dense image annotations". In: *International Journal of Computer Vision (IJCV)* 123.1, pp. 32–73 (cit. on pp. 59, 60, 74).

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012a). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Curran Associates, Inc., pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf (cit. on p. 1).

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012b). "Imagenet classification with deep convolutional neural networks". In: *NeurIPS* (cit. on p. 8).

Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell (2017). "Simple and scalable predictive uncertainty estimation using deep ensembles". In: *Advances in neural information processing systems*. Vol. 30 (cit. on p. 106).

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning". In: *nature* 521.7553, pp. 436–444 (cit. on p. 7).

Lee, Kimin, Kibok Lee, Honglak Lee, and Jinwoo Shin (2018). "A simple unified framework for detecting out-of-distribution samples and adversarial attacks". In: *Advances in neural information processing systems* 31 (cit. on pp. 106, 160).

Lempitsky, Victor and Andrew Zisserman (2010). "Learning to count objects in images". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on pp. 57, 59).

Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer (2019). "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension". In: *arXiv preprint arXiv:1910.13461* (cit. on p. 20).

Li, Junnan, Dongxu Li, Caiming Xiong, and Steven Hoi (2022). "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation". In: *International Conference on Machine Learning*. PMLR, pp. 12888–12900 (cit. on p. 127).

Li, Linjie, Jie Lei, Zhe Gan, and Jingjing Liu (2021). "Adversarial vqa: A new benchmark for evaluating the robustness of vqa models". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2042–2051 (cit. on p. 105).

Li, Xiujun, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. (2020). "Oscar: Object-semantics

aligned pre-training for vision-language tasks". In: *European Conference on Computer Vision*. Springer, pp. 121–137 (cit. on p. 19).

Liang, Shiyu, Yixuan Li, and Rayadurgam Srikant (2018). "Enhancing the reliability of out-of-distribution image detection in neural networks". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on p. 106).

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick (2014a). "Microsoft coco: Common objects in context". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on pp. 11, 19).

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick (2014b). "Microsoft coco: Common objects in context". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*. Springer, pp. 740–755 (cit. on pp. 60, 74, 155).

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick (2014c). "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer, pp. 740–755 (cit. on p. 114).

Lin, Ziqian, Sreya Dutta Roy, and Yixuan Li (2021). "Mood: Multi-level out-of-distribution detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15313–15323 (cit. on p. 106).

Liu, Evan Z, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn (2021). "Just train twice: Improving group robustness without training group information". In: *International Conference on Machine Learning*. PMLR, pp. 6781–6792 (cit. on pp. 26, 110).

Liu, Jiang, Chenqiang Gao, Deyu Meng, and Alexander G. Hauptmann (2018). "DecideNet: Counting Varying Density Crowds Through Attention Guided Detection and Density Estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 59).

Liu, Weitang, Xiaoyun Wang, John Owens, and Yixuan Li (2020). "Energy-based out-of-distribution detection". In: *Advances in Neural Information Processing Systems* 33, pp. 21464–21475 (cit. on p. 106).

Liu, Yuting, Miaojing Shi, Qijun Zhao, and Xiaofang Wang (2019). "Point in, box out: Beyond counting persons in crowds". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 59).

Lowe, David G (1999). "Object recognition from local scale-invariant features". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Vol. 2. Ieee, pp. 1150–1157 (cit. on p. 8).

Lu, Cewu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei (2016). "Visual relationship detection with language priors". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*. Springer, pp. 852–869 (cit. on p. 1).

Lu, Jiasen, Dhruv Batra, Devi Parikh, and Stefan Lee (2019). "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks". In: *Advances in Neural Information Processing Systems (NeurIPS)* (cit. on pp. 65, 68, 81, 88, 91, 96, 97, 99, 156, 157, 170, 175–177).

Malinowski, Mateusz and Mario Fritz (2014a). "A multi-world approach to question answering about real-world scenes based on uncertain input". In: *Advances in Neural Information Processing Systems (NIPS)* 27 (cit. on p. 11).

Malinowski, Mateusz and Mario Fritz (2014b). "Towards a Visual Turing Challenge". In: *Learning Semantics* (cit. on p. 59).

Manjunatha, Varun, Nirat Saini, and Larry S. Davis (Nov. 2019a). "Explicit Bias Discovery in Visual Question Answering Models". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. arXiv: 1811.07789. URL: http://arxiv.org/pdf/1811.07789v1 (cit. on pp. 21, 38).

Manjunatha, Varun, Nirat Saini, and Larry S. Davis (2019b). "Explicit Bias Discovery in Visual Question Answering Models". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 24, 82).

Marino, Kenneth, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi (2019). "Ok-vqa: A visual question answering benchmark requiring external knowledge". In: *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pp. 3195–3204 (cit. on pp. 105, 161).

Marsden, Mark, Kevin McGuinness, Suzanne Little, Ciara E Keogh, and Noel E O'Connor (2018). "People, penguins and petri dishes: Adapting object counting models to new visual domains and object types without forgetting". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 59).

Mårtensson, Gustav, Daniel Ferreira, Tobias Granberg, Lena Cavallin, Ketil Oppedal, Alessandro Padovani, Irena Rektorova, Laura Bonanni, Matteo Pardini, Milica G Kramberger, et al. (2020). "The reliability of a deep learning model in clinical out-of-distribution MRI data: a multicohort study". In: *Medical Image Analysis* 66, p. 101714 (cit. on p. 102).

McCoy, Tom, Ellie Pavlick, and Tal Linzen (2019). "Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3428–3448 (cit. on p. 24).

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). "Efficient estimation of word representations in vector space". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on pp. 1, 9).

Misra, Ishan, C Lawrence Zitnick, Margaret Mitchell, and Ross Girshick (2016). "Seeing through the human reporting bias: Visual classifiers from noisy human-

centric labels". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2930–2939 (cit. on p. 23).

Nam, Junhyun, Hyuntak Cha, Sung-Soo Ahn, Jaeho Lee, and Jinwoo Shin (2020). "Learning from Failure: De-biasing Classifier from Biased Classifier". In: *Advances in Neural Information Processing Systems* 33 (cit. on pp. 25, 26, 96, 98, 155–157, 175–177).

Nguyen, Duy-Kien, Vedanuj Goswami, and Xinlei Chen (2021). "Movie: Revisiting modulated convolutions for visual counting and beyond". In: *Proceedings of the International Conference on Learning Representations* (cit. on p. 111).

Onoro-Rubio, Daniel and Roberto J López-Sastre (2016). "Towards perspective-free object counting with deep learning". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on pp. 57, 59).

OpenAI, TB (2022). "Chatgpt: Optimizing language models for dialogue". In: *OpenAI* (cit. on p. 128).

Ouyang, Long, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. (2022). "Training language models to follow instructions with human feedback". In: *arXiv preprint arXiv:2203.02155* (cit. on p. 128).

Ovadia, Yaniv, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek (2019). "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift". In: *Advances in neural information processing systems* 32 (cit. on p. 106).

Perez, Ethan, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville (2018). "Film: Visual reasoning with a general conditioning layer". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)* (cit. on p. 59).

Pezeshki, Mohammad, Oumar Kaba, Yoshua Bengio, Aaron C Courville, Doina Precup, and Guillaume Lajoie (2021). "Gradient starvation: A learning proclivity in neural networks". In: *Advances in Neural Information Processing Systems (NeurIPS)* 34, pp. 1256–1272 (cit. on p. 22).

Quack, Till, Vittorio Ferrari, Bastian Leibe, and Luc Van Gool (2007). "Efficient mining of frequent and distinctive feature configurations". In: *2007 11th IEEE International Conference on Computer Vision*. IEEE Computer Society, pp. 1–8 (cit. on p. 83).

Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. (2021). "Learning transferable visual models from natural language supervision". In: *International Conference on Machine Learning*. PMLR, pp. 8748–8763 (cit. on pp. 10, 111).

Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever (2018). "Improving language understanding with unsupervised learning". In: (cit. on pp. 9, 19).

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. (2019). "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8, p. 9 (cit. on pp. 9, 19).

Ramakrishnan, Sainandan, Aishwarya Agrawal, and Stefan Lee (2018). "Overcoming language priors in visual question answering with adversarial regularization". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on pp. 29, 30, 33, 39, 43–47, 168, 173).

Rame, Alexandre, Corentin Dancette, and Matthieu Cord (2022). "Fishr: Invariant gradient variances for out-of-distribution generalization". In: *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 18347–18377 (cit. on pp. 5, 33).

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun (2015). "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on pp. 18, 69).

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016). ""Why should I trust you?" Explaining the predictions of any classifier". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (cit. on p. 24).

Robertson, Alexander M and Peter Willett (1998). "Applications of n-grams in textual information systems". In: *Journal of Documentation* (cit. on p. 9).

Rohrbach, Anna, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko (2018). "Object Hallucination in Image Captioning". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (cit. on p. 23).

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams (1986). "Learning representations by back-propagating errors". In: *nature* 323.6088, pp. 533–536 (cit. on p. 7).

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. (2015a). "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115.3, pp. 211–252 (cit. on p. 8).

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei (2015b). "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3, pp. 211–252 (cit. on p. 15).

Sagawa, Shiori, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang (2020). "Distributionally Robust Neural Networks". In: *ICLR* (cit. on p. 26).

Santoro, Adam, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Tim Lillicrap (2017). "A simple neural network module for relational reasoning". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on pp. 59, 60, 65).

Schölkopf, Bernhard, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio (2021). "Toward causal representation learning". In: *Proceedings of the IEEE* 109.5, pp. 612–634 (cit. on p. 23).

Sehwag, Vikash, Mung Chiang, and Prateek Mittal (2021). "Ssd: A unified framework for self-supervised outlier detection". In: *arXiv preprint arXiv:2103.12051* (cit. on pp. 121, 160).

Selvaraju, Ramprasaath R, Stefan Lee, Yilin Shen, Hongxia Jin, Shalini Ghosh, Larry Heck, Dhruv Batra, and Devi Parikh (2019). "Taking a hint: Leveraging explanations to make vision and language models more grounded". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 28).

Shah, Harshay, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli (2020). "The pitfalls of simplicity bias in neural networks". In: *Advances in Neural Information Processing Systems* 33, pp. 9573–9585 (cit. on p. 22).

Sharma, Piyush, Nan Ding, Sebastian Goodman, and Radu Soricut (2018). "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning". In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL Short Papers)* (cit. on pp. 19, 97).

Shen, Sheng, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer (2021). "How Much Can CLIP Benefit Vision-and-Language Tasks?" In: *arXiv preprint arXiv:2107.06383* (cit. on p. 111).

Sheng, Sasha, Amanpreet Singh, Vedanuj Goswami, Jose Magana, Tristan Thrush, Wojciech Galuba, Devi Parikh, and Douwe Kiela (2021). "Human-adversarial visual question answering". In: *Advances in Neural Information Processing Systems* 34, pp. 20346–20359 (cit. on pp. 104, 105, 113, 117).

Shrestha, Robik, Kushal Kafle, and Christopher Kanan (2019). "Answer Them All! Toward Universal Visual Question Answering Models". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 45, 173).

Shrestha, Robik, Kushal Kafle, and Christopher Kanan (2020). "A negative case analysis of visual grounding methods for VQA". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8172–8181 (cit. on pp. 96, 97, 156, 157, 175–177).

Sidorov, Oleksii, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh (2020). "Textcaps: a dataset for image captioning with reading comprehension". In: *European conference on computer vision*. Springer, pp. 742–758 (cit. on p. 102).

Simonyan, Karen and Andrew Zisserman (2015). "Very deep convolutional networks for large-scale image recognition". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on pp. 15, 16).

Sindagi, Vishwanath A and Vishal M Patel (2018). "A survey of recent advances in cnn-based single image crowd counting and density estimation". In: *Pattern Recognition Letters* 107, pp. 3–16 (cit. on pp. 59, 65).

Singh, Amanpreet, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela (2022). "Flava: A foundational language and vision alignment model". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15638–15650 (cit. on p. 19).

Singh, Amanpreet, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach (2019). "Towards vqa models that can read". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8317–8326 (cit. on p. 105).

Sivic, Josef and Andrew Zisserman (2003). "Video Google: a text retrieval approach to object matching in videos". In: *Proceedings Ninth IEEE International Conference on Computer Vision*, 1470–1477 vol.2 (cit. on p. 8).

Soudry, Daniel, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro (2018). "The implicit bias of gradient descent on separable data". In: *The Journal of Machine Learning Research* 19.1, pp. 2822–2878 (cit. on p. 22).

Stock, Pierre and Moustapha Cisse (Sept. 2018a). "ConvNets and ImageNet Beyond Accuracy: Understanding Mistakes and Uncovering Biases". In: *The European Conference on Computer Vision (ECCV)* (cit. on p. 21).

Stock, Pierre and Moustapha Cisse (2018b). "ConvNets and ImageNet Beyond Accuracy: Understanding Mistakes and Uncovering Biases". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on p. 24).

Sun, Yiyou, Yifei Ming, Xiaojin Zhu, and Yixuan Li (2022a). "Out-of-distribution Detection with Deep Nearest Neighbors". In: *arXiv preprint arXiv:2204.06507* (cit. on p. 106).

Sun, Yiyou, Yifei Ming, Xiaojin Zhu, and Yixuan Li (2022b). "Out-of-distribution Detection with Deep Nearest Neighbors". In: *arXiv preprint arXiv:2204.06507* (cit. on pp. 121, 160).

Tan, Hao and Mohit Bansal (2019). "Lxmert: Learning cross-modality encoder representations from transformers". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (cit. on pp. 19, 65, 68).

Teney, Damien, Ehsan Abbasnedjad, and Anton van den Hengel (2020a). "Learning what makes a difference from counterfactual examples and gradient su-

pervision". In: *European Conference on Computer Vision*. Springer, pp. 580–599 (cit. on pp. 28, 53, 126).

Teney, Damien, Peter Anderson, Xiaodong He, and Anton Van Den Hengel (2018). "Tips and tricks for visual question answering: Learnings from the 2017 challenge". In: *CVPR* (cit. on p. 111).

Teney, Damien, Kushal Kafle, Robik Shrestha, Ehsan Abbasnejad, Christopher Kanan, and Anton van den Hengel (2020b). "On the Value of Out-of-Distribution Testing: An Example of Goodhart's Law". In: *arXiv preprint arXiv:2005.09241* (cit. on pp. 28, 62, 82, 96, 98, 156, 157, 175–177).

Thomason, Jesse, Daniel Gordon, and Yonatan Bisk (2019). "Shifting the Baseline: Single Modality Performance on Visual Navigation & QA". In: *NACL* (cit. on p. 23).

Tian, Jing, Michael H Azarian, and Michael Pecht (2014). "Anomaly detection using self-organizing maps-based k-nearest neighbor algorithm". In: *PHM Society European Conference*. Vol. 2. 1 (cit. on p. 106).

Torralba, Antonio and Alexei A. Efros (June 2011a). "Unbiased look at dataset bias". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. URL: http://dx.doi.org/10.1109/CVPR.2011.5995347 (cit. on p. 21).

Torralba, Antonio and Alexei A. Efros (2011b). "Unbiased look at dataset bias". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 21).

Touvron, Hugo, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou (2021). "Training data-efficient image transformers & distillation through attention". In: *International Conference on Machine Learning*. PMLR, pp. 10347–10357 (cit. on p. 10).

Tran, Dustin, Jeremiah Liu, Michael W. Dusenberry, Du Phan, Mark Collier, Jie Ren, Kehang Han, Zi Wang, Zelda Mariet, Huiyi Hu, Neil Band, Tim G. J. Rudner, Karan Singhal, Zachary Nado, Joost van Amersfoort, Andreas Kirsch, Rodolphe Jenatton, Nithum Thain, Honglin Yuan, Kelly Buchanan, Kevin Murphy, D. Sculley, Yarin Gal, Zoubin Ghahramani, Jasper Snoek, and Balaji Lakshminarayanan (2022). *Plex: Towards Reliability using Pretrained Large Model Extensions*. URL: https://arxiv.org/abs/2207.07411 (cit. on p. 104).

Trott, Alexander, Caiming Xiong, and Richard Socher (2018). "Interpretable counting for visual question answering". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on pp. 33, 59, 60, 66–68, 73).

Turian, Joseph, Lev Ratinov, and Yoshua Bengio (2010). "Word representations: a simple and general method for semi-supervised learning". In: *Proceedings of the 48th annual meeting of the association for computational linguistics*, pp. 384–394 (cit. on p. 9).

Uno, T, T Asai, Y Uchida, and H Arimura (2003). "An efficient algorithm for enumerating frequent closed item sets". In: *Proceedings of Frequent Itemset Mining Implementations* 90 (cit. on p. 82).

Valle-Perez, Guillermo, Chico Q Camargo, and Ard A Louis (2018). "Deep learning generalizes because the parameter-function map is biased towards simple functions". In: *arXiv preprint arXiv:1805.08522* (cit. on p. 22).

Vapnik, Vladimir (1999). "An overview of statistical learning theory". In: *IEEE transactions on neural networks* (cit. on p. 7).

Varshney, Neeraj, Swaroop Mishra, and Chitta Baral (2022). "Investigating Selective Prediction Approaches Across Several Tasks in IID, OOD, and Adversarial Settings". In: *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 1995–2002. URL: https://aclanthology.org/2022.findings-acl.158 (cit. on pp. 104, 106).

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on pp. 9, 10, 18, 60, 67, 112).

Vries, Harm de, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron C. Courville (2017). "GuessWhat?! Visual object discovery through multi-modal dialogue". In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 1).

Wang, Haoqi, Zhizhong Li, Litong Feng, and Wayne Zhang (2022). "ViM: Out-Of-Distribution with Virtual-logit Matching". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4921–4930 (cit. on p. 106).

Wang, Haoran, Weitang Liu, Alex Bocchieri, and Yixuan Li (2021). "Can multi-label classification networks know what they don't know?" In: *Advances in Neural Information Processing Systems* 34, pp. 29074–29087 (cit. on p. 106).

Wang, Peng, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang (2022). "Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework". In: *International Conference on Machine Learning*. PMLR, pp. 23318–23340 (cit. on pp. 19, 20, 103, 111, 112, 127, 166, 172).

Wang, Wenhui, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. (2022). "Image as a foreign language: Beit pretraining for all vision and vision-language tasks". In: *arXiv preprint arXiv:2208.10442* (cit. on p. 20).

Wang, Zirui, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao (2022). "Simvlm: Simple visual language model pretraining with weak supervision". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on p. 20).

Weizenbaum, Joseph (1966). "ELIZA—a computer program for the study of natural language communication between man and machine". In: *Communications of the ACM* 9.1, pp. 36–45 (cit. on p. 9).

Whitehead, Spencer, Suzanne Petryk, Vedaad Shakib, Joseph Gonzalez, Trevor Darrell, Anna Rohrbach, and Marcus Rohrbach (2022a). "Reliable Visual Question Answering: Abstain Rather Than Answer Incorrectly". In: *ECCV* (cit. on p. 33).

Whitehead, Spencer, Suzanne Petryk, Vedaad Shakib, Joseph Gonzalez, Trevor Darrell, Anna Rohrbach, and Marcus Rohrbach (2022b). "Reliable Visual Question Answering: Abstain Rather Than Answer Incorrectly". In: *arXiv:2204.13631v3* (cit. on pp. 103, 104, 106–109, 111–115, 117, 159, 175).

Whitehead, Spencer, Hui Wu, Heng Ji, Rogerio Feris, and Kate Saenko (2021). "Separating skills and concepts for novel visual question answering". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5632–5641 (cit. on p. 113).

Witten, Ian H, Zane Bray, Malika Mahoui, and Bill Teahan (1999). "Text mining: A new frontier for lossless compression". In: *Proceedings DCC'99 Data Compression Conference (Cat. No. PR000096)*. IEEE, pp. 198–207 (cit. on p. 9).

Wolpert, David H (1992). "Stacked generalization". In: *Neural networks* 5.2, pp. 241–259 (cit. on p. 110).

Yang, Zichao, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola (2016). "Stacked attention networks for image question answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21–29 (cit. on pp. 17, 18, 43, 46, 96, 97, 99, 156, 157, 175–177).

El-Yaniv, Ran and Yair Wiener (2010). "On the Foundations of Noise-free Selective Classification". In: *Journal of Machine Learning Research* 11, pp. 1605–1641 (cit. on pp. 103, 107).

Yu, Zhou, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian (2019). "Deep modular co-attention networks for visual question answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6281–6290 (cit. on pp. 18–20, 111).

Yu, Zhou, Jun Yu, Jianping Fan, and Dacheng Tao (2017). "Multi-modal Factorized Bilinear Pooling with Co-Attention Learning for Visual Question Answering". In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 1839–1848 (cit. on p. 16).

Yu, Zhou, Jun Yu, Chenchao Xiang, Jianping Fan, and Dacheng Tao (2018). "Beyond Bilinear: Generalized Multi-modal Factorized High-order Pooling for Visual Question Answering". In: *IEEE Transactions on Neural Networks and Learning Systems* (cit. on p. 16).

Yuan, Junsong, Ying Wu, and Ming Yang (2007). "Discovery of collocation patterns: from visual words to visual phrases". In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1–8 (cit. on p. 83).

Zhang, Yan, Jonathon Hare, and Adam Prügel-Bennett (2018). "Learning to count objects in natural images for visual question answering". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on pp. 33, 59, 60, 65–68, 73, 74, 174).

Zhao, Jieyu, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang (2017a). "Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-level Constraints". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (cit. on p. 23).

Zhao, Jieyu, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang (2017b). "Men also like shopping: Reducing gender bias amplification using corpus-level constraints". In: *arXiv preprint arXiv:1707.09457* (cit. on p. 22).

APPENDIX A

ADDITIONAL RESULTS FOR CHAPTER 4

In this appendix, we report additional results for Chapter 4.

TallyQA-Even-Odd In Table A.1, we display the number of odd and even triplets in each set of TallyQA-Even-Odd where 90% of triplets have been removed ($p = 90\%$), and other datasets where $p = \{0, 50, 100\}$.

	Training set		Validation set		Testing set	
p%	Odd	Even	Odd	Even	Odd	Even
0 %	87,289	137,102	9,635	15,292	23,138	15,451
50 %	43,643	137,102	4,815	15,292	23,138	7,719
90 %	8,725	137,102	969	15,292	23,138	1,551
100 %	0	137,102	0	15,292	23,138	0

Table A.1. – Number of *image-question-count* triplets for each set generated by our Even-Odd-p% strategy when applied on the TallyQA dataset (Even-Odd-0% leads to the the original TallyQA distribution). Numbers of triplets for intermediate values of p can be obtained with linear interpolation.

APPENDIX B

ADDITIONAL EXPERIMENTS FOR CHAPTER 5

In this appendix, we report additional experiments for Chapter 5. We explore two variants of our evaluation benchmark: First, in Section B.1, we show results using the ground-truth visual labels instead of the detected objects. Second, in Section B.2, we share results on the VQA v1 dataset.

B.1 Results with ground-truth visual labels

We report in Table B.1 the results of our analysis with ground-truth visual labels from the COCO (T.-Y. Lin et al. 2014b) dataset, instead of labels detected with Faster R-CNN. We make similar observations to the main experiments of the paper: bias-reduction methods often degrade performances, on both easy and counterexample split. A few methods slightly improve the counterexamples score, but much less than on VQA-CP. The only method which improves both overall and counterexamples scores is LfF (Nam et al. 2020). We observed similar results on the dataset with detected labels, reported in Table 1 of the main paper.

B.2 Results on VQA v1

We report in Table B.2 the results of our analysis on the VQA v1 dataset. We observe similar results as in Table 1 from the main paper. Most bias-reduction methods degrade performances on the counterexamples split, and only LfF (Nam et al. 2020) improves performances on all three splits.

	Approaches	Overall	**Counterexamples (ours)**	**Easy (ours)**
	Number of examples	*214,354*	*63,925*	*135,324*
Baselines	Shortcuts	42.14	0.43	65.95
	Image-Only	23.70	2.92	35.39
	Question-Only	44.12	13.98	60.88
VQA models	SAN – *grid features*	55.61	28.99	70.04
	UpDown	63.52 (+0.00)	37.77 (+0.00)	77.52 (+0.00)
	BLOCK	63.89	37.06	78.52
	VilBERT – *pretrained*†	67.77	43.32	81.27
	UpDown is used as a base architecture for bias-reduction methods			
Bias-reduction methods	RUBi	61.88 (-1.64)	36.05 (-1.72)	75.84 (-1.68)
	LMH + RMFE	60.12 (-3.40)	34.97 (-2.80)	73.80 (-3.72)
	ESR	62.96 (-0.56)	37.22 (-0.55)	76.98 (-0.54)
	LMH	61.15 (-2.37)	37.82 (+0.05)	73.91 (-3.61)
	LfF	63.57 (+0.05)	38.18 (+0.41)	77.44 (-0.08)
	LMH+CSS	53.55 (-9.97)	37.27 (-0.50)	62.30 (-15.22)
	RandImg	63.34 (-0.18)	38.13 (+0.36)	77.05 (-0.47)

Table B.1. – Results of our VQA-CE evaluation protocol with **ground-truth visual labels**. We report accuracies on VQA v2 full validation set and on our two subsets: **Counterexamples** and **Easy** examples. We re-implemented all models and bias-reduction methods. †VilBERT is pre-trained on Conceptual Caption and fine-tuned on VQA v2 training set. Scores in (green) and (red) are relative to UpDown. We evaluate SAN (Yang et al. 2016), UpDown (Anderson et al. 2018a), BLOCK (Ben-Younes et al. 2019b),VilBERT (J. Lu et al. 2019), RUBi (Cadene et al. 2019c), LMH + RMFE (Gat et al. 2020), ESR (Shrestha et al. 2020), LMH (C. Clark et al. 2019), LfF (Nam et al. 2020), LMH+CSS (L. Chen et al. 2020), RandImg (Teney et al. 2020b).

	Approaches	Overall	**Counterexamples (ours)**	**Easy (ours)**
	Number of examples	*121,512*	*40,052*	*80,539*
Baselines	Shortcuts	44.71	0.05	67.35
	Image-Only	24.39	1.75	35.83
	Question-Only	49.20	13.48	67.27
	SAN – *grid features*	58.35	26.09	74.58
	UpDown	62.83 (+0.00)	31.71 (+0.00)	78.49 (+0.00)
Bias-reduction methods	*UpDown is used as a base architecture for bias-reduction methods*			
	RUBi	55.82 (-7.01)	23.87 (-7.84)	71.90 (-6.59)
	LMH + RMFE	62.97 (+0.14)	31.09 (-0.62)	79.02 (+0.53)
	ESR	63.03 (+0.20)	31.50 (-0.21)	78.91 (+0.42)
	LMH	59.74 (-3.09)	32.80 (+1.09)	73.30 (-5.19)
	LfF	63.26 (+0.43)	32.05 (+0.34)	78.97 (+0.48)
	RandImg	62.87 (+0.04)	31.09 (-0.62)	78.87 (+0.38)

Table B.2. – Results of our VQA-CE evaluation protocol on **VQA v1** full validation set and on our two subsets: **Counterexamples** and **Easy** examples. We re-implemented all models and bias-reduction methods. Scores in (green) and (red) are relative to UpDown. We evaluate SAN (Yang et al. 2016), UpDown (Anderson et al. 2018a), BLOCK (Ben-Younes et al. 2019b),VilBERT (J. Lu et al. 2019), RUBi (Cadene et al. 2019c), LMH + RMFE (Gat et al. 2020), ESR (Shrestha et al. 2020), LMH (C. Clark et al. 2019), LfF (Nam et al. 2020), LMH+CSS (L. Chen et al. 2020), RandImg (Teney et al. 2020b).

APPENDIX C

ADDITIONAL EXPERIMENTS FOR CHAPTER 7

C.1 Jointly Training OFA and Selector

Discussed in Section 6.5.1, for training Selector, we follow a staged procedure (Whitehead et al. 2022b): The VQA model is first trained until convergence on the VQA task. Then, the weights are frozen, Selector is added to the model, and Selector is learned on top of the frozen model.

Since we are able to train OFA and Selector on the same data, a natural comparison to make is between the staged training procedure we use and joint training (i.e., simultaneously optimizing the VQA model and Selector), similar to (Geifman and El-Yaniv 2019). We experiment with joint training by summing their losses. We perform this on OFA-Base, training both OFA-Base and Selector with the full A+B data. We also experiment with first joint training OFA-Base and Selector until OFA-Base has converged for the VQA task, freezing OFA-Base, and continuing to fine-tune Selector on A+B.

The results in Table C.1 illustrate that joint training decreases the overall performance of the Selector. All metrics yield worse performance with joint training alone, though the gap shrinks when freezing the VQA model and continuing to fine-tune Selector. This is even though the overall VQA accuracy remains roughly the same with or without joint training. We conjecture that the reason for this may be that joint training creates a somewhat non-stationary optimization problem for Selector. Specifically, the VQA model's representations and VQA accuracy are changing throughout training. This means that the statistics of the inputs and training targets for Selector (see Section 6.5.1) are changing, which may make optimizing Selector more difficult. Other techniques may be needed in order to properly optimize the VQA model and Selector together.

Training	Acc	$\mathcal{C}$@1%	$\mathcal{C}$@5%	$\mathcal{C}$@10%	AUC
ID (100% VQA v2)					
joint	75.08	16.04	42.78	65.91	8.11
joint+FT	75.08	24.42	50.01	69.20	7.21
staged	75.18	**26.64**	**50.80**	**69.56**	**7.10**
90% VQA v2, 10% AdVQA					
joint	71.97	10.74	34.61	53.81	10.12
joint+FT	71.97	18.17	42.44	60.50	8.98
staged	72.00	**19.72**	**42.70**	**60.84**	**8.90**

Table C.1. – Comparison of joint and staged training of OFA-Base and Selector. FT indicates that Selector is further fine-tuned after OFA-Base converges on the VQA training objective. All models are trained on A+B.

C.2 OOD Detection features

Inspired by Fisch et al. 2022, we train Selector with out-of-distribution detection scores computed with KNN (Sun et al. 2022b) or SSD (Sehwag et al. 2021) as added features. We share the results in Appendix C, Section C.2.

To compute those metrics, we use the representations from the encoder of OFA. We average the output question tokens q_i and the image tokens v_i, which respectively yield $\bar{q}$ and $\bar{v}$. We compute OOD detection features for each representation with respect to the training data. The computed features are as follows:

kNN (Sun et al. 2022b). Given an input example, we compute the cosine distance to the k nearest neighbors in the training data. This distance is used as an OOD score: higher scores signify more "in-distribution" examples, while lower scores signify "out-of-distribution". We use the efficient vector-search library FAISS (Jeff Johnson et al. 2019) to compute the distances and identify the k closest points. We experimented with various numbers of neighbors from 1 to 1000 and found no significant improvements for any value. We also experimented with using the distance to *correct* and *incorrect* neighbors, to align the distances to our task of selective prediction.

SSD (Sehwag et al. 2021). SSD is a parametric OOD-detection method that first builds k clusters in feature space and then fits a multivariate normal distribution for each of the k ensembles of features. For a new example, the Mahalanobis distance (K. Lee et al. 2018) to this normal distribution is used as an OOD score. Note that for a classification task, the labels might be used as clusters, but we prefer to use a cluster-based algorithm, as the VQA answers do not represent a

coherent ensemble of image or question concepts. We experimented with various numbers of clusters in the range of $[1, 1000]$, and saw no improvements.

Results For these OOD detection features, we give them as additional inputs to the Selector to provide a signal for whether a given example is ID or OOD. We find that these features do not bring significant improvements to our evaluation metrics (Table C.2).

	Acc.	$\mathcal{C}$@1%	$\mathcal{C}$@5%	$\mathcal{C}$@10%	AUC
Selector	71.25	19.05	41.83	59.55	9.29
Selector + KNN	71.25	19.92	41.78	59.75	9.27
Selector + SSD	71.25	18.99	41.90	59.27	9.27

Table C.2. – OOD-Detection baselines. Scores are reported on the Mixed ID/OOD, composed of 90% VQA and 10% AdVQA.

C.3 Augmenting Selector training with known OOD data

As discussed in Section 6.3.3, we also try training Selector on the B set, along with some known OOD datasets similar to Kamath et al. 2020. This may help learn to discard hard examples which are very far from its training distribution. For this experiment, we use the training sets of OK-VQA (Marino et al. 2019), which has the same image distribution but a different question distribution, and of VizWiz (Gurari et al. 2018), which has both image and question distribution shifts compared to VQA v2. We see in Table C.3 that this method is not very successful at improving reliability in our adversarial evaluation setting. Contrary to the findings of Kamath et al. 2020 for text-only question answering, on our Selective VQA task, adding this known OOD data during training decreases the performance of our selector on unknown OOD data at test time. Overall, it appears that more traditional approaches for handling OOD examples may have difficulty generalizing to this multimodal setting.

C.4 Additional OOD experiments

In this section, we share additional results on other mixtures of ID + out-of-distribution (OOD) AdVQA data. Table C.4, Table C.5, and Table C.6 respectively show the results for 33%, 50%, and 66% AdVQA, respectively. We also show the results with threshold selection on the in-distribution validation set in Table C.7

Training Set f	Selector g	$\mathcal{C}$@1%	$\mathcal{C}$@5%	$\mathcal{C}$@10%	AUC
		90% VQA v2, 10% AdVQA			
A	B	**19.00**	41.64	58.97	9.34
A	B + OK-VQA	18.38	**42.33**	**59.80**	**9.17**
A	B + OK-VQA + VizWiz	18.48	41.08	59.40	9.36
		50% VQA v2, 50% AdVQA			
A	B	**2.68**	**15.98**	26.72	18.97
A	B + OK-VQA	1.73	15.37	26.33	**18.86**
A	B + OK-VQA + VizWiz	2.56	14.93	**26.82**	19.08

Table C.3. – Results with exposure to known OOD examples for OFA-Base. OOD = OK-VQA + VizWiz. **Bold** denotes best and underline is second best per table section.

for the in-distribution testing set, and Table C.8 for the 10% AdVQA data testing set.

VQA Model f		Selection function g			Acc ↑	$\mathcal{C}$@$\mathcal{R}$ in % ↑			AUC ↓	Φ_1	Φ_{10}	Φ_{100}
Name	Training set	Name	Training Set	Targets		1%	5%	10%				
CLIP-ViL	A	MaxProb	-	-	58.36	0.00	7.08	21.97	20.62	36.59	-1.47	-14.38
		Selector	B	Self	58.36	5.87	17.41	29.21	18.90	38.76	7.11	-2.20
	A+B	MaxProb	-	-	59.29	1.11	10.17	24.99	19.58	38.42	2.99	-9.79
		Selector	A+B	Self	59.29	0.07	11.21	25.86	19.28	39.17	5.90	-7.37
		Selector	A+B	LYP	59.29	7.07	19.13	31.53	17.94	39.85	12.67	3.40
OFA Base	A	MaxProb	-	-	64.08	0.01	18.83	34.15	15.71	46.05	5.33	-28.66
		Selector	B	Self	64.08	3.59	26.29	39.78	14.54	46.77	13.58	-10.18
	A+B	MaxProb	-	-	64.63	0.03	17.57	33.94	15.43	46.32	2.11	-19.21
		Selector	A+B	Self	64.63	5.11	25.83	40.13	14.09	47.58	10.75	-21.18
		Selector	A+B	LYP	64.63	9.41	27.89	42.0	13.80	48.03	11.89	-2.81
OFA Large	A	MaxProb	-	-	67.57	0.01	18.90	41.23	13.62	50.48	10.59	-24.77
		Selector	B	Self	67.56	11.50	30.24	49.03	11.93	52.36	18.02	-8.92
	A+B	MaxProb	-	-	67.78	0.03	19.92	42.32	13.47	50.81	7.41	-37.62
		Selector	A+B	Self	67.77	5.39	29.41	48.93	12.08	51.53	15.37	-16.32
		Selector	A+B	LYP	67.77	10.94	31.32	50.11	11.75	52.44	18.18	-11.46

Table C.4. – Results on a mixed ID/OOD setting, composed of 66.7% VQA v2 data (Test split in Table 6.2) and 33.3% AdVQA examples. Discussion in Section 6.5.4.

VQA Model f		Selection function g			Acc ↑	$\mathcal{C}@\mathcal{R}$ in % ↑			AUC ↓	Φ_1	Φ_{10}	Φ_{100}
Name	Training set	Name	Training Set	Targets		1%	5%	10%				
CLIP-ViL	A	MaxProb	-	-	52.66	0.00	3.08	9.77	26.57	27.65	-13.20	-20.60
		Selector	B	Self	52.66	4.19	10.29	18.17	24.49	30.62	-2.24	-7.94
	A+B	MaxProb	-	-	53.83	0.97	3.66	12.27	25.23	29.82	-6.40	-15.61
		Selector	A+B	Self	53.83	0.04	5.52	13.38	24.96	30.82	-2.96	-11.50
		Selector	A+B	LYP	53.83	3.41	11.19	20.42	23.22	31.85	5.49	-0.04
OFA Base	A	MaxProb	-	-	59.17	0.01	5.78	18.48	20.80	38.14	-6.11	-29.98
		Selector	B	Self	59.18	2.68	15.98	26.72	18.97	38.99	0.58	-20.48
	A+B	MaxProb	-	-	59.61	0.06	6.91	20.86	20.17	38.45	-12.19	-31.48
		Selector	A+B	Self	59.62	2.29	15.78	27.38	18.70	39.88	-0.74	-36.10
		Selector	A+B	LYP	59.62	3.98	17.13	28.53	18.30	40.35	-0.49	-11.27
OFA Large	A	MaxProb	-	-	63.02	0.31	11.53	27.85	17.18	43.42	-3.11	-34.01
		Selector	B	Self	63.01	5.56	20.11	35.51	15.52	45.49	6.48	-26.57
	A+B	MaxProb	-	-	62.93	0.12	6.22	26.58	17.58	43.40	-6.02	-40.93
		Selector	A+B	Self	62.93	1.00	18.55	33.48	16.03	44.14	2.57	-43.03
		Selector	A+B	LYP	62.93	3.51	19.74	34.18	15.78	45.03	4.19	-29.46

Table C.5. – Results on a mixed ID/OOD setting, composed of 50% VQA v2 data (Test split in Table 6.2) and 50% AdVQA examples. Discussion in Section 6.5.4.

VQA Model f		Selection function g			Acc ↑	$\mathcal{C}@\mathcal{R}$ in % ↑			AUC ↓	Φ_1	Φ_{10}	Φ_{100}
Name	Training set	Name	Training Set	Targets		1%	5%	10%				
CLIP-ViL	A	MaxProb	-	-	46.66	0.00	0.00	3.04	33.67	18.32	-24.68	-28.56
		Selector	B	Self	46.66	1.91	5.65	10.09	31.43	22.00	-11.50	-12.05
	A+B	MaxProb	-	-	47.94	0.67	1.28	5.59	32.08	20.87	-16.85	-21.99
		Selector	A+B	Self	47.94	0.05	1.44	5.49	31.79	22.20	-11.69	-15.28
		Selector	A+B	LYP	47.94	2.13	6.60	10.44	29.77	23.60	-0.77	-0.89
OFA Base	A	MaxProb	-	-	53.71	0.00	0.45	8.44	26.47	29.64	-17.60	-43.15
		Selector	B	Self	53.77	1.99	8.24	16.43	24.49	30.75	-9.75	-30.58
	A+B	MaxProb	-	-	54.28	0.03	0.53	10.16	25.72	29.96	-25.56	-44.75
		Selector	A+B	Self	54.26	1.52	8.79	16.23	24.15	31.88	-12.68	-52.56
		Selector	A+B	LYP	54.26	1.95	9.71	17.11	23.79	32.38	-12.12	-20.65
OFA Large	A	MaxProb	-	-	57.69	0.13	3.65	14.24	22.36	34.91	-16.36	-49.70
		Selector	B	Self	57.71	3.03	11.20	22.04	20.44	37.45	-5.27	-39.01
	A+B	MaxProb	-	-	57.52	0.08	0.54	13.41	22.87	34.70	-20.37	-56.21
		Selector	A+B	Self	57.5	0.46	9.02	20.14	21.10	35.39	-10.72	-61.53
		Selector	A+B	LYP	57.5	0.08	10.28	19.93	20.94	36.60	-8.58	-44.52

Table C.6. – Results on a mixed ID/OOD setting, composed of 33.3% VQA v2 data (Test split in Table 6.2) and 66.7% AdVQA examples. Discussion in Section 6.5.4.

VQA Model f		Selection function g			Acc ↑	$\mathcal{R} = 1\%$		$\mathcal{R} = 5\%$		$\mathcal{R} = 10\%$	
Name	Training set	Name	Training Set	Targets		$\mathcal{R}$	$\mathcal{C}$	$\mathcal{R}$	$\mathcal{C}$	$\mathcal{R}$	$\mathcal{C}$
CLIP-ViL	A	MaxProb	-	-	69.98	0.86	3.49	4.55	31.59	9.60	52.35
		Selector	B	Self	69.98	0.72	13.26	4.74	36.66	9.97	55.58
	A+B	MaxProb	-	-	70.72	1.08	6.67	4.59	32.85	9.83	54.47
		Selector	A+B	Self	70.72	1.10	7.60	4.78	34.16	9.73	54.63
		Selector	A+B	LYP	70.72	0.85	16.78	4.96	38.30	10.08	57.34
OFA Base	A	MaxProb	-	-	74.87	1.18	5.32	4.96	45.45	9.96	66.44
		Selector	B	Self	74.87	1.06	25.73	4.92	49.47	10.11	69.52
	A+B	MaxProb	-	-	75.18	0.82	4.32	4.98	46.03	10.08	67.88
		Selector	A+B	Self	75.18	1.14	27.88	5.23	51.76	10.09	69.87
		Selector	A+B	LYP	75.18	1.00	27.84	5.17	52.44	10.35	71.31
OFA Large	A	MaxProb	-	-	77.77	1.20	8.64	4.74	52.06	9.77	74.05
		Selector	B	Self	77.33	0.89	28.27	4.89	57.02	10.12	76.45
	A+B	MaxProb	-	-	77.78	1.12	19.26	4.83	52.69	9.94	74.82
		Selector	A+B	Self	77.78	1.12	19.26	4.83	52.69	9.94	74.82
		Selector	A+B	LYP	77.77	1.02	31.86	5.10	59.24	9.97	77.24

Table C.7. – Results on the ID VQA v2 evaluation set (Test split in Table 6.2). Thresholds for desired risk level are selected on the in-distribution Val split. Discussion in Section 6.5.5.

VQA Model f		Selection function g			Acc ↑	$\mathcal{R} = 1\%$		$\mathcal{R} = 5\%$		$\mathcal{R} = 10\%$	
Name	Training set	Name	Training Set	Targets		$\mathcal{R}$	$\mathcal{C}$	$\mathcal{R}$	$\mathcal{C}$	$\mathcal{R}$	$\mathcal{C}$
CLIP-ViL	A	MaxProb	-	-	66.35	1.83	3.21	6.25	29.53	12.05	50.06
		Selector	B	Self	66.35	0.95	12.14	5.75	33.92	11.78	52.33
	A+B	MaxProb	-	-	67.12	1.59	6.11	5.97	30.70	12.02	52.14
		Selector	A+B	Self	67.12	1.52	6.97	6.04	31.95	11.63	51.83
		Selector	A+B	LYP	67.12	1.14	15.26	5.81	35.46	11.72	54.08
OFA Base	A	MaxProb	-	-	71.59	1.69	4.88	6.54	43.00	12.11	64.13
		Selector	B	Self	71.60	1.47	23.68	6.00	46.13	12.01	66.51
	A+B	MaxProb	-	-	72.00	1.30	3.95	6.56	43.59	12.05	65.67
		Selector	A+B	Self	72.02	1.60	25.72	6.49	48.75	11.82	67.13
		Selector	A+B	LYP	72.01	1.38	25.61	6.27	48.97	12.07	68.25
OFA Large	A	MaxProb	-	-	74.36	1.74	15.43	6.61	49.83	11.96	71.98
		Selector	B	Self	74.37	1.32	25.86	6.03	53.88	11.93	74.15
	A+B	MaxProb	-	-	74.77	1.92	7.98	6.34	49.94	11.89	72.50
		Selector	A+B	Self	74.77	1.48	27.51	6.20	55.21	11.93	74.92
		Selector	A+B	LYP	74.77	1.42	29.26	6.27	56.01	11.81	75.14

Table C.8. – Results on the mixed 90% VQA v2 + 10% AdVQA evaluation set (VQA v2 data is from the Test split in Table 6.2). Thresholds for desired risk level are selected on our in-distribution Val set. Discussion in Section 6.5.5.

LIST OF FIGURES

LIST OF TABLES