Bias and reasoning in visual question answering

Corentin Kervadec

THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE LYON
opérée au sein de
INSA LYON

École Doctorale 512
Informatique et Mathématique de Lyon
(INFOMATHS)

Spécialité
Informatique

Présentée par

Corentin Kervadec

Pour obtenir le grade de
DOCTEUR de L'UNIVERSITÉ DE LYON

Sujet de la thèse :

Bias and Reasoning in Visual Question Answering

Biais et raisonnement dans les systèmes de questions réponses visuelles

Soutenue publiquement le 9 Décembre 2021, devant le jury composé de :

M. David Picard	École des Ponts ParisTech	Rapporteur – Président
M. Nicolas Thome	CNAM	Rapporteur
Mme. Cordelia Schmid	INRIA - Google	Examinatrice
M. Damien Teney	IDIAP	Examinateur
Mme. Akata Zeynep	University of Tübingen	Examinatrice
M. Christian Wolf	INSA Lyon - LIRIS	Directeur de thèse
M. Grigory Antipov	Orange Innovation	Co-encadrant de thèse
M. Moez Baccouche	Orange Innovation	Co-encadrant de thèse

N° d'ordre NNT : 2021LYSEI101

Département FEDORA – INSA Lyon - Ecoles Doctorales

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	**CHIMIE DE LYON** **https://www.edchimie-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage secretariat@edchimie-lyon.fr	**M. Stéphane DANIELE** C2P2-CPE LYON-UMR 5265 Bâtiment F308, BP 2077 43 Boulevard du 11 novembre 1918 69616 Villeurbanne **directeur@edchimie-lyon.fr**
E.E.A.	**ÉLECTRONIQUE, ÉLECTROTECHNIQUE, AUTOMATIQUE** **https://edeea.universite-lyon.fr** Sec. : Stéphanie CAUVIN Bâtiment Direction INSA Lyon Tél : 04.72.43.71.70 secretariat.edeea@insa-lyon.fr	**M. Philippe DELACHARTRE** INSA LYON Laboratoire CREATIS Bâtiment Blaise Pascal, 7 avenue Jean Capelle 69621 Villeurbanne CEDEX Tél : 04.72.43.88.63 **philippe.delachartre@insa-lyon.fr**
E2M2	**ÉVOLUTION, ÉCOSYSTÈME, MICROBIOLOGIE, MODÉLISATION** **http://e2m2.universite-lyon.fr** Sec. : Sylvie ROBERJOT Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.e2m2@univ-lyon1.fr	**M. Philippe NORMAND** Université Claude Bernard Lyon 1 UMR 5557 Lab. d'Ecologie Microbienne Bâtiment Mendel 43, boulevard du 11 Novembre 1918 69 622 Villeurbanne CEDEX **philippe.normand@univ-lyon1.fr**
EDISS	**INTERDISCIPLINAIRE SCIENCES-SANTÉ** **http://ediss.universite-lyon.fr** Sec. : Sylvie ROBERJOT Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.ediss@univ-lyon1.fr	**Mme Sylvie RICARD-BLUM** Institut de Chimie et Biochimie Moléculaires et Supramoléculaires (ICBMS) - UMR 5246 CNRS - Université Lyon 1 Bâtiment Raulin - 2ème étage Nord 43 Boulevard du 11 novembre 1918 69622 Villeurbanne Cedex Tél : +33(0)4 72 44 82 32 **sylvie.ricard-blum@univ-lyon1.fr**
INFOMATHS	**INFORMATIQUE ET MATHÉMATIQUES** **http://edinfomaths.universite-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage Tél : 04.72.43.80.46 infomaths@univ-lyon1.fr	**M. Hamamache KHEDDOUCI** Université Claude Bernard Lyon 1 Bât. Nautibus 43, Boulevard du 11 novembre 1918 69 622 Villeurbanne Cedex France Tél : 04.72.44.83.69 **hamamache.kheddouci@univ-lyon1.fr**
Matériaux	**MATÉRIAUX DE LYON** **http://ed34.universite-lyon.fr** Sec. : Yann DE ORDENANA Tél : 04.72.18.62.44 yann.de-ordenana@ec-lyon.fr	**M. Stéphane BENAYOUN** Ecole Centrale de Lyon Laboratoire LTDS 36 avenue Guy de Collongue 69134 Ecully CEDEX Tél : 04.72.18.64.37 **stephane.benayoun@ec-lyon.fr**
MEGA	**MÉCANIQUE, ÉNERGÉTIQUE, GÉNIE CIVIL, ACOUSTIQUE** **http://edmega.universite-lyon.fr** Sec. : Stéphanie CAUVIN Tél : 04.72.43.71.70 Bâtiment Direction INSA Lyon mega@insa-lyon.fr	**M. Jocelyn BONJOUR** INSA Lyon Laboratoire CETHIL Bâtiment Sadi-Carnot 9, rue de la Physique 69621 Villeurbanne CEDEX **jocelyn.bonjour@insa-lyon.fr**
ScSo	**ScSo** **https://edsciencessociales.universite-lyon.fr** Sec. : Mélina FAVETON INSA : J.Y. TOUSSAINT Tél : 04.78.69.77.79 melina.faveton@univ-lyon2.fr	**M. Christian MONTES** Université Lumière Lyon 2 86 Rue Pasteur 69365 Lyon CEDEX 07 **christian.montes@univ-lyon2.fr**

Roses are red,
Violets are blue...
But should VQA expect them to?

ABSTRACT

This thesis addresses the Visual Question Answering (VQA) task through the prism of biases and reasoning. VQA is a visual reasoning task where a model is asked to automatically answer questions posed over images. Despite impressive improvement made by deep learning approaches, VQA models are notorious for their tendency to rely on dataset biases. The large and unbalanced diversity of questions and concepts involved in the task, and the lack of well-annotated data, tend to prevent deep learning models from learning to "*reason*". Instead, it leads them to perform "*shortcuts*", relying on specific training set statistics, which is not helpful for generalizing to real-world scenarios.

Because the root of this generalization curse is first and foremost a task definition problem, our first objective is to rethink the evaluation of VQA models. Questions and concepts being unequally distributed, the standard VQA evaluation metric, consisting in measuring the overall in-domain accuracy, tends to favour models which exploit subtle training set statistics. If the model predicts the correct answer of a question, is it necessarily reasoning? Can we detect when the model prediction is right for the right reason? And, at the opposite, can we identify when the model is "*cheating*" by using statistical shortcuts? We overcome these concerns by introducing the GQA-OOD benchmark: we measure and compare accuracy over both rare and frequent question-answer pairs, and argue that the former is better suited to evaluate the reasoning abilities. We experimentally demonstrate that VQA models, including bias reduction methods, dramatically fail in this setting.

Evaluating models on benchmarks is important but not sufficient, it only gives an incomplete understanding of their capabilities. We conduct a deep analysis of a state-of-the-art Transformer VQA architecture, by studying its internal attention mechanisms. Our experiments provide evidence of the existence of operating reasoning patterns, at work in the model's attention layers, when the training conditions are favourable enough. More precisely, they appear when the visual representation is perfect, suggesting that uncertainty in vision is a dominating factor preventing the learning of reasoning. By collaborating with the data visualization experts, we have participated in the design of VisQA, a visual analytics tool exploring the question of reasoning vs shortcuts in VQA.

Finally, drawing conclusion from our evaluations and analyses, we come up with methods for improving VQA model performances. First, we propose to directly supervise the reasoning through a proxy loss measuring the fine-grained word-object alignment. We demonstrate, both experimentally and theoretically, the benefit of such reasoning supervision. Second, we explore the transfer of reasoning patterns learned by a visual oracle, trained with perfect visual input, to a standard VQA model with imperfect visual representation. Experiments show the transfer improves generalization and allows decreasing the dependency on dataset biases. Furthermore, we demonstrate that the reasoning supervision can be used as a catalyst for transferring the reasoning patterns.

RÉSUMÉ

"De quelle couleur est le terrain de tennis ? Quelle est la taille du chien ? Y a-t-il une voiture à droite du vélo sous le cocotier ?" Répondre à ces questions fondamentales est le sujet de la tâche appelée *question-réponses visuelle* (VQA, en anglais), dans laquelle un agent doit répondre à des questions posées sur des images.

CONTEXTE ET MOTIVATIONS

Plus précisément, le VQA requiert de mettre au point un agent capable de maitriser une grande variété de compétences : reconnaître des objets, reconnaitre des attributs (couleur, taille, matériaux, etc.), identifier des relations (*e.g.* spatiales), déduire des enchainements logiques, etc. C'est pourquoi, le VQA est parfois désigné comme un test de Turing visuel (GEMAN et al. 2015), dont le but est d'évaluer la capacité d'un agent à raisonner sur des images. Cette tâche a récemment connu d'important progrès grâce à l'utilisation des réseaux de neurones et de l'apprentissage profond (GOODFELLOW et al. 2016).

Après une revue détaillée de l'État de l'Art sur le VQA, ainsi qu'une définition de notre utilisation du terme *raisonnement* (Partie I), nous nous intéressons à la question suivante (Partie II) : *les modèles de VQA actuels raisonnent-ils vraiment ?* La mise en œuvre d'une nouvelle méthode d'évaluation (GQA-OOD) nous permettra de répondre négativement à cette question. En particulier, nous mettrons en évidence la tendance des modèles à apprendre des raccourcis (GEIRHOS et al. 2020), autrement appelés *biais*, présent dans les données d'entrainement, mais heurtant les capacités de généralisation. Nous proposerons alors, dans une troisième partie (Partie III) une analyse approfondie des mécanismes d'attention appris par les réseaux de neurones artificiels. Nous étudierons quels sont les enchainements aboutissant à un raisonnement, ou, au contraire, à une prédiction biaisée par un raccourci frauduleux. La dernière et quatrième partie (Partie IV) tire conclusion de nos évaluations et analyses, afin de développer de nouvelles méthodes améliorant les performances des modèles de VQA.

RÉSUMÉ DES CONTRIBUTIONS

Les contributions sont divisées en trois grandes parties "*Évaluer, Analyser, Améliorer*" :

ÉVALUER (PARTIE II) Nous proposons une nouvelle méthode d'évaluation – appelée GQA-OOD – permettant de mieux appréhender les capacités de raisonnement des systèmes de VQA. En particulier, nous mesurons le taux de bonnes réponses prédites par l'agent en fonction de la rareté de la réponse dans les données d'entrainement. Notre étude expérimentale montre que les systèmes de l'État-de-l'Art, incluant les méthodes spécifiquement conçues pour réduire l'impact des biais, échouent à répondre aux questions dont

la réponse est rare. Ce résultat mets en exergue la tendance des modèles à apprendre des biais dans les données d'entrainement, au lieu de raisonner.

ANALYSER (PARTIE III) Dans le but de compléter notre évaluation du biais et du raisonnement dans les systèmes de VQA, nous conduisons une analyse poussée des mécanismes d'attention appris par les modèles. Plus précisément, nous dressons une étude détaillée des cartes d'attention apprises par des modèles basés sur une architecture Transformers (VASWANI et al. 2017). Dans ce contexte, nous présentons VISQA, un outil de visualisation interactif, dont nous avons participé à la conception, en collaboration avec Théo Jaunet. De plus, nous mettons en œuvre une analyse statistique de ces mêmes cartes d'attention, afin de mettre en évidence l'existence de patterns de raisonnement émergeant durant l'apprentissage, lorsque les données visuelles sont parfaites.

AMÉLIORER (PARTIE IV) Enfin, nous exploitons les résultats de nos analyses et évaluations et mettons au point plusieurs méthodes améliorant les performances des systèmes de VQA. Dans un premier temps, nous montrons qu'il est possible de directement superviser le raisonnement durant l'apprentissage, au moyen d'une utilisation judicieuse des annotations de nos jeux de données, et que cela permets d'améliorer le taux de bonne prédiction de nos modèles. Dans un second temps, nous concevons une méthode permettant de transférer les patterns de raisonnement appris lorsque les conditions d'entrainement sont favorables (données visuelles parfaites), vers un modèle traitant des données réalistes, mais bruitées. Nous montrons que ce transfert améliore les performances sur le VQA, et qu'il est complémentaire avec la méthode de supervision précédemment présentée.

*
* *

En conclusion, cette thèse a pour objet l'étude du raisonnement visuel dans des réseaux de neurones artificiels entrainés par apprentissage profond, dans le cadre du VQA. Mais surtout, ce qui nous intéressera en premier lieu, c'est l'évaluation et l'analyse de l'influence qu'ont les biais, présents dans les données d'apprentissage, sur les prédictions de nos modèles. Ce sujet de recherche pourra se résumer par ces quelques vers détournés d'une comptine anglaise :

Roses are red,
Violets are blue...
But should VQA expect them to ?

REMERCIEMENTS

Maintenant, il paraît que je suis Docteur en IA ! Mais, ce que je vais retenir de ma thèse, c'est avant tout toutes les personnes que j'ai pu rencontrer et apprécier, et qui m'ont accompagné durant ce périple. Cette page ne suffira pas à leur rendre hommage, mais elle contribuera, je l'espère, à exprimer la gratitude que je ressens à leur égard.

Evidemment, je commence par remercier mes encadrants. Merci Christian. À chaque fois, je m'émerveille de voir ton engagement et ta passion pour la recherche. Mais surtout, c'est ta bienveillance sans faille que je retiendrais. Merci Moez. Ta capacité à voir le positif, même quand on est au fond du trou, m'étonnera toujours. Et pourtant, la plupart du temps, tu as raison. Merci Grigory. Je fus ton premier doctorant, mais j'espère que je ne serai pas le dernier, pour que tu puisses faire profiter à d'autres ton encadrement exceptionnel. Je sais que je peux toujours compter sur toi, aussi bien pour savoir comment changer des couches de réseau de neurones que pour changer des couches de bébés.

Réussir une thèse requiert d'être bien entouré : par chance, je l'ai été. Merci à l'équipe MAS d'Orange Innovation qui m'a accueillie en son sein. Merci Khaoula, Olivier L., Stéphane, Olivier Z., Nicolas, Michel. Claudia, Benoît, Patrice, Pierrick, Emmanuel, et tous les autres. Un merci spécial à Valentin, tu as été un modèle pour moi, depuis mes premiers pas de stagiaire jusqu'au jour de ma soutenance. Je pense également aux doctorants, ex-doctorants et jeunes chercheurs du LIRIS avec qui j'ai partagé le quotidien durant de (trop) courts séjours dans leurs magnifiques préfabriqués de l'INSA Lyon. Merci Edward, Fabien, Quentin D., Quentin P., Assem, Eric, Guillaume et Steeven. En particulier, merci Pierre et Théo J. avec qui j'ai eu l'immense plaisir de collaborer. Enfin, merci aux GPUs qui ont parfois été mes seuls amis durant ses trois années : les cafards du Liris (Oggy, Joey, Deedee, Marky et Bob), le DGX et ses camarades GTX, RTX, etc.

Car ce sont eux qui, par leurs critiques et remarques exigeantes, m'ont permis d'avancer dans ma thèse, je remercie toutes les personnes ayant usé de leur expertise pour évaluer mes travaux. Je pense à Nicolas Thome et Eric Guérin, merci d'avoir participé aux comités de suivi annuel. Je remercie également les rapporteurs et examinateurs de mon jury, David Picard, Nicolas Thome, Cordelia Schmid, Damien Teney and Akata Zeynep. Avec un remerciement spécial pour Damien Teney, dont j'ai croisé la route à plusieurs reprises durant ma thèse, et qui a été un modèle et une source d'inspiration.

Il me reste à remercier ma famille, qui a pris soin d'injecter l'amour de la science dans mon réseau de neurones non-artificiels. Merci à mes amis, et en particulier à Amaury et Théo L. pour tous les bons moments partagés. Merci Adèle, pour ton soutien sans faille, mais aussi pour m'avoir montré que la vie est belle, même quand rien ne marche dans la thèse. Merci d'avoir collaboré à la fabrication d'Élie, venu au monde le dernier jour de ma thèse, et qui est surement le modèle de VQA le plus évolué que j'ai conçu à ce jour.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

NLP	Natural Language Processing
VQA	Visual Question Answering
OOD	Out-Of-Distribution
SOTA	State-Of-The-Art
GT	Ground Truth
AGI	Artificial General Intelligence
DL	Deep Learning
ML	Machine Learning
CV	Computer Vision
CNN	Convolutional Neural Network
GN	Graph Network
SGD	Stochastic Gradient Descent

Chapter 1

GENERAL INTRODUCTION

1.1 CONTEXT AND MOTIVATION

What *color is the tennis court? How fat is the dog? How big is the car to the right of the bicycle underneath the mango tree?* These are existential questions addressed by the VQA task, where an agent answers questions posed over an image.

But above all, VQA aims at studying the emergence of artificial reasoning (*cf.* Chapter 2 for an attempt to define "*reasoning*"). Initially devised as a "*visual turing test*" (Geman et al. 2015), VQA measures the ability of an artificial agent to learn various high-level general representations of concepts of the physical world as well as their interactions: object and attribute recognition, comparison, logical composition, relation detection, etc. Contrary to abstract reasoning tasks – such as variants of the Raven's Progressive Matrices (Barrett et al. 2018; Chollet 2019) – VQA stands out for its multi-modality. The reasoning process is guided by language (through the question) and grounded by vision. Thus, it resembles traditional computer vision tasks such as image retrieval or image captioning, the difference being that VQA involves multi-modal and high-dimensional data as well as complex decision functions requiring latent representations and multiple hops.

Recent advances in Deep Learning (DL) (Goodfellow et al. 2016), combined with the construction of large-scale datasets, have pushed forward the emergence of powerful VQA models. Actually, a VQA model takes advantage of advances in several subfields of DL in order to fulfill three main tasks:

① Understanding the question, by leveraging methods from Natural Language Processing (NLP) like the Transformer architecture (Vaswani et al. 2017) or BERT pre-training (Devlin et al. 2019).

② Understanding the visual scene, by leveraging approaches from Computer Vision (CV) such as object detectors (Ren et al. 2015).

③ Fusing information between vision and language, borrowing models from the multimodal fusion domain, *e.g.* bilinear fusion (Ben-Younes et al. 2017) or Transformer-based cross attention (Yu et al. 2019).

Figure 1.1 provides two illustrative questions, extracted from GQA. In Figure 1.1a the VQA model has to answer the question "*how fat is the animal on the sand?*". It requires to:

(a) *"How fat is the animal on the sand?"*

(b) *"How big is the giraffe on the right?"*

Figure 1.1 – Samples of questions addressed by the Visual Question Answering (VQA) task. Source: GQA dataset (Hudson et al. 2019b).

① analyze the question, to find that the answer must be a size descriptor related to an animal; ② encode the image pixels into a high-level semantic representation where each object is described, *e.g.* a fat blond dog, a large area of sand, etc.; ③ align the question and visual features in order to find the relationships between them, *e.g.* the animal whose size we want to know is the blond dog. Figure 1.1b involves similar mechanisms, but with other concepts and a different reasoning. Indeed, VQA is famous for the wide variety of concepts and reasoning skills it covers.

DO VQA MODELS REASON? On VQAv2, a widely adopted VQA dataset (*cf.* Chapter 3), the State-Of-The-Art (SOTA) already reaches a performance almost competitive with humans (*cf.* Chapter 4). However, despite these impressive improvements brought by DL, it remains unclear if VQA models reason (in Chapter 2, we provide our definition of "*reasoning*"). More precisely, we observe that these models lack robustness and are brittle to many kinds of variation in the data. As an illustration, replacing a single question's word by a synonym can potentially have a dramatic impact on the predicted answer. In fact, we will show in Part II, that the performance drops as soon as the evaluation domain slightly deviates from that of training. This phenomenon is due to the fact that DL models tend to capture spurious correlations found in the training data (also called biases), which do not align with the task's objective. This so-called *shortcut learning* (Geirhos et al. 2020) is characteristic of DL, but the wide diversity of concepts covered by VQA makes it particularly sensitive to it. Figure 1.2 provides two examples of shortcuts learned by the VQA baseline method UpDn (Anderson et al. 2018). In Figure 1.2a, the baseline wrongly predicts that a "*mirror*" is on the wall, because it is infrequent to have a "*star*" on the wall in the training corpus. Similarly, in Figure 1.2b, the baseline fails to predict that the shirt is brown because the training contains a larger amount of blue shirts. *Thereby, before being a* "visual turing test"*, VQA can be seen as a test-bed for studying shortcut learning in DL.*

ON THE IMPORTANCE OF STUDYING SHORTCUTS Beyond the question of reasoning in VQA, shortcut learning potentially leads to the emergence of weak DL models, lacking robustness against many types of variation in the data. This can be problematic for certain

(a) "*What is on the wall?*" – UpDn: A mirror. In the corpus, it is more frequent to find a mirror on the wall rather than a star.

(b) "*Is the shirt brown or blue?*" – UpDn: Blue. In the corpus, shirts are more likely to be blue than brown.

Figure 1.2 – VQA models – here, the UpDn baseline (Anderson et al. 2018) – are notorious for exploiting biases in datasets to find shortcuts instead of performing high-level reasoning. Reproduced from: GQA dataset (Hudson et al. 2019b).

applications, *e.g.* if the model's predictions are used to make critical decisions. Furthermore, shortcut learning tends to exaggerate biases present in the training data. While some of them are useful, others can be particularly harmful. Thus, such algorithmic biases can have negative impacts on our society, raising ethical questions. As an illustration, Buolamwini et al. (2018) demonstrate how gender classification models can be affected by social biases in the training data, leading to discriminative decisions. Closer to VQA, in their paper called *Women also Snowboard*, Hendricks et al. (2018) point out the gender discrimination found in the predictions of image captioning models. Therefore, it seems to be of prime importance to better to evaluate and analyse shortcuts in DL in order to better understand them and mitigate their influence. In this thesis, we propose to address this question through the VQA task.

EVALUATE • ANALYSE • IMPROVE In light of these problems, we propose to address several aspects of VQA under the motto *evaluate, analyse, improve*. *Evaluate*, because DL progress is driven by benchmarks, and we think that it is a priority to set up VQA evaluation methods able to quantify the amount of shortcuts learned by a model. *Analyze*, because evaluation metrics only provide one view of a much more complex system, it is essential to conduct analyses in order to better diagnose strengths and weaknesses of VQA models and to enhance their interpretability. *Improve*, because our ultimate goal is to come up with better models, more robust against shortcut learning, we draw conlusion from our evaluations and analyses in order to improve the models.

1.2 CONTRIBUTIONS OF THE THESIS

1.2.1 *Organization of the manuscript*

The manuscript is organized as follows. Part I provides the background necessary for the reader to understand our work. Then, Part II, Part III, and Part IV, introduce the contributions of this thesis.

PART I (BACKGROUND) provides the background knowledge required to understand the contributions introduced in the thesis. It includes an overview of the DL approaches for VQA, and a discussion on the notion of *reasoning* and *shortcut learning* in DL. We assume that the reader is already familiar with DL and neural networks (*cf.* Goodfellow et al. (2016)). – Chapter 2 and Chapter 3.

PART II (EVALUATE) focuses on the evaluation of VQA models. More precisely, we wonder: *can we measure the reasoning ability of VQA models?* This part begins with a comprehensive study of popular datasets and benchmarks used in VQA, with a critical review of their strengths and weaknesses. We show that the standard evaluation metric (*i.e.* the overall accuracy) is not sufficient to measure the robustness against many kinds of variations (linguistic reformulations, visual editions, distribution shift, etc.), which is related to the reasoning capacity. Hence, we introduce GQA-OOD, a benchmark devised to evaluate VQA models in Out-Of-Distribution (OOD) setting. We experimentally demonstrate that SOTA VQA models – even those specifically designed for bias reduction – fail in our OOD setting. – Chapter 4 and Chapter 5

PART III (ANALYZE) complements the evaluation with an extensive analysis of reasoning and bias exploitation in VQA. Resulting from a collaboration with Théo Jaunet – a PhD candidate working on explainable AI with data visualization – we develop VISQA, an interactive tool targeting the instance-based analysis of the attention mechanisms learned by a SOTA Transformer-based VQA model. In addition, we extend VISQA with a dataset-level analysis. In particular, we propose to study the emergence of reasoning patterns in the attention maps learned by a perfect sighted model (fed with ground truth visual input) and compare it with the standard setting. We experimentally demonstrate that the oracle model more easily learns to relate attention to the task at hand, suggesting a better reasoning. – Chapter 6 and Chapter 7.

PART IV (IMPROVE) draws conclusions from the *evaluate* and *analyze* parts and proposes to improve the VQA model performances. Two directions are explored: (1) supervising the reasoning trough additional objective losses, and (2) transferring the knowledge learned by an oracle with perfect sight to a deployable model. We provide experimental and theoretical evidences demonstrating the effectiveness of these approaches, as well as their complementary. – Chapter 8 and Chapter 9.

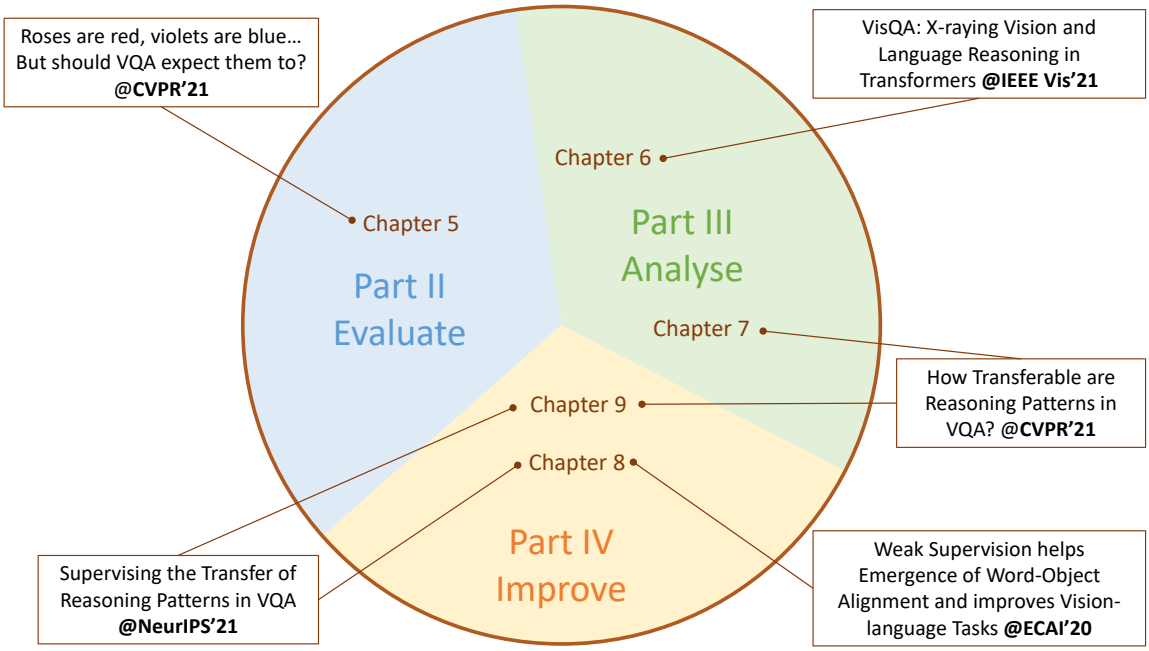


Figure 1.3 – Organization of the manuscript

1.2.2 *List of publications*

This manuscript is based on the material published in the following papers (Figure 1.3 shows where the papers are localized in the thesis):

- Corentin Kervadec, Grigory Antipov, Moez Baccouche, and Christian Wolf (2021b). "Roses Are Red, Violets Are Blue... but Should Vqa Expect Them To?" In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* - Chapter 5;

- Theo Jaunet, Corentin Kervadec, Romain Vuillemot, Grigory Antipov, Moez Baccouche, and Christian Wolf (2021). "VisQA: X-raying Vision and Language Reasoning in Transformers". In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)* - Chapter 6;

- Corentin Kervadec, Theo Jaunet, Grigory Antipov, Moez Baccouche, Romain Vuillemot, and Christian Wolf (2021c). "How Transferable are Reasoning Patterns in VQA?". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* - Chapter 7 and Chapter 9;

- Corentin Kervadec, Grigory Antipov, Moez Baccouche, and Christian Wolf (2019). "Weak Supervision helps Emergence of Word-Object Alignment and improves Vision-Language Tasks". In: *European Conference on Artificial Intelligence (ECAI)* - Chapter 8;

- Corentin Kervadec, Christian Wolf, Grigory Antipov, Moez Baccouche, and Madiha Nadri (2021d). "Supervising the Transfer of Reasoning Patterns in VQA". in: *Advances in Neural Information Processing Systems (NeurIPS)* - Chapter 8 and Chapter 9.

1.2.2.1 *Software and dataset contributions*

The work conducted in this thesis has contributed to the following list of software and released dataset:

- GQA-OOD: a benchmark devised to evaluate VQA in OOD setting and introduced in Chapter 5. It is publicly available at `https://github.com/gqa-ood/GQA-OOD`.
- VisQA: a visual analytic tool that explores the question of reasoning vs bias exploitation in VQA models, introduced in Chapter 6 and publicly available at `https://visqa.liris.cnrs.fr`. It is the fruit of a collaboration with Théo Jaunet, a PhD candidate working on explainable AI with data visualization.

1.3 INDUSTRIAL CONTEXT

This thesis is part of an academia-industry collaboration between INSA Lyon and Orange Innovation (the R&D division of the telecommunication company Orange). As a telecommunication operator handling tons of data every day, Orange is highly interested in the automatic understanding methods based on Machine Learning (ML). In particular, this thesis was initiated by the Multimedia contents Analysis technologieS (MAS) research team of Orange, conducting research on various ML-related topics, such as face recognition (identity, gender, age, etc.), and speech analysis (*e.g.* automatic speech recognition, speaker recognition and diarization). In this context, conducting research on VQA allows to build an expertise on the automated processing of multimodal content – here, image and text – which can be used for various purposes, such as language-based image indexation or multimodal chatbots to improve the customer experience. Furthermore, Orange is also sensitive to the ethical issues of the use of AI. With the intention of building algorithms respectful towards individuals – *e.g.* without social biases – it is essential for Orange to better understand how DL is impacted by shortcut learning.

*
* *

Part I

BACKGROUND: VQA & REASONING

Chapter 2

REASONING VS. SHORTCUT LEARNING

2.1 AN ATTEMPT TO DEFINE "REASONING"

In this thesis, we want to address the problem of *automated reasoning*. More precisely, we target the VQA task where an agent has to predict answers to questions posed over images. In order to fulfill the task, the agent is required to master several skills. Among them, there are perception skills, *e.g.* recognizing an object and its attributes, or recognizing words. But this is not sufficient to solve VQA. In addition to that, the agent is also required to compare, relate, solve logical entailment, etc. Naturally, the first word coming to our mind to describe this set of skills is the ability to "*reason*".

What does it mean to "reason"? While it is common to say that a neural network reasons, we rarely take the time to think about what it really means. At the risk of deceiving the reader, this chapter is not intended to provide an exact definition of "*reasoning*". This would require knowledge and expertise going far beyond the scope of this thesis. At the same time, it would be dishonest to dismiss the question and continue to use a term whose lack of definition leaves too much room for interpretation. That is why, this chapter is our modest attempt to define – or, at least, provide cues on what is – "*reasoning*". In order to narrow the question, we propose to focus on DL, and in particular we try to explain what we mean by "*reasoning*" in the context of VQA.

2.2 REASONING, INDUCTION AND INTELLIGENCE

Reasoning is the deduction of inferences or interpretations from premises.

— Wiktionary (2021)

A plausible definition of "reasoning" could be "algebraically manipulating previously acquired knowledge in order to answer a new question".

— Bottou (2014)

While elegant and concise, these definitions do not provide much information on what "*reasoning*" means. What should be the nature of the inferences and interpretations? What are the conditions such that a knowledge manipulation causes "*reasoning*"? It seems realistic to think that "*reasoning*" only appears under certain conditions.

INDUCTION AND INTELLIGENCE In DL, we conjecture that "*reasoning*" is related to "*induction*" and "*intelligence*". Beforehand, let us define a close friend of the "*induction*", namely "*deduction*":

"*Flowers have petals, a rose is a flower, so every rose has petals*"

The statement above is a "*deductive reasoning*", where a conclusion ("*every rose has petals*") is inferred from premises ("*flowers have petals*" and "*a rose is a flower*"). It is a *top-down* logic, then the validity of the reasoning depends on the quality of the premises. However, it is generally not possible to learn those premises in DL. Neural networks only have access to a restricted set of *i.i.d.* data samples, which is only partially representative of the infinite variations of the real world. In that context, instead of performing *top-down* "*deduction*", DL leverages *bottom-up* "*induction*". Let us take a new example:

"*This rose has thorns, the next rose has thorns, another rose has thorns. So all roses have thorns*" [1]

This does correspond to the main steps of reasoning involved in "*induction*". Unlike "*deduction*", the conclusion of an "*inductive*" reasoning is *probable* rather than *certain*. Is that a problem? We think it is not, because it corresponds to the mechanism involved in experimental sciences, as shown by Popper (1934). This line of thought has led to discoveries like Newton's second law, relativity, and the standard model of physics. However, as seen in the example below, it can lead to wrong theories:

"*This rose is red, the next rose is red, another rose is red. So all roses are red*"

This type of spurious induction is called a "*shortcut*" (Geirhos et al. 2020) (see Section 2.4). As a consequence, Induction as a principle of finding truth does not consistently lead to either all false or all true statements; The quality of the result depends on various factors, including the data from which the conclusions are derived, and the algorithm itself.

Therefore, it might be relevant to relate "*inductive reasoning*" to the faculty of *intelligence*. Legg et al. (2007) survey numerous definition of "*intelligence*" and propose their own version:

> *Intelligence measures an agent's ability to achieve goals in a wide range of environments.*
>
> — Legg et al. (2007)

In that context, "*intelligent reasoning*" denotes the process of organizing inference and interpretations in a way that it generalizes to multiple settings and across various environments. Thereby, it appears that "*reasoning*" requires "*scaling to ever-larger search spaces and understanding the world broadly*" (Bommasani et al. 2021), hence implying properties such as consistency, causality, or compositionality.

2.3 THE MANY FACES OF "REASONING"

The previous definitions remain vague, and are hardly usable in practice. Bottou (2014) tells us that rather than searching for a unique definition of "*reasoning*", it might be more fruitful to consider the many faces of "*reasoning*". It defines different types of reasoning, from which we find: first order logic reasoning, probabilistic reasoning, causal reasoning, or even social reasoning.

1. Because we are not botanist, we still believe that all roses do have thorns ☺.

FIRST ORDER LOGIC REASONING is probably the first facet of reasoning which comes to mind. In few words, first order logic is a powerful mathematical tool allowing to derive logical inference between subjects and predicates. However, there is strong evidence that the human brains do not perform only that type of reasoning. For instance, first order logic is not expressive enough to describe all the nuance of natural language (Bottou 2014). Moreover, the discrete nature of first order logic leads to an expensive computation cost because it generally involves large combinatorial searches.

PROBABILISTIC REASONING treats the problem by manipulating conditional probability distributions. This is the type of reasoning which is typically used in ML. Contrary to first order logic, the continuous nature of probability distributions allows to reduce the computation cost, by using probability theory tools such as Bayesian inference. In addition, it makes possible to reason under uncertainty, which is inevitable when dealing with real-world data.

CAUSAL REASONING highlights one of the major limitation of probabilistic reasoning. Let us consider the correlation between "*it is raining*" and "*people are carrying umbrellas*". In the context of probabilistic reasoning, this correlation is predictive: if "*people are carrying umbrellas*", it is highly probable that "*it is raining*". However, the probabilistic framework does not tell us about the effect of an intervention: if "*it is raining*" but "*people throw away their umbrellas*", is it still raining? Answering this question requires to model the relation of causality between premises. Here, it is the rain which causes people to carry an umbrella, and not the inverse. Pearl et al. (2000) propose to counteract this issue with causal inference. More precisely, it defines a three-level abstraction called the *ladder of causation*. The first step, "*association*", consists in modelling correlations between events, this is what is done in probabilistic reasoning. The second step, "*intervention*", requires modeling the conditional probability distribution of the effect of an intervention (*cf.* the previous example involving umbrellas). Finally, the third step, "*counterfactual*", indicates a full comprehension of the causal relationships, such that it is possible to predict what would have been the present state considering an alternate version of a past event (*e.g.* "*if it was snowing instead of raining, what people would have done?*"). Recently, DL approaches try to adopt insights from causal inference, *e.g.* in vision-language understanding (Teney et al. 2020a) or in counterfactual learning of physics (Baradel et al. 2019).

From a human point of view, "*reasoning*" might not always be rational. Thereby, we can also cite other forms of reasoning, which move away from a mathematical point of view but, in a way, come closer to human reasoning. Even if it is out of the scope of this thesis, commonsense and social reasoning are essential when designing an agent able to reason in the real world.

COMMONSENSE REASONING is a form of reasoning allowing to make presumptions about the type and essence of ordinary situations humans encounter every day (Wikipedia 2021). This implies the ability to make intuitive judgments about the nature of physical objects (*e.g.* a dropped object falls straight down, a solid object cannot pass through another solid object, etc.), taxonomic properties, and peoples' intentions. Therefore,

(A) *Cow: 0.99*, Pasture: 0.99, Grass: 0.99, No Person: 0.98, Mammal: 0.98

(B) No Person: 0.99, Water: 0.98, Beach: 0.97, Outdoors: 0.97, Seashore: 0.97

(C) No Person: 0.97, *Mammal: 0.96*, Water: 0.94, Beach: 0.94, Two: 0.94

Figure 2.1 – Illustration of shortcut learning in an image recognition algorithm. We observe that it generalizes poorly to a new environment. While the cow in 'common' contexts (*e.g.* Alpine pastures) is detected and classified correctly (A), cows in uncommon contexts (beach, waves, and boat) are not detected (B) or classified poorly (C). Reproduced from: Beery et al. (2018)

commonsense reasoning differs from first order logic, probabilistic or causal reasoning as it relies more on intuition and human psychology rather than on modelling relations (logical, probabilistic or causal) between events. At the same time, this form of reasoning is highly desirable when designing an agent to "*think like humans*", *e.g.* if its purpose is to assist people (a chatbot for instance).

SOCIAL REASONING is related to the ability to change its viewpoint. Placing oneself in somebody else's shoes generally induces changes in the way we perceive the world and human intentions (Bottou 2014). As an illustration, the fact that different cultures do not necessarily share the same representation of the world (Descola 2013) (*e.g.* the representation of color) might be an evidence that human reasoning is subjective. This form of reasoning might be useful in the context of modeling social interactions.

These definitions are not completely satisfying, as they cannot fully describe the way humans reason. In any case, human reasoning displays neither the limitations of logical inference nor those of probabilistic inference (Bottou 2014). However, this set of definitions provides a useful tool for evaluating and designing reasoning algorithms.

2.4 REASONING AS THE OPPOSITE OF SHORTCUT LEARNING

In the context of DL, it is simpler to define "*reasoning*" by what it is not. In particular, in this thesis, we define "*reasoning*" as the opposite of exploiting biases and spurious correlation in the training data.

SHORTCUTS AND BIASES Mitchell (1980) define the term "*bias*" to refer to "*any basis for choosing one generalization over another, other than strict consistency with the observed training instances*". In this work, we abusively refer to "*bias*" as the bad ones, *i.e.* a generalization

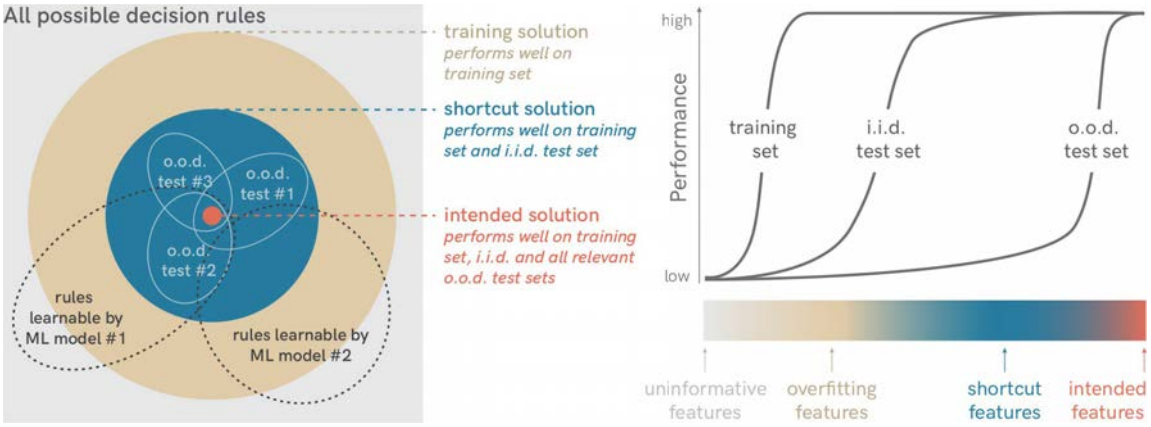


Figure 2.2 – Taxonomy of decision rules. We propose to define reasoning as the ability to learn *intended features*, *i.e.* decision rules which perform well in both training, in-distribution test and out-of-distribution test sets. Reproduced from: Geirhos et al. (2020)

choice which does not generalize to unseen settings. More precisely, our definition of bias exploitation is aligned with the notion of "*shortcut learning*" introduced by Geirhos et al. (2020): "*decision rules that perform well on standard benchmarks but fail to transfer to more challenging testing conditions*". This can be related to the "*simplicity bias*" (Shah et al. 2020), referring to the tendency of models trained with Stochastic Gradient Descent (SGD) (and its variants) to find simple approximations. Paradoxically, it is considered at the same time as a reason for the success of neural nets generalization but also as a cause for their lack of robustness. In few words, the "*simplicity bias*" explains why neural nets tend to exclusively rely on the simplest features while ignoring the complex ones, leading to decision rules which depend on biases found in training data rather than on a complex reasoning.

CONSEQUENCES OF SHORTCUT LEARNING As an illustration, Figure 2.1 shows an image recognition algorithm which have learned to detect the presence of a cow depending on the context (*e.g.* the background) rather than on the animal's characteristics. When evaluated on uncommon contexts, such as a cow in the water or on the beach, this recognition algorithm fails to generalize. Furthermore, as already mentioned in Chapter 1, shortcuts also raise ethical concerns. For instance, Buolamwini et al. (2018) alert on the tendency of gender classification models to be affected by social biases in the training data. Similarly, in the context of vision and language understanding, Hendricks et al. (2018) demonstrate that image captioning models learn gender discriminatory decision rules. This reinforces the stakes of the study of reasoning vs shortcut learning in DL.

OOD EVALUATION Therefore, we propose to define "*reasoning*" following the decision rules taxonomy introduced by Geirhos et al. (2020) (*cf.* Figure 2.2). In particular, we refer to "*reasoning*" as the process leading to the *intended solution*, *i.e.* a decision rule which performs well on the training set, in-distribution and all relevant OOD test sets. In that context, OOD evaluation – which consists in pushing the evaluation beyond *i.i.d.* examples – can be viewed as an effective way to measure shortcut exploitation vs reasoning. That

is why, we propose in Chapter 5 the GQA-OOD benchmark, devised to evaluate the OOD performance of VQA models.

However, we have to keep in mind that defining "*reasoning*" as the opposite of "*shortcut learning*" is also not completely satisfying. It is quite possible that, at some point, a DL model performs something which is neither "*reasoning*" nor exploiting shortcuts. Nevertheless, as we will show in Part II and Part III, detecting shortcut exploitation is an effective way to evaluate reasoning.

2.5 VQA: A VISUAL REASONING TASK?

VQA is often understood as a proxy task for evaluating the "*reasoning*" ability of artificial agents on vision and language inputs (Geman et al. 2015). Indeed, this task requires to understand a visual scene at both general and fine-grained levels. Moreover, it involves skills such as object and attribute recognition, transitive relation tracking, spatial reasoning, logical inference and comparisons, counting or memorizing (Hudson et al. 2019b). More importantly, VQA stands out from other visual understanding tasks because the question to be answered is not determined until run time: VQA models have to adapt the reasoning to the task at hand, by reading the question. Thus, solving VQA might require a general reasoning model able to process a wide variety of questions. This recalls one of the reasoning properties we gave, namely the fact that "*reasoning*" implies to "*generalizes to multiple settings and across various environments*".

LIMITATIONS The popularity of VQA is probably due to practical reasons. As the questions' answers generally contain a few words only, it is easy to automatically evaluate models on million of examples. However, VQA also suffers from several limitations, hindering its ability to evaluate "*reasoning*". First, VQA evaluation is actually not as easy as it seems: naively measuring the prediction accuracy tends to favor models relying on shortcuts instead of reasoning (in Part II we propose a new evaluation method to counter this issue). Second, the variant of "*reasoning*" addressed in VQA is limited, and not comparable with human capacities. Indeed, it mostly involves probabilistic and common-sense reasoning. Causal reasoning has only been recently introduced (Shah et al. 2019; Agarwal et al. 2020), and is still at an exploration stage. Social reasoning is absent: VQA databases are mostly representative of the occidental culture and based on the English language. Nevertheless, we do think that VQA is a preliminary and necessary step paving the way for the emergence of intelligent reasoning systems.

Chapter 3

VISUAL QUESTION ANSWERING

3.1 CONTEXT: VISION-AND-LANGUAGE UNDERSTANDING

VQA (Antol et al. 2015) consists in predicting the answer to questions asked about an input image. Answering the questions requires a wide variety of skills: finding relations, counting, comparing colors or other visual features, materials, sizes, shapes, *etc*. Thereby, VQA lies in *vision and language understanding*, a broad area that can take several forms at many levels of granularity. At the same time, it is also a *reasoning* task (*cf.* Chapter 2).

3.1.1 *Vision and language tasks*

Some vision and language tasks focus on matching problems, as for instance *Image Retrieval*, which requires finding the most relevant image given a query sentence (Karpathy et al. 2015a; Lee et al. 2018). The inverse problem — namely *Sentence Retrieval* — has also been explored (Karpathy et al. 2015a). A similar task with finer granularity is *Visual Grounding*, where the model must associate image regions to words or sentences (Kazemzadeh et al. 2014; Plummer et al. 2015). Other tasks require more high-level reasoning over images and sentences, which, in general, requires multi-modal interactions but also the ability to compare, count or find relations between objects in the image. We can cite VQA, but also the binary task of *Language-driven Comparison of Images*, which takes as input triplets $(img_1, img_2, sentence)$ and requires predicting whether the sentence truly describes the image pair (Suhr et al. 2019), or the visual entailment task (Xie et al. 2019), where the goal is to predict whether the image semantically entails the text.

Finally, some tasks involve the generation of one modality from the other. *Image captioning* consists in translating an image into text (Lin et al. 2014). Other tasks aim to generate questions about an image (Li et al. 2018). Inversely, it is also possible to generate an image from a caption (Mansimov et al. 2016; Ramesh et al. 2021).

3.1.2 *Reasoning tasks*

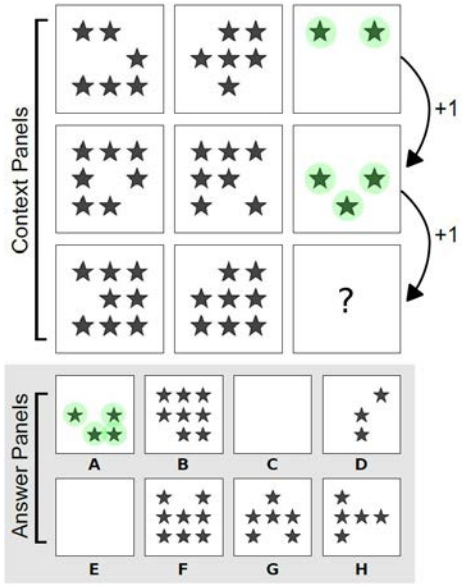
VQA is also a reasoning task (*cf.* Chapter 2). As such, it can be compared to the task defined by the Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al. 2016), which contains 100K questions posed by crowd workers on a set of Wikipedia articles, as

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall?
gravity

What is another main form of precipitation besides drizzle, rain, snow, sleet, and hail?
graupel

(a) Question-answer pairs for a sample passage in the SQuAD dataset. Reproduced from: Rajpurkar et al. (2016).



(b) Measuring abstract reasoning in the form of Raven's Progressive Matrices. Reproduced from: Barrett et al. (2018).

Figure 3.1 – Example of two other reasoning tasks: (a) textual question answering and (b) abstract reasoning.

shown in Figure 3.1a. However, at contrary to VQA, SQuAD only contains text. Embodied Question Answering (Das et al. 2018) goes further than VQA by allowing the model to *interact* with is environment while answering the question. An agent is spawned at random in a 3D environment, and has to move and interact with it to answer questions such as *"what color is the fish tank?"*. This addresses a more realistic type of reasoning, where interaction is as much important as perception. Similarly, the ALFRED (Shridhar et al. 2020) dataset combined an interactive visual environment with natural language directives. Another direction of work focuses on abstract reasoning, taking inspiration from human IQ tests. As an illustration, the benchmarks devised by Barrett et al. (2018) and Chollet (2019) are both variants of Raven's Progressive Matrices (see Figure 3.1b), where the model has to predict complex sequences under various generalization settings.

3.2 VQA DATASETS

Progress on VQA has been driven by the existence of large-scale datasets. One of the first large-scale datasets was VQAv1 (Antol et al. 2015) with $\sim 76K$ questions over 25K realistic images. It started a new task, but was soon found to suffer from biases. Goyal et al. (2017) pointed to strong imbalances among the presented answers and proposed the second (improved) version: VQAv2. Johnson et al. (2017) introduced the fully synthetic CLEVR dataset, designed to evaluate reasoning capabilities. Its strong point is its detailed and structured annotation. In Hudson et al. (2019b), CLEVR is adapted to real-world images resulting in the automatically created GQA dataset (1.7M questions), offering a better

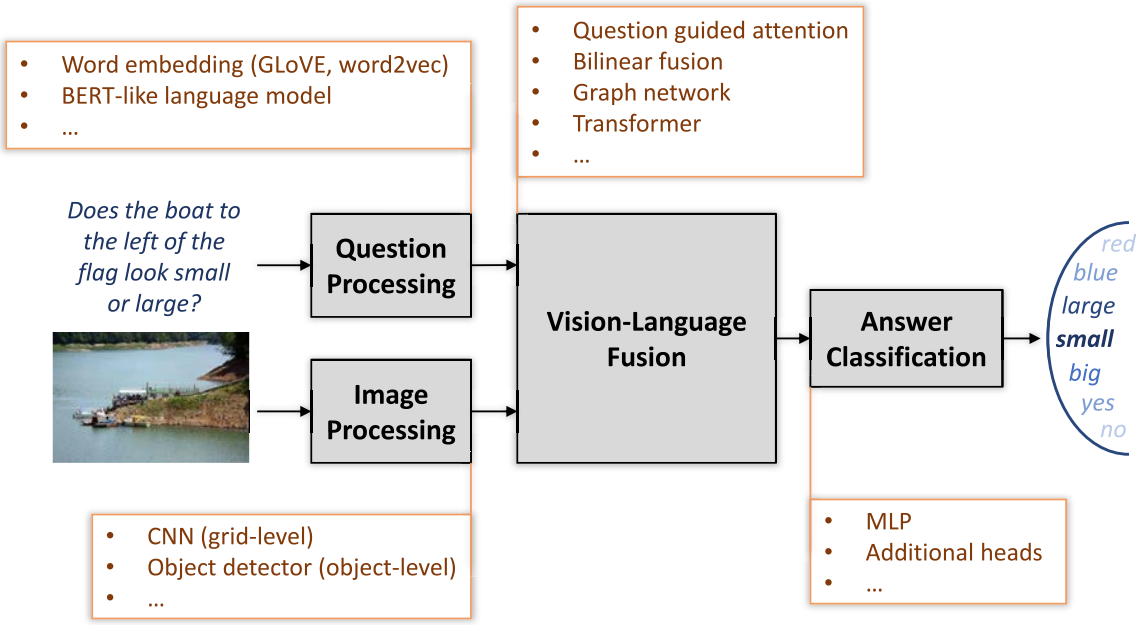


Figure 3.2 – Schematic illustration of the standard VQA pipeline in earlier work of the literature. In this line of work, the pipeline is decomposed into separate models processing the image, the question, the multimodal fusion and the answer classification. Recent works move towards holistic approaches, where the frontiers between these models are less pronounced (*cf.* Figure 3.9).

control on dataset statistics. We let the reader refer to Chapter 4 to get a comprehensive overview of the corpora and benchmarks used in VQA.

3.3 DISSECTING THE VQA PIPELINE

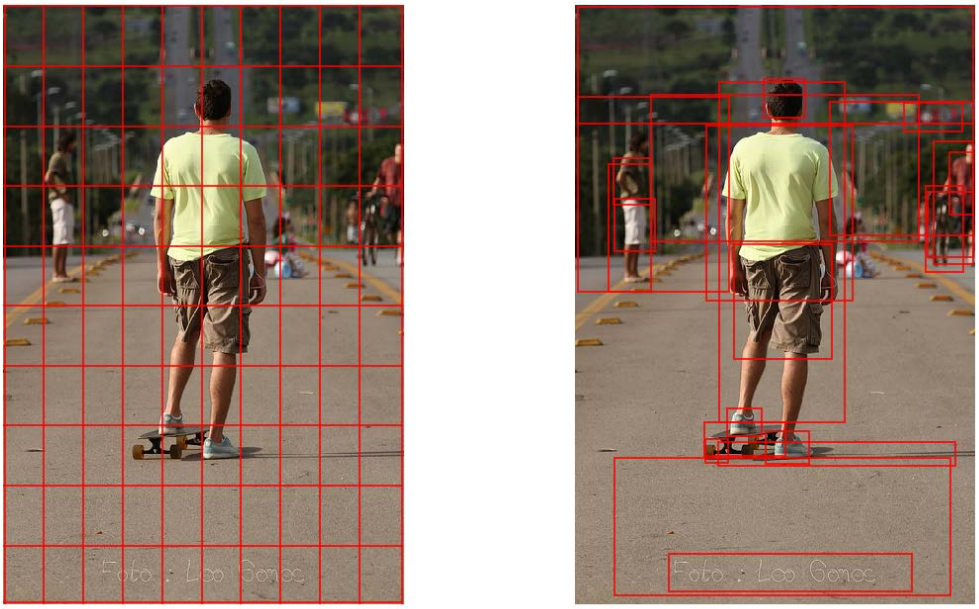
We now describe the standard VQA pipeline, taking as input an image-question pair and returning the predicted answer. Usually, the VQA problem is formalized as follows. Given a visual input v and a question q, the predicted answer $\hat{y}$ can be written as:

$$\hat{y} = \arg\max_{y \in \mathcal{A}} p_\Theta(y|v,q) \tag{3.1}$$

where Θ is the set of model parameters and y is the ground truth answer taken in the dictionary $\mathcal{A}$. As illustrated in Figure 3.2, early work in vision and language understanding focused on separate models for each modality, followed by multi-modal fusion. We will see that recent approaches move toward holistic architectures where both modality are jointly learned (see Figure 3.9).

3.3.1 *Processing the question*

The input question is processed using NLP methods. For instance, one can translate word's tokens into numerical representation using pre-trained embedding — such as word2vec (Mikolov et al. 2013) or GloVe (Pennington et al. 2014) – which contains a



(a) Grid-level features

(b) Object-level features

Figure 3.3 – Images can be represented in two ways: (a) *grid-level*, where the image is uniformly paved following a grid structure; or (b) *object-level*, when the image is decomposed into semantic objects. Reproduced from: Anderson et al. (2018).

semantic representation of the words. Thereafter, in early work, a recurrent neural network — such as LSTM (Hochreiter et al. 1997) or GRU (Cho et al. 2014) — is used to encode the whole sentence into a unique representation. More recently, pre-trained BERT-like (Devlin et al. 2019) models are directly plugged to the VQA architecture to replace those standard words embeddings and recurrent networks.

3.3.2 *Processing the image*

Similarly, the image is processed using CV methods. As shown in Figure 3.3, two main approaches are used: *grid-level* and *object-level*.

GRID-LEVEL FEATURES As in Xu et al. (2015), early work employs a Convolutional Neural Network (CNN) to extract features from the image. In particular, the use of a ResNet (He et al. 2016) pre-trained on Imagenet (Deng et al. 2009) is a popular option. We call them grid-level features as they uniformly pave the image, as shown in Figure 3.3a.

OBJECT-LEVEL FEATURES The Bottom-Up Top-Down architecture (UpDn) (Anderson et al. 2018) introduces the use of object level features for VQA and image captioning. As shown in Figure 3.3b, this type of features is computed for objects and salients regions of the images, which are obtained using a pre-trained object detector such as Faster-RCNN (Ren et al. 2015). More recently, Zhang et al. (2021) propose an improved

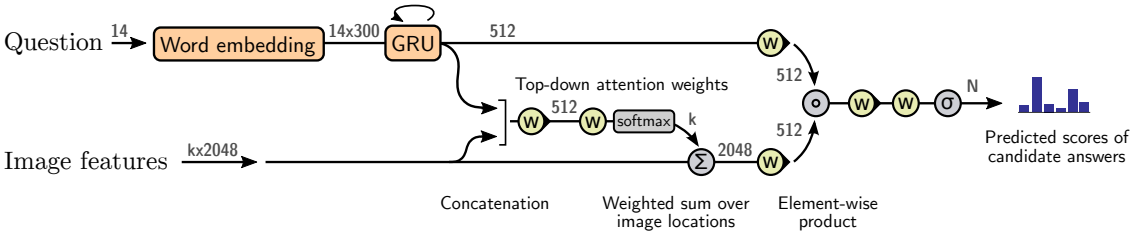


Figure 3.4 – The Bottom-Up Top-Down (UpDn) architecture is a strong VQA baseline. As shown in this schematic illustration, a question guided attention is applied on top of the object-level visual features. Then, vision and language features are fused using an element wise product. Reproduced from Anderson et al. (2018).

object-level representation called VinVL, specifically designed for vision and language tasks.

In practice, object-level features are preferred over the grid-level ones. Indeed, the former generally leads to a higher accuracy, probably because they bring an additional level of abstraction allowing to reason over objects rather than over pixels. However, it remains unclear what really is the advantage of object vs grid-level features. Recently, Jiang et al. (2020) revisited grid-level features and showed they can work surprisingly well while running much faster (object detectors such as Faster-RCNN (Ren et al. 2015) generally add a significant computation overhead). In addition, we will show in Part III that object detectors suffer from inaccuracies that can potentially interfere with learning of reasoning.

3.3.3 *Fusion: from late fusion to multimodal attention*

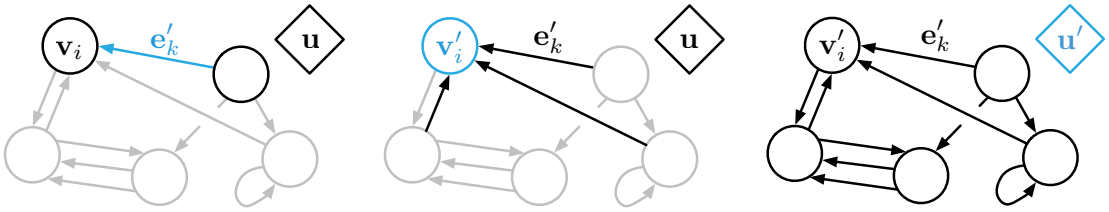
Vision and language modalities need to be fused. This is a fundamental operation, as the whole reasoning process depends on the ability to correctly align vision and language. While early work focuses on late fusion, it is now admitted that a more complex fusion process is required. This implies the use of attention, bilinear fusion, graph networks and, more recently, Transformers (Vaswani et al. 2017).

QUESTION GUIDED ATTENTION Xu et al. (2015) make use of a soft attention mechanism for VQA, where the image regions are weighted by the question. This allows the model to learn to *attend* to specific parts of the image, depending on the question. As an illustration, if $v = \{v_i\}$ is the image representation, where each v_i corresponds to different region features, and q is the question representation, then the attention over the vision is defined as:

$$\hat{v} = \sum_i a_i v_i \tag{3.2}$$

where the attention weights a_i are computed as follows:

$$a_i = \phi(v_i, q) \tag{3.3}$$



(a) Edge update (b) Node update (c) Global update

Figure 3.5 – The Graph Network (GN) framework introduces relational inductive biases in DL architectures by considering the input data as a graph. It works by iteratively updating nodes, edges and global states. Reproduced from Battaglia et al. (2018)

where $\phi(\cdot)$ is a learnable neural network. Then, the fused representation m is obtained by fusing q with the attention product $\hat{v}$:

$$m = f(q, \hat{v}) \tag{3.4}$$

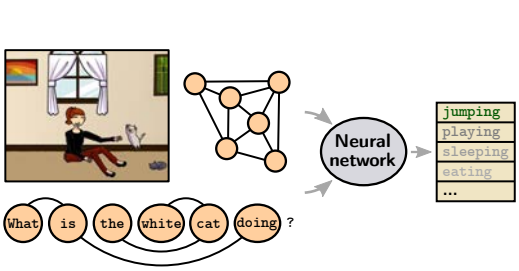
where $f(\cdot)$ is a learnable fusion module, *e.g.* an addition plus a MLP. Yang et al. (2016) go one step further and propose a Stacked Attention Networks (SAN), composed of several iterations of attention in order to perform multi-hop reasoning. Besides, as shown in Figure 3.4, the UpDn (Anderson et al. 2018) model adapts this attention mechanism to object-level features.

BILINEAR FUSION Bi-linear fusion is a more expressive family of models, helping to learn high level associations between question and visual concepts in the image. They consist in encoding fully-parameterized bilinear interactions between the question $q \in \mathbb{R}^{d_q}$ and the image $v \in \mathbb{R}^{d_v}$ representations. It is expressed as follows:

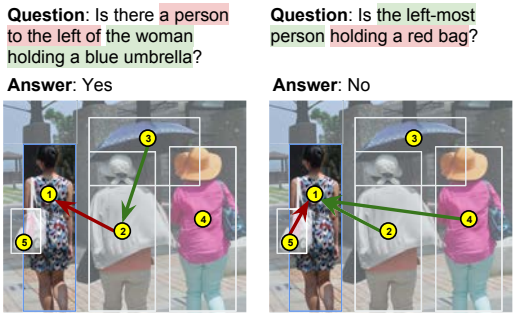
$$m = (\mathcal{T} \times_1 q) \times_2 v \tag{3.5}$$

with $\mathcal{T} \in \mathbb{R}^{d_q \times d_v \times d_m}$ a learnable tensor. The operator $\times_i$ is the *i-mode* product between a tensor and a matrix. However, such a formulation suffers from over parametrization and therefore overfitting. Subsequent work address this by using compact bilinear pooling (Fukui et al. 2016), low-rank bilinear pooling (Kim et al. 2016), or even by creating low-rank decomposition of the fusion tensors, either through Tucker tensor compositions as in MUTAN (Ben-Younes et al. 2017), or block tensor decomposition like in BLOCK (Ben-Younes et al. 2019). Finally, Kim et al. (2018) combine bilinear fusion with attention mechanisms to obtain their Bilinear Attention Network (BAN).

RELATIONAL INDUCTIVE BIASES Although it was already perceptible in some bilinear fusion methods, other approaches introduce relational inductive biases into the fusion architecture, in the form of variants of GN (Battaglia et al. 2018). This fusion paradigm consists in representing the image-question pair as a graph, where the nodes are question words and image regions (or objects). It turns out that many VQA architectures fall into the GN framework drawn by Battaglia et al. (2018) and illustrated in Figure 3.5. GN works by iteratively applying the following operations: (a) the *edge update*, *i.e.* the message passing mechanism allowing to circulate the information between nodes; (b) the *node*



(a) Graph VQA by Teney et al. (2017)



(b) LCGN by Hu et al. (2019)

Figure 3.6 – Graph VQA and LCGN are two methods based on variants of GN, introducing relational inductive biases into the vision-language fusion.

update, which contextualizes each node given the messages from its neighborhood; and (c) the *global update*; which can be viewed as an update of the general state of the graph. In this context, Teney et al. (2017), Norcliffe-Brown et al. (2018) and Hu et al. (2019) propose variants of Graph Convolutional Networks (Kipf et al. 2017) applied to visual objects and question words. Figure 3.6 provides a schematic illustration for two of these methods, namely Graph VQA (Teney et al. 2017) and LCGN (Hu et al. 2019). Besides, the Relation Network (Santoro et al. 2017) is also a GN which only considers the pairwise interactions between visual objects.

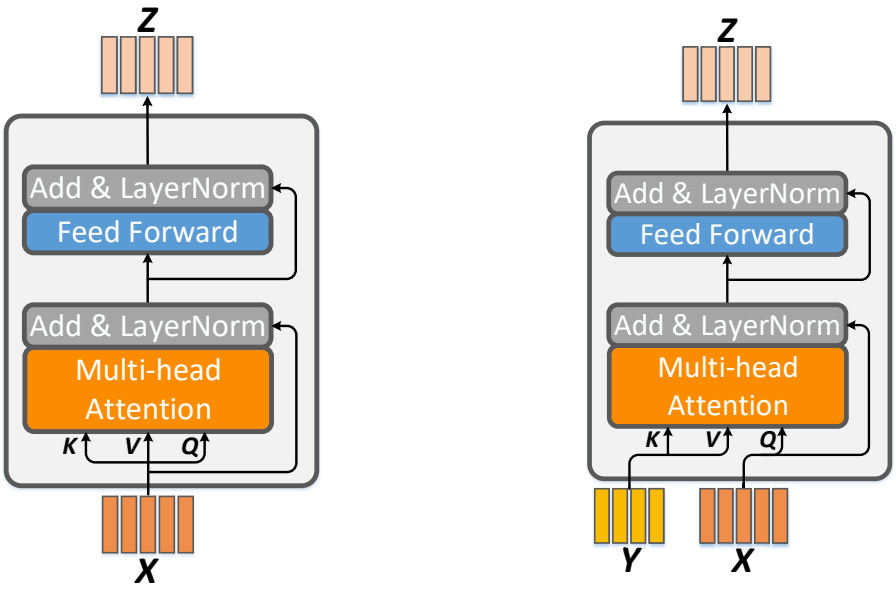
TRANSFORMER The Transformer (Vaswani et al. 2017) architecture can be viewed as a special case of the GN framework, combining message passing with an efficient use of attention. It is composed of a succession of self attention layers, illustrated in Figure 3.7a. Given an input set $\boldsymbol{x} = (\boldsymbol{x_1}, \ldots, \boldsymbol{x_n})$ of the embeddings of the same length d, they calculate an output sequence:

$$\boldsymbol{x_i'} = t\_(\boldsymbol{x}) = \sum_j \boldsymbol{\alpha}_{ij} \boldsymbol{x}_j^v \tag{3.6}$$

by defining the query $\boldsymbol{x}^q$, key $\boldsymbol{x}^k$ and value $\boldsymbol{x}^v$ vectors which are calculated with the respective trainable matrices $\boldsymbol{x}^q = \boldsymbol{W}^q \boldsymbol{x}$, $\boldsymbol{x}^k = \boldsymbol{W}^k \boldsymbol{x}$ and $\boldsymbol{x}^v = \boldsymbol{W}^v \boldsymbol{x}$. In particular, $\boldsymbol{x}^q$ and $\boldsymbol{x}^k$ are used to calculate the self-attention weights $\boldsymbol{\alpha}_{.j}$ as follows:

$$\boldsymbol{\alpha}_{.j} = (\boldsymbol{\alpha}_{1j}, \ldots, \boldsymbol{\alpha}_{ij}, \ldots, \boldsymbol{\alpha}_{nj}) = \sigma\left(\frac{\boldsymbol{x}_1^{q\,T} \boldsymbol{x}_j^k}{\sqrt{d}}, \ldots, \frac{\boldsymbol{x}_i^{q\,T} \boldsymbol{x}_j^k}{\sqrt{d}}, \ldots, \frac{\boldsymbol{x}_n^{q\,T} \boldsymbol{x}_j^k}{\sqrt{d}}\right) \tag{3.7}$$

with σ being the softmax operator. Yu et al. (2019) and Gao et al. (2019) propose to model the multimodal interactions via adapting Transformer principles to vision and language. In particular, they reformulate the unimodal self-attention layer to obtain a multimodal guided-attention layer. This layer is designed to let information circulate between vision and language (see Figure 3.7). In guided-attention, contextualizing vision with language requires extracting *key* and *value* vectors from language, and query from vision (and vice versa). The main advantage of Transformers is that they are able to consider both intra-modality (inside a modality) and inter-modality (fusion between modalities) relationships,



(a) Self-Attention

(b) Guided Attention

Figure 3.7 – Adaptation of the self-attention operation (a) to the vision and language multi-modality. A guided-attention (b) layer is used to contextualize modality X with modality Y. Reproduced from: Yu et al. (2019).

leading to richer fusion. Some models use a two-streams architecture – *e.g.* LXMERT (Tan et al. 2019) or VilBERT (Lu et al. 2019) – where vision and language are first processed in parallel by self-attention layers and then fused using guided-attention layers. Others use one-stream architecture – *e.g.* UNITER (Chen et al. 2020) – where a concatenation of vision and language is directly fed to a Transformer. However, Bugliarello et al. (2020) experimentally show that there are no significant differences between both approaches. As the LXMERT architecture is widely used in this thesis, we propose a detailed overview in Section 3.5.

3.3.4 *Training: from task-specific to multitask*

We also observe the evolution of training from task-specific supervision signals to a set of different losses, which are related to general vision-language understanding, and whose supervision signal can successfully be transferred to different downstream tasks. Recent work shows that a joint pre-training over both modalities can benefit downstream vision-language tasks. This is achieved by setting up strategies to learn a vision-language representation in a multitask fashion similar to BERT (Devlin et al. 2019) in Natural Language Processing (NLP). Thereby, approaches such as LXMERT (Tan et al. 2019), VilBERT (Lu et al. 2019) or OSCAR (Li et al. 2020b) use Transformer architectures to learn a vision-language encoder trained on a large-scale amount of image-sentence pairs. As shown in Figure 3.8, pre-training is done through diverse losses such as: language or vision reconstruction, cross-modality matching and even VQA. The encoder is then transferred to specific vision-language tasks, where they generally achieve SOTA results.

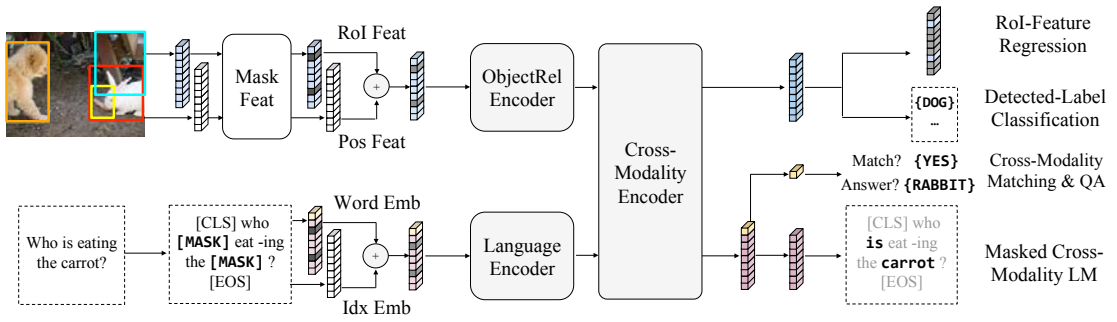


Figure 3.8 – The LXMERT pre-training leverages a set of different losses related to vision-language understanding: language or vision reconstruction, cross-modality matching and even VQA. It is combined with the use of a Transformer based architecture. Reproduced from Tan et al. (2019).

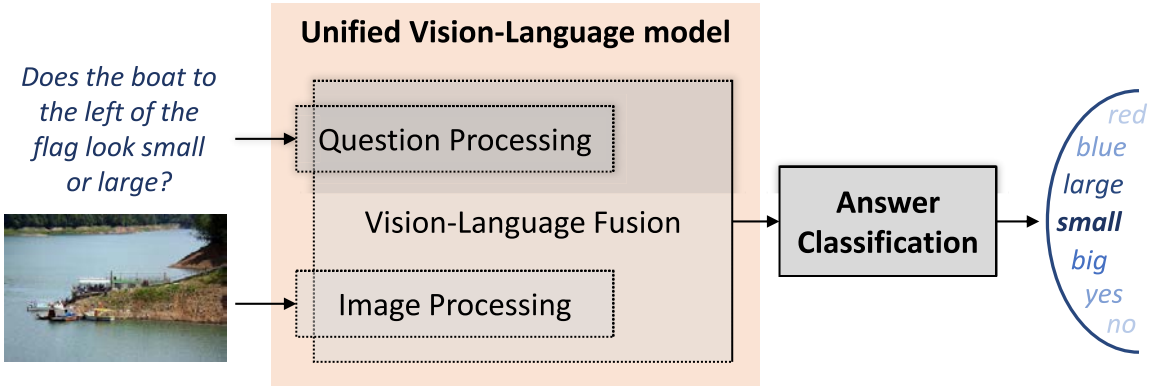


Figure 3.9 – VQA moves towards holistic approaches where a unified vision-language encoder is trained on a large-scale dataset in a multitask fashion.

We observe the same trend for video and language representation learning (Sun et al. 2019).

3.3.5 *From separated to holistic models*

Interestingly, we observe a pronounced tendency to move from separated models – composed of independent components having a specific purpose as shown in Figure 3.2 – to holistic approaches. On the architecture side, especially in the fusion part, models admit more and more degrees of freedom to compute both intra- and inter-modal relationships in a unified vision-language encoder (*cf.* Figure 3.9). On the training side, large-scale vision-language pre-training with weak supervision is now preferred to task specific supervision strategies. As a consequence, model architectures tend to be more and more general and less hand-crafted. For this reason, in this thesis, our efforts concentrate on the training objectives and algorithms: we propose to evaluate and analyze what have been learned by VQA models (*cf.* Part II and Part III) and design new approaches for pretraining and transfer (*cf.* Part IV).

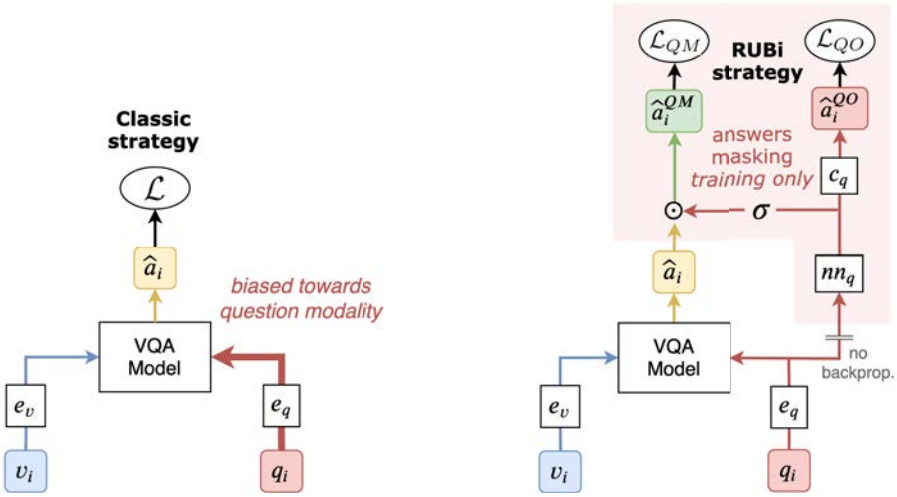


Figure 3.10 – RUBi is a training strategy aiming at mitigating question biases in VQA. During the training, a question-only branch is added to the base model. At test time, the additional branch is removed. Reproduced from Cadene et al. (2019)

3.3.6 *Symbolic representation for visual reasoning*

Aside from these connectionist approaches, others address the visual reasoning problem by constructing a symbolic view of vision, language and of the reasoning process. Thus, Yi et al. (2018) use reinforcement learning to learn a program generator predicting a functional program from a given question. The Neural State Machine (Hudson et al. 2019a) predicts a probabilistic graph from the image to obtain an abstract latent space which is then processed as a state machine. Alternatively, MMN (Chen et al. 2021) is a Meta Module Network for compositional visual reasoning. It is based on a hybrid approach combining neural module networks (NMN) (Andreas et al. 2016) and monolithic architectures (such as Transformer-based ones). The former, NMN, is based on hand-crafted neural network program blocks and is supposed to lead to better compositionality and interpretability. The latter, which is a monolithic architecture, performs its operations in a latent space and has been shown to be experimentally more efficient. MMN tries to combine the best of both worlds.

3.4 Attempts to reduce the bias-dependency

Despite efforts to design complex architectures, VQA models suffer from significant generalization inability (*cf.* Part II). They tend to answer questions without using the image, and even when they do, they do not always exploit relevant visual regions (Das et al. 2016). They tend to overly rely on dataset biases (Hendricks et al. 2018), and are not able to generalize to unseen distributions (Agrawal et al. 2018).

Mitigating biases Assuming that biases are on the language side, Ramakrishnan et al. (2018) set up an adversarial game against a question-only adversary to regularize training. Similarly, RUBi (Cadene et al. 2019) makes use of a question-only branch in

addition to a base model during training to prevent it from learning textual biases (*cf.* Figure 3.10). The training process is then formalized as follows:

$$\hat{y} = \arg\max_{y \in \mathcal{A}} \underbrace{p_{\Theta_1}(y|v,q)}_{\text{base model}} \underbrace{p_{\Theta_2}(y|q)}_{\text{blind branch}} \tag{3.8}$$

The blind branch is supposed to learn the question biases instead of the base model. Hence, at test time, the blind branch is omitted. In the same way, Clark et al. (2019) regularize model predictions using question type statistics from the training set. They propose two variants: Bias Product (BP) and Learned-Mixin (LM). BP is similar to RUBi but differs in directly taking training set statistics to infer question type biases during training. The question type biases are fused with the base model predictions using a product of experts, and removed during testing. LM is an improved version of BP. In this version, the question bias is dynamically weighted by the base model in order to control its influence. An entropy penalty can be added to the loss to prevent the model to ignore the bias. Other approaches force VQA models to attend to the most significant visual regions from humans' perspective (Wu et al. 2019; Selvaraju et al. 2019). However, these methods rely on the known construction of the evaluation split (Teney et al. 2020c), and we will show their limitations in Chapter 5. Alternatively, Teney et al. (2020b) propose a knowledge agnostic de-bias method, showing that training a model on multiple non-*i.i.d.* sets leads to a better OOD generalization.

INJECTING CAUSALITY A promising direction of work for reducing the bias dependency is the use of insights from causal inference in VQA. Abbasnejad et al. (2020) introduce a data augmentation method based on the generation of counterfactual examples. Teney et al. (2020a) and Gokhale et al. (2020a) design a novel supervision loss constraining pairs of counterfactuals (minimally dissimilar samples) to have their gradient aligned with their vector difference in the input space.

3.5 CASE STUDY: LXMERT

The last section of this chapter is dedicated to a detailed overview of LXMERT (Tan et al. 2019), a neural model which is widely used in this thesis, because of its use of self-attention combined with efficient large-scale self-supervised pretraining. It is composed of a VL-Transformer architecture trained with BERT-like losses.

3.5.1 *VL-Transformer architecture*

The key strength of the Transformer-based architecture is its ability to contextualize input representations. This is achieved by a sequence of transformations of the input vectors, and the key mechanism behind these transformations is the concept of attention (self-attention). Language-only and vision-only layers are referred below as *intra*-modal transformers layers, while language-vision layers are referred as the *inter*-modal ones. In this context, we present the Transformer architecture illustrated in Figure 3.11 which we call VL-Transformer, and which corresponds to the one used in LXMERT (Tan et al. 2019).

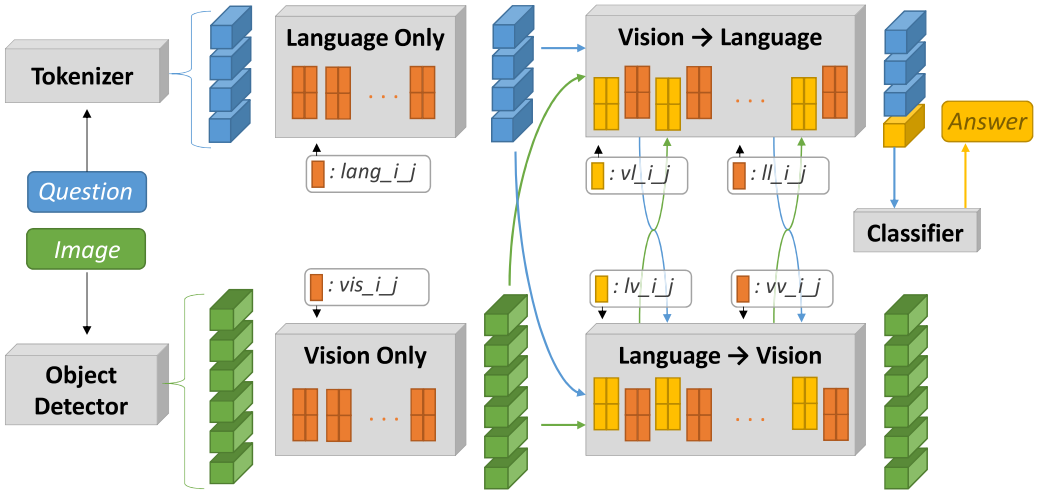


Figure 3.11 – Schematic illustration of the VL-Transformer architecture used in the thesis. Question and image are first tokenized. They are then encoded using vision (in green) and language (in blue) only Transformers (Vaswani et al. 2017). At the next step, the information flow between the two modalities (bidirectional) thanks to inter-modality Transformers (Tan et al. 2019). Finally, the answer is predicted from the *'CLS'* token. Yellow and orange rectangles represent respectively inter- and intra-modality attention heads. i and j are the layer and head indices used for naming attention heads through the thesis.

We use the following naming convention for the VL-Transfomer: each layer is named as *xxx_i_j*, where $xxx \in \{\text{lang}, \text{vis}, \text{vl}, \text{lv}, \text{ll}, \text{vv}\}$ denotes the layer type (*e.g.* vision-only intra-modal layer, vision-language inter-modal layer, etc.), while i and j are respectively the layer and head indices.

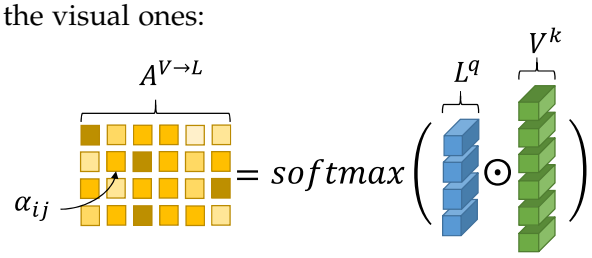
VISION INPUT On the vision side, we use an object detector – Faster-RCNN (Ren et al. 2015) – to extract object level visual features from the input image as in Anderson et al. (2018). Similar to hard attention mechanisms, this enforces the system to reason on the object level rather than on the pixel level or global level. In particular, the visual input embeddings are concatenations of 2048-dimensional object embeddings and the corresponding 4-dimensional bounding box coordinates.

LANGUAGE INPUT On the language side, sentences are tokenized using the WordPiece tokenizer (Wu et al. 2016). As common in language processing, a special token `[CLS]` is added at the beginning of the tokenized sentence, which encodes the multimodal information of the image and sentence. The transformation of this token, performed during the forward pass through the network, corresponds to the prediction of the answer to the task. Tokens are embedded into d-dimensional vectors using a look-up table learned during the training phase. The index position of the word is added to the dense vector as a positional encoding in order to obtain index-aware word level embeddings.

INTRA-MODALITY Visual and language modalities are firstly processed independently using a two-streams approach (*cf.* Figure 3.11). More precisely, the self-attention heads *lang_i_j* are used to encode the words of the question, as described in the example above.

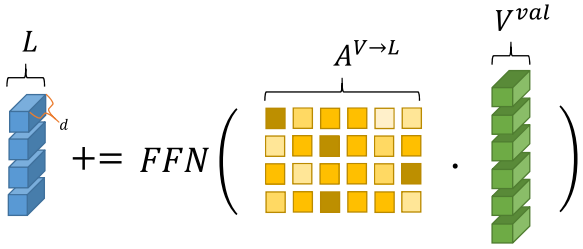
In the same spirit, the *vis_i_j* heads encode the visual modality, *i.e.* the different objects and theirs embeddings.

INTER-MODALITY Then, in order to take into account the inter-modality structure of the input, this architecture let the information flow between language and vision, as shown in Figure 3.11. This contextualization is bidirectional: from the question's words to visual objects in *lv_i_j*, and vice-versa in *vl_i_j* (*lv* means 'language to vision' while the opposite *vl* means 'vision to language'). This requires a minor, but essential, modification of the intra-modality transformer. In particular, the use of guided-attention (Yu et al. 2019) to operate on both modalities. More precisely, the *query* vectors are taken from the modality to be contextualized, and the *key* and *value* vectors from the other one. Thereby, in the case of the vision to language heads, *vl_i_j*, attention maps $A^{V \to L}$ are computed as the outer product between the *query* projections L^q of the language embeddings and the *key* projections V^k of the visual ones:



(3.9)

A row-wise softmax function is applied, such that each attention map's row sums to 1. Then, the language embeddings L are updated with the *value* projections V^{val} of visual tokens:



(3.10)

where "$+=$" represents a residual connection and FFN is a trainable feed-forward layer. For the sake of clarity, we omit the description of the multi-head mechanism in the notation of Equation 3.9 and Equation 3.10. Nevertheless, it is important to notice that the inter-modality Transformers are multi-headed, similarly to the intra-modal ones. As shown in Figure 3.11, each *lv* or *vl* attention head is immediately followed by an intra-modal attention head called, respectively, *vv* or *ll*.

MULTI-HEAD To increase the learning power of the described self-attention mechanism, the attention layers in Transformers are often *multi-headed*. This means that, at each layer, h attention maps are computed in parallel. These parallel operations are called the *attention heads*. At the end of each Transformer layer, the outputs of the attention heads are concatenated and followed by token-wise residual connections and feed-forward layers.

INTERPRETING ATTENTION MAPS In this thesis, we will sometimes focus on the interpretation of the attention maps, as these maps contain the information which is crucial for the Transformer's functionality. Indeed, these maps tell us to what extent a

given token has been contextualized by its neighbors. A low attention value α_{ij} indicates a weak interaction between tokens i and j. Inversely, a high value is an indicator of the strong information flow from j to i. Therefore, attention maps provide strong insights on how our VL-Transformer has modeled the question, the image, and, more importantly, the relationships between both modalities.

ANSWER PREDICTIONS The VQA task is finally achieved by decoding the final representation of the textual `[CLS]` token using a 2-layered neural network. In particular, our model outputs a probability vector over the set of the most frequent answers found in the training set. The final predicted answer is then the one with the highest score.

HYPERPARAMETERS In the following chapters of this thesis, we use two versions of the VL-Transformer architecture. The *original version*, similar to the one used in LXMERT (Tan et al. 2019), is composed of 9 language only layers, 5 vision only layers, and 5 cross modal layers. Its hidden size is set to d=768 and the number of per-layer heads to h=12. Thus, it is composed of 212M parameters. The *compact version* has the same number of layers, but a smaller hidden size d=128 and only h=4 heads per layers. It allows reducing computation time and memory overhead as it has only 26M trainable parameters. Following Anderson et al. (2018), we use 36 objects per-images.

3.5.2 *LXMERT pre-training*

The so-defined vision-language encoder is trained following the recently widely-adopted strategy of combining BERT-like (Devlin et al. 2019) self-supervised signals with task-specific supervision signals, which has been applied to various problems in vision and language — *e.g.* in Tan et al. (2019) or Lu et al. (2019). Following Tan et al. (2019), it combines four supervision signals: vision masking, language masking, image-sentence matching and VQA, which are briefly described below. This pre-training allows to learn a general vision-language understanding. Thereafter, a fine-tuning can be necessary to adapt to the downstream task.

VISION/LANGUAGE MASKING This signal aims to supervise the encoder's ability to reconstruct missing information in language and vision. More precisely, it randomly mask each language token (resp. visual object) with a probability of 0.15 and ask the model to predict the missing words (resp. objects). Therefore, two classifiers are added – for *vision masking* [1] and *language masking* – on top of the vision language encoder and supervised via a cross-entropy loss. Tan et al. (2019) proposes to take the object detector prediction as ground truth in order to get over the disparity of visual annotation. Additionally, the model is also supervised to regress the masked objects' features via L2 loss.

IMAGE-SENTENCE MATCHING BERT (Devlin et al. 2019) proposes *next sentence prediction* supervision by asking to predict if two sentences are consecutive in a given text, or randomly sampled from a corpus. Its vision-language equivalent is *image-sentence*

1. It is worth noticing that vision masking requires to predict both the object classes and their attributes (*e.g.* color, materials, *etc.*)

matching, where the model has to predict whether a given sentence matches a given image or not. Thus, in each sentence-image pair, the image is randomly replaced with a probability of 0.5. A feed-forward layer is added on top of the `[CLS]` output embedding to predict whether the pair matches or not. This global matching is supervised using a binary cross-entropy loss.

VISUAL QUESTION ANSWERING The VL-Transformer is applicable to a wide range of vision-language problems. At the same time, independently of the target vision-language task, pretraining on VQA helps reasoning, as shown by Tan et al. (2019). The VQA task is defined as a classification problem over a set of most frequent answers. This classification is performed from a prediction head attached to the `[CLS]` token and supervised using a cross-entropy loss.

*
* *

Part II

EVALUATE: WHERE WE LEARN THAT VQA MODELS ARE (STILL) NOT REASONING

INTRODUCTION

In the year 2021 AD, deep learning based VQA models already achieve close-to-human performance on VQAv2.

- *Sir, does it mean that we solved Artificial General Intelligence (AGI)?*
- Of course, *not*. Try to ask your own questions to a SOTA VQA model, and you will be convinced: it is so easy to fool them!
- *But Figure 3.12 is clear, only few years to wait before reaching super-human accuracy!*
- Not really. Actually, I am not sure that we are correctly evaluating the reasoning ability of VQA models. But let me explain all from the beginning!

Evaluating reasoning in VQA is a difficult task. In part because it is hard to define what "*reasoning*" is, but also because evaluation can be fooled by many confounders. As explained in Chapter 2, we can define reasoning as "*algebraically manipulating previously acquired knowledge in order to answer a new question*" (Bottou 2014). In practice, we choose to define *reasoning* by opposition to *biased prediction*, when the model leverage statistical shortcuts (often present in the training data) in order to infer predictions. While being effective on the popular benchmarks, shortcut learning leads to the emergence of models brittle to many kinds of variations in the data: linguistic reformulations, visual editions, distribution shift, etc. Thus, several works (*e.g.* Agrawal et al. (2016)) have alerted on the urgent need to define new evaluation methods, taking into account this shortcut dependency. These methods can take the form of OOD benchmarks, measuring to what extent the models generalize to unseen settings. However, as we will see in this part, most of the OOD benchmarks are subject to many issues related to the presence of unwanted confounder, potentially hindering the performance measures. Therefore, we propose the GQA-OOD benchmark, our contribution to the evaluation of reasoning in VQA. Part II is organized as follows:

CHAPTER 4 is an extension of the related work (Chapter 3), including a comprehensive study of the most popular databases and benchmarks used in VQA. This chapter will be an opportunity for the reader to familiarize himself with the stakes and limitations of the VQA task. In particular, we provide a critical review of the benchmarks dedicated to the evaluation of models' robustness, showing that they are not sufficient to properly measure the VQA reasoning ability.

CHAPTER 5 introduces our GQA-OOD benchmark, dedicated to the OOD evaluation of VQA models. We argue that it answers to most of the concerns raised in Chapter 4, leading to a better estimation of the reasoning capability. This benchmark allows us to experimentally demonstrate that current SOTA VQA models are prone to the usage of shortcut in the data, and are highly ineffective in the OOD setting.

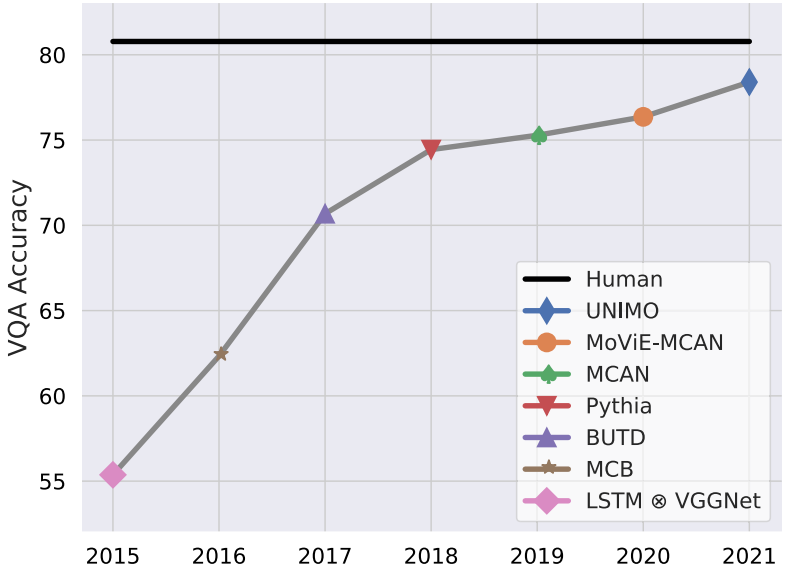


Figure 3.12 – VQA models achieve near human performance on VQAv2. Reproduced from Sheng et al. (2021).

This part has led to the publication of the following conference paper:

- Corentin Kervadec, Grigory Antipov, Moez Baccouche, and Christian Wolf (2021b). "Roses Are Red, Violets Are Blue... but Should Vqa Expect Them To?" In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*;

Chapter 4

PITFALLS OF VQA EVALUATION

4.1 INTRODUCTION

This chapter aims at drawing a comprehensive review of popular databases and evaluation benchmarks dedicated to the VQA task. It is motivated by the fact that the data is one of the central aspect of deep learning based VQA approaches. Indeed, the recent performances of VQA models are largely due to the construction of large-scale corpuses. Each database differs in many ways. *What is the nature of the images, synthetic or real? Are the question generated automatically? Or is it human questions? How is the annotation, weak or detailed? How diverse are the questions? Which reasoning capacities are covered? How is the quality of the annotations?* We propose to review the most popular datasets in light of these questions.

In a second part, we focus on how to evaluate VQA model's robustness. We show that the initial metric, *i.e.* the widely used overall accuracy, is not sufficient to properly assess the models' reasoning ability. Several benchmarks have been proposed to improve the VQA evaluation, focusing on diverse aspect of VQA: linguistic and visual robustness, consistency, compositionality, etc. We will show that VQA evaluation is a hot topic: approaches are numerous, sometimes contradictory, and they often fall into worrying pitfalls.

The purpose of this chapter is to help the reader delving deep into the numerous challenges raised by the VQA task – and also its limitations – through the lens of data. This comprehensive study of VQA evaluation methods will also motivate the Chapter 5, where we introduce our contribution to the VQA evaluation.

4.2 VQA DATASETS

We first overview the most popular datasets used for VQA. Table 4.1 provides a summary of their different characteristics. At first glance, we observe that the datasets differ by the nature of their data. Some corpora are fully synthetic (*e.g.* CLEVR from Johnson et al. (2017)), while others are partially synthetic (only the questions are automatically generated, *e.g.* GQA from Hudson et al. (2019b)) or 100% generated by humans (*e.g.* VQAv2 from Goyal et al. (2017)). But the devil is in the details, so we propose a detailed

Dataset	#I (K)	#Q (M)	Real images	Natural questions	Amount of annotation	Human acc. (%)	SOTA (%)
VQAv1	205	0.6	✓	✓	-	83.3	-
VQAv2	205	1.1	✓	✓	-	80.8	81.3
CLEVR	100	0.9			++	92.6	> 99
VizWiz	34	0.03	✓	✓	-	75.0	54.8
GQA	113	1.7	✓		+	89.3	64.7

Table 4.1 – Overview of the most popular VQA datasets. Note that GQA statistics corresponds to its balanced version. SOTA accuracies are taken from: evalai leaderboard (VQAv2 and VizWiz test-std), Zhang et al. (2021) (GQA test-std) and Yi et al. (2018) (CLEVR test). As VQAv1 is no longer used, we do not provide the SOTA.

Figure 4.1 – Illustration of a balanced pair in VQAv2. *"Which plane wing has a logo under it?"*. Both images are similar but have a different answer. Source: Goyal et al. (2017)

overview of each one of the most popular datasets, providing both quantitative and qualitative descriptions.

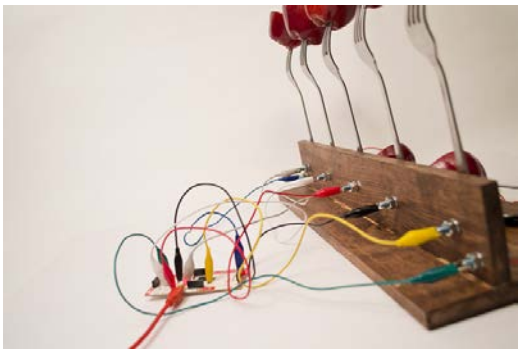
4.2.1 *The VQA dataset: versions 1 and 2*

One of the first large-scale datasets was VQAv1 (Antol et al. 2015) with $\approx$ 0.6M questions over 205K realistic images, but it was soon found to suffer from biases: a blind model (which has only access to the question) is able to achieve $\approx$ 50% of the accuracy! Goyal et al. (2017) point to strong imbalances among the expected ground-truth answers. As an illustration, "*tennis*" is the correct answer for 41% of the "*What sport...?*" questions. As a consequence, they propose the second (improved) version of the dataset: VQAv2.

DATA DISTRIBUTION To mitigate the language priors found in VQAv1, the VQAv2 authors balance the dataset by collecting complementary images, such that each question is associated to a pair of similar images with different answers, as shown in Figure 4.1. However, the experiments show that biases remain problematic as a blind model still reaches 44% of accuracy on VQAv2.

Stump a smart robot! Ask a question about this scene that a human can answer, but a smart robot probably can't!
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, it can recognize the scene (e.g., kitchen, beach), people's expressions and poses, and properties of objects (e.g., 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.

Figure 4.2 – Annotators' directives for the VQA dataset. Source: Antol et al. (2015).



(a) "*What is this machine going to do?*" GT:?

(b) "*Would it be a difficult bet, to suggest whether the bench or the tree will last longest?*" GT: No.

Figure 4.3 – Tricky questions from VQAv1 and VQAv2. Sources: Antol et al. (2015) and Goyal et al. (2017)

REASONING SKILLS The VQA dataset is composed of open-ended questions asked by humans. This allows to collect interesting and diverse questions, going beyond simple low-level computer vision knowledge: object detection, activity recognition, commonsense reasoning, OCR, counting, etc.

LIMITATIONS However, in addition to the imbalanced data distribution, the VQA dataset suffers from weaknesses due to its collection process. In both versions, questions are collected using Amazon Mechanical Turk workers. The directives for the annotator were formulated as in Figure 4.2. In a few words, annotators were asked to "*fool a smart robot*". This sometimes resulted in tricky questions moving away from the initial objective of measuring the visual reasoning skills. On the extreme level, Figure 4.3 shows two inappropriate questions found in VQAv1 and VQAv2: the first one requires to *imagine* what is the machine's purpose (without consensus on the ground truth), while the second requires making a *subjective judgment* about a bet. It is difficult to quantitatively estimate the proportion of these questions. However, the fact that the collection process is explicitly encouraged to "*stump a smart robot*" suggest that these tricky questions are not isolated cases. In addition, the VQA dataset contains a large proportion of questions where the image content is not sufficient to find the answer. Thereby, in VQAv1, 18% of the questions

(a) "*What type of dog is this?*" GT:german shepherd

(b) "*Did Goldilocks, traditionally, encounter this creature?*" GT: No.

Figure 4.4 – Question requiring common-sense knowledge in VQAv2. Sources: Goyal et al. (2017)

requires external knowledge such as baseball team, clothing brand, dog breed, etc. (*cf.* Figure 4.4). Although this external knowledge is considered to belong to commonsense , it produces questions going beyond the simple visual Turing test initially targeted by the VQA task. Because of this, and due to the difficulty of collecting large-scale annotations, the quality of annotation is questionable: $\approx$ 17% of the VQAv1 questions cannot be answered by a human.

4.2.2 *VizWiz: VQA for visually impaired people*

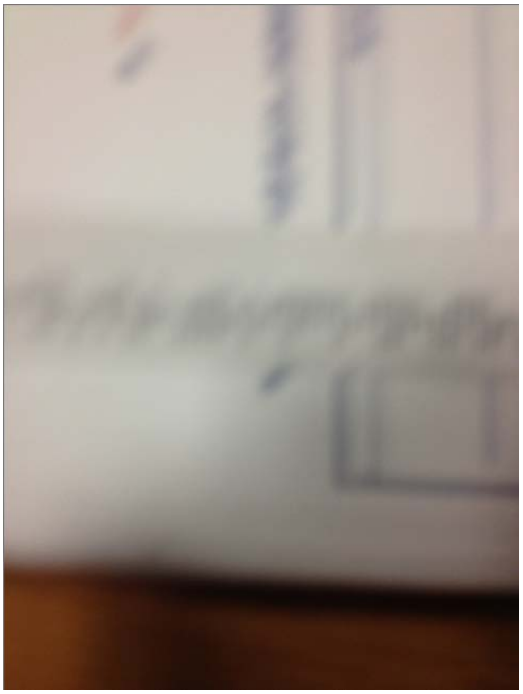
VizWiz (Gurari et al. 2018) pushes the realness of VQA to its extreme limits. This dataset gathers over "*31,000 visual questions originating from visually impaired people who each took a picture using a mobile phone and recorded a spoken question about it*". As a result, VizWiz is probably the VQA dataset which is the best aligned with a real-world usage. It differs from VQAv2 in several aspects: (1) the questions are targeted to help a person asking for an information on the image, and not to stump a smart robot (*cf.* Figure 4.5a); (2) as images are captured by visually impaired photographers, they are often poor quality and sometimes not answerable (*cf.* Figure 4.5b); (3) questions are spoken, and so are more conversational. As a result, at time of writing, the SOTA only reaches an accuracy of 54.8% (*cf.* VizWiz leaderboard), making it one of the most challenging VQA dataset.

REASONING SKILLS One of the main challenge of VizWiz lies on the perception side, where it is required to cope with low-quality images. Therefore, less importance is given to the evaluation of reasoning. Nevertheless, it still requires diverse interesting skills such as detecting when a question is answerable, reading, counting, understanding evasive questions, etc.

4.2.3 *The synthetic CLEVR*

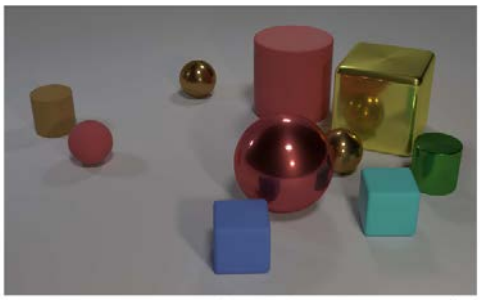
On the opposite side, Johnson et al. (2017) introduced the fully synthetic CLEVR dataset, designed to diagnose reasoning capabilities by disentangling perception from reasoning.

(a) *"What's the name of this product?"* GT: basil.



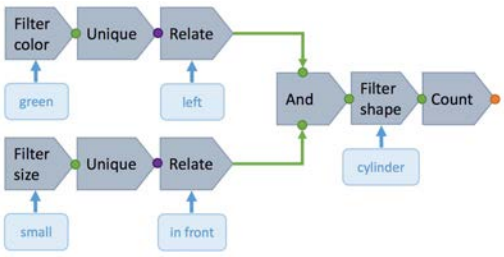
(b) *"Alright, and what does this label say?"* GT: unsuitable.

Figure 4.5 – VizWiz samples. Source: Bhattacharya et al. (2019)



Q: Are there an equal number of large things and metal spheres?
Q: What size is the cylinder that is left of the brown metal thing that is left of the big sphere? **Q:** There is a sphere with the same size as the metal cube; is it made of the same material as the small red sphere?
Q: How many objects are either small cylinders or metal things?

(a) Illustrative questions.



(b) A functionnal program.

Figure 4.6 – CLEVR samples. Source: Johnson et al. (2017)

It results in a very simple environment. As shown in Figure 4.6a, images are composed of simple 3D objects arranged on a planar surface, where each object is determined by its shape (cube, cylinder or sphere), color (8 colors), material (rubber or metal), size (large or small) and position (x and y).

REASONING SKILLS The images are procedurally annotated with complex questions including attribute identification, counting, comparison, spatial relationships and logical operation. In addition, the strong point of CLEVR is its detailed and structured annotation. As shown in Figure 4.6b, each question is translated into a functional program composed of individual operations. This allows for precise evaluation of the reasoning skills.

LIMITATIONS Despite the apparent complexity of the questions, SOTA models already reach an accuracy above 99% (Yi et al. 2018) on CLEVR. This suggests that the VQA bottleneck is in combining reasoning with perception, rather than in the abstract reasoning alone. Indeed, when the environment becomes more complex (as in real world) it leaves more place for visual uncertainty, which can be one of the cause of shortcut learning. We will analyze this in Part III.

4.2.4 *GQA: VQA on image scene graphs*

Taking the best of both worlds, Hudson et al. (2019b) adapt CLEVR to real-world images. It results in the automatically created GQA dataset (1.7M questions), offering a better control on dataset statistics. In particular, each image is associated with a scene graph of the image's objects, attributes and relations (which have been manually annotated), allowing to automatically generate questions using pre-defined templates. As in CLEVR, each question is associated with a functional program that specifies the reasoning steps needed to be taken to answer it. As a result, GQA can be viewed as a compromise between a controlled environment (like in CLEVR) and realistic data (like in VizWiz). Since its creation, the GQA dataset has been rapidly adopted by the VQA research community.

DATA DISTRIBUTION Significant efforts have been made to mitigate the data biases by smoothing the answer distribution of all question groups (grouped according to their context). Interestingly, the data smoothing has been applied to GQA such that "*it retains the general real-world tendencies*" (Hudson et al. 2019b): it thus still contains natural biases, which will be studied in Chapter 5. As in the VQA dataset, a blind model still achieves a relatively high accuracy of 41%.

REASONING SKILLS The GQA dataset covers a large variety of reasoning skills such as object and attribute recognition, transitive relation tracking, spatial reasoning, logical inference and comparisons. In order to give to the reader a better understanding of the skills covered by GQA, we provide illustrative samples: spatial reasoning (Figure 4.7a), object recognition (Figure 4.7b), attribute recognition (Figure 4.8a), logical inference (Figure 4.8b), weather classification (Figure 4.9a), and comparison (Figure 4.9b). It is worth noticing that GQA does cover neither counting questions nor OCR, which are generally brittle to annotation errors. In addition, GQA focuses on factual questions,

(a) *"Is the cabbage to the left or to the right of the carrot that is to the left of the broccoli?"* GT: Left.

(b) *"What piece of furniture is it?"* GT: Sofa.

Figure 4.7 – GQA samples: (a) spatial reasoning, (b) object recognition. Source: Hudson et al. (2019b).

(a) *"What color is the trash can in the top?"* GT: Brown.

(b) *"Are there both fences and helmets in the picture?"* GT: Yes.

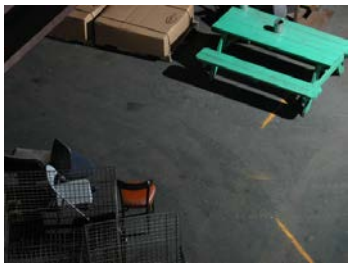
Figure 4.8 – GQA samples: (a) color detection, (b) logical operation. Source: Hudson et al. (2019b).

(a) *"How is the weather?"* GT: Rainy.

(b) *"Are the napkin and the cup the same color?"* GT: Yes.

Figure 4.9 – GQA samples: (a) weather classification, (b) comparison. Source: Hudson et al. (2019b).

(a) *"Are there any people to the right of the umbrella that looks dark blue?"* GT: No.



(b) *"Which kind of furniture is blue?"* GT: Desk.

(c) *"What is the food on the plate of the food called?"* GT: Cookie.

Figure 4.10 – Issues in GQA annotation: (a) not answerable, (b) annotation error, (c) odd syntax. Source: Hudson et al. (2019b).

where the answer can always be predicted from the image only, without requiring external knowledge as in VQAv2.

LIMITATIONS Its semisynthetic nature is also the cause of several limitations. Because the questions are synthetic, they have a limited linguistic diversity. As an illustration, GQA only covers 88.8% and 70.6% of VQAv2's questions and answers. The template-based generation can also result in strange wording, as shown in Figure 4.10c where the question is "*What is the food on the plate of the food called?*". Furthermore, generating such a large-scale dataset favors the emergence of noisy annotations. It is relatively frequent to encounter ambiguous questions. As an illustration, in Figure 4.10b both the table and the chair are blue, and in Figure 4.10a there is more than one "*umbrella that looks dark blue*". Thus, as shown in Table 4.1, $\approx$ 11% of the GQA questions cannot be answered (which is still lower than in VQAv1).

Overall, despite these limitations, GQA involves a larger variety of reasoning skills (spatial, logical, relational and comparative) than in datasets with human questions (such as VQAv2), making it more suitable for evaluating reasoning. Additionally, it limits the requirement of extremely domain-specific knowledge unavailable during training, *e.g.* the logo of a specific baseball team or the breed of a dog. At the same time, because it is based on real images, it is more challenging than CLEVR, and allows studying the vision bottleneck. For these reasons, we use GQA as a testbed for the majority of our studies conducted in this thesis, while being aware of its limitations.

4.2.5 *Other datasets*

Many datasets have not been cited in this overview, such as Visual7W (Zhu et al. 2016), or the pioneering work DAQUAR (Malinowski et al. 2014) first introducing the VQA task. We let the reader refer to Wu et al. (2017) for a detailed overview of older VQA datasets. We can also cite the TDIUC dataset (Kafle et al. 2017), being close to GQA by essence. They propose to divide the questions into 12 different types, including absurd questions. More importantly, they develop several metrics aiming to provide an unbiased score of the model performances.

4.3 MEASURING ROBUSTNESS IN VQA

Are we really sure that our VQA models reason? Despite the numerous datasets (and dedicated benchmarks) available for the VQA task, the question persists. Hudson et al. (2019b) inform us that models (even the baselines) learn to predict answers that are often plausible, suggesting they have learned a consistent representation of the world. But, at the same time, Agrawal et al. (2016) reveal that VQA models are "*myopic*" (tend to fail on sufficiently novel instances), often "*jump to conclusions*" (converge on a predicted answer after "*listening*" to just half of the question), and are "*stubborn*" (do not change their answers across images). Many attempts have been recently taken to try to better evaluate VQA in the form of variants of the existing datasets. These benchmarks can be seen as OOD evaluations, each one focusing on measuring the robustness against a specific variation (syntactic, visual, multi-modal, etc.).

4.3.1 *The standard metric: overall accuracy*

VQA is generally considered as a classification task over a large dictionary, ranging from 1000 to 3000 possible answers depending on the dataset. Hence, the standard metric used for the majority of the datasets is *overall accuracy*, *i.e.* the proportion of the correctly predicted answers over the amount of total predictions. However, we note some subtle variants. For instance, in VQAv1, VQAv2 and VizWiz, each question is answered by ten annotators and the evaluation metric takes into account the (non) agreements between them, by weighting the accuracy. Interestingly, the pioneering work led by Malinowski et al. (2014) initially proposed a metric taking into account the semantic of the prediction, which was then abandoned in favor of overall accuracy.

4.3.2 *Robustness against linguistic variation*

VQA-Rephrasing (Shah et al. 2019) proposes to evaluate the robustness against linguistic variation. For this purpose, they manually reformulate the questions of VQAv2 while making sure that the answer remains the same. For instance, the question "*What is in the basket?*" is reformulated to "*What does the basket mainly contain?*". Despite the apparent simplicity of the modification, the benchmark shows a very weak robustness of SOTA models against linguistic reformulations. As an illustration, the baseline UpDn (Anderson et al. 2018) accuracy decreases from 61.5% to 51.2% when evaluated on original and reformulated questions respectively.

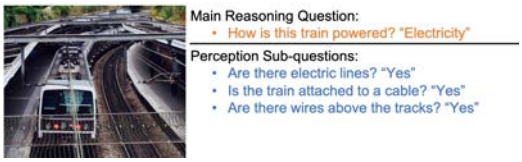
4.3.3 *Robustness against visual variation*

It is also possible to evaluate the robustness again visual variations, as proposed by IV/CV-VQA (Agarwal et al. 2020). This benchmark is constructed by applying semantic editions to the images. In particular, a GAN-based resynthesis model is used to remove some objects from the image. Two types of modifications are explored:

(a) CV-VQA: "*How many giraffes are there?*"

(b) IV-VQA: "*Is there a cat?*"

Figure 4.11 – Robustness against visual variations. Source: Agarwal et al. (2020).



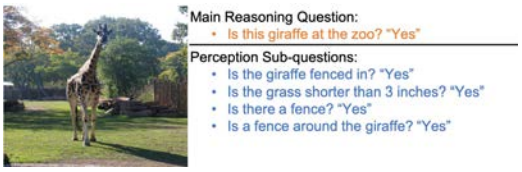


Figure 4.12 – VQA-Introspect: measuring consistency. Source: Selvaraju et al. (2020)

- *InVariant (IV-VQA) set (Figure 4.11b)*: removing objects not required to answer the question. This should not have any impact on the prediction, except if the model is relying on visual shortcuts.
- *CoVariant (CV-VQA) set* (Figure 4.11a): removing objects such that the answer change. It is limited to counting questions, where removing one of the important objects reduces the numerical answer by one.

Afterward, we can measure how the model is affected by the visual intervention. In particular, the authors observe that VQA models are brittle to visual variations. More importantly, this lack of robustness is present even when the model initially predicted the correct question, *i.e.* the model flips its prediction to an incorrect one after the image modification. This suggests (and we will confirm it in Chapter 5) that providing a correct answer does not necessarily imply the presence of a reasoning process, and that models tend to exploit spurious correlation in the data.

4.3.4 *Consistency across questions*

Several works propose to measure the consistency of the predictions. Hudson et al. (2019b) introduces the *consistency* metric in GQA, measuring if the model does not contradict itself when answering several questions of the same image. Similarly, Ray et al. (2019) construct L-ConVQA a benchmark evaluating the consistency and showing similar results. Their most strict metric – *perfect-consistency*, measuring the proportion of consistent question sets where all the questions have been correctly answered – barely reaches 40% with the UpDn baseline, showing that there is a large room for improvement.

VQA-Introspect (Selvaraju et al. 2020) – based on VQAv2 – goes one step further, and proposes to measure the consistency while splitting questions into *reasoning* and *perception*:

- *Perception* questions can be answered by detecting or recognizing a low-level property in the image, *e.g.* "*What is next to the table?*")

- *Reasoning* questions can be answered by solving several *perception* questions, *e.g.* "*Are the giraffes in their natural habitat?*"

Thereby, as shown in Figure 4.12, each *reasoning* question is related to a group of *perception* questions. In that context, VQA-Introspect evaluates the consistency by measuring when a *reasoning* question is correctly answered while the associated *perception* questions are also correct. Authors analyze two types of inconsistency: (a) when *reasoning* is correct but not *perception*, it is probable that the model is using a shortcut instead of reasoning; (b) inversely, if *perception* is correct but not *reasoning*, it indicates a reasoning failure. Interestingly, the baseline – Pythia (Jiang et al. 2018) – achieves a relatively high consistency of 70%.

4.3.5 *Compositionality*

One property of reasoning is compositionality. In VQA, this corresponds to the ability to answer questions resulting from a combination of sub-question, *e.g.* "*Is the man wearing a hat and glasses?*". GQA contains many of such question, but other benchmarks specifically focus on the evaluation of compositionality.

VQA-LOL (Gokhale et al. 2020b) proposes to tackle VQA "*under the lens of logic*". They augment the VQAv2 dataset by adding logical compositions and linguistic transformations (negation, disjunction, conjunction and antonyms). They show that the LXMERT (Tan et al. 2019) model trained on VQAv2 does not perform better than random on composed questions.

CLOSURE (Bahdanau et al. 2019) conducts a similar evaluation on top of the CLEVR dataset, by constructing new questions resulting from unseen associations of known linguistic structures (mostly through referring expressions). Here again, they show that models poorly generalize to these settings, loosing 15% to 35% of their baseline accuracy. Also built upon CLEVR, CoGenT (Johnson et al. 2017) measures the compositional generalization by evaluating models on unseen combination of attributes (*e.g.* the training set contains blue sphere and green cubes while the test set contains green sphere and blue cubes). Without surprise, many models also fail in this setting.

4.3.6 *Multimodal robustness*

Other works evaluate robustness against distribution shifts. We call this *multimodal robustness* as it does not specifically focus on language or vision, but rather on the multimodal context.

The VQA-CP2 (Agrawal et al. 2018) dataset was a first of its kind and paved the way for follow-up work on bias reduction methods in VQA. It has been constructed by reorganizing the training and validation splits of VQAv2 (and VQAv1) aiming to maximise differences in answer distributions between training and test splits. Basically, rare answers in the train set become frequent answers in the test set, as shown in Figure 4.13. They experimentally demonstrate that VQA models are brittle to changes in the distribution. As an illustration, the baseline UpDn (Anderson et al. 2018) has its accuracy decreased from 65% in in-domain to 39% in OOD (Teney et al. 2020c).

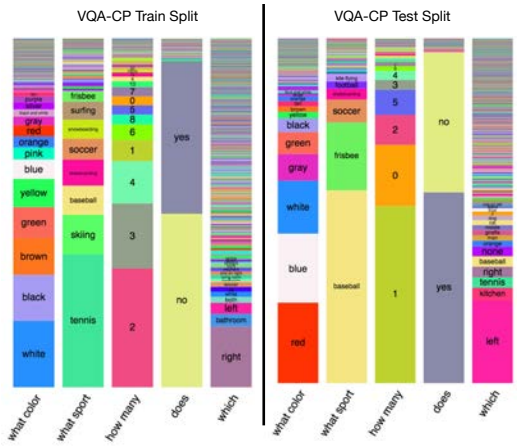


Figure 4.13 – VQA-CP: distribution in train vs test. Source:Agrawal et al. (2018)

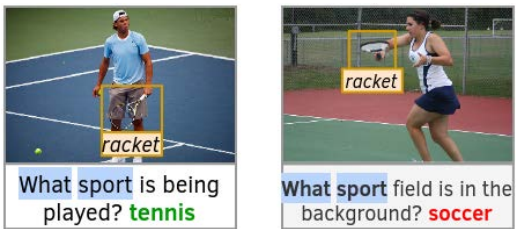


Figure 4.14 – The association of "*What sport*" with the presence of a *racket* is frequently associated to the answer "*tennis*". VQA-CE proposes to evaluate models specifically when these shortcut are not effective. Source: Dancette et al. (2021)

In a similar trend, VQA-CE (Dancette et al. 2021) propose to evaluate VQA models on *counterexamples*, where relying on shortcuts is ineffective. For this purpose, they first apply a mining algorithm on top of VQAv2 in order to extract frequent associations of *{words, visual objects, answer}*, which is the *easy* question set. Then, they create a *counterexample* set, containing questions which contradict the frequent associations. As an illustration, in Figure 4.14, the association of the words "*what sport*" and the visual object *racket* frequently lead to the answer "*tennis*". The *counterexample* set will contain samples having "*what sport*" in the question and a *racket* in the image, but with an answer which is not "*tennis*". Once again, results are clear: The UpDn baseline achieves 77% on the *easy* set while reaching only 34% on *counterexamples*.

4.3.7 *Adversarial robustness*

Finally, we observe a very recent and promising trend for adversarial benchmarks involving humans in the loop. adVQA (Sheng et al. 2021) and AVVQA (Li et al. 2021) have been similarly constructed by asking human annotators to find questions where a SOTA VQA model was failing, using the VQAv2 images. They found that it was surprisingly easy to trick the SOTA models, showing once again their lack of robustness. In that setting, the UpDn baseline has its accuracy decreased from 68% to 20%. It is worth noticing that this data generation process could also be used for data-augmentation during the training (but in that case, the benchmark no longer measures adversarial robustness). However, a risk exists that these adversarial datasets lead to questions irrelevant to the "*visual turing test*" objective (*i.e.* evaluating the visual reasoning ability), as already noticed for VQAv2 questions.

Benchmark	Target	No validation	Violate Goodhart	ID/OOD re-train
VQA-Rephrasing	Lingual robustness	✗		
CV/IV-VQA	Visual robustness	✗	✗	
VQA-Introspect L-ConVQA	Consistency	✗	✗	
VQA-LOL CLOSURE CoGenT	Compositionality		✗	✗
VQA-CE VQA-CP **gqa-ood (ours)**	Multimodal robustness	✗ ✗	✗	✗
adVQA AVVQA	Adversarial robustness		✗	

Table 4.2 – We compare several dataset variants dedicated to the robustness evaluation. Many of them suffer from serious weakness: lack of validation set, violation of Goodhart law or impossibility to evaluate on in- and out-of-distribution without retraining.

4.4 PITFALLS OF VQA EVALUATION

All in all, it seems that VQA models are far from being robust against many types of variations. This suggests that they heavily rely on spurious shortcuts instead of reasoning. In deep learning, benchmarks are useful for making diagnoses and shedding light on the model's weaknesses. More importantly, benchmarks are also a powerful tool driving the design of new methods. Therefore, if a benchmark is not properly devised, it will lead to the emergence of models with unwanted behaviors.

We have seen that overall accuracy, the standard metric in VQA gives us a wrong estimation of the model's reasoning performances. It is then legitimate to ask: *are these robustness benchmarks trustworthy for designing robust VQA models?*

Unfortunately, we have reasons to be skeptical. Evaluating reasoning is difficult, and in many cases evaluation methods are biased by spurious confounders, leading to negative results. Recent works have raised concerns about such OOD evaluation protocols. In particular, Teney et al. (2020c) point out several pitfalls observed when evaluating VQA in OOD setting using VQA-CP (Agrawal et al. 2018). We briefly overview the principal criticisms, which are summed up in Table 4.2.

4.4.1 *Violating Goodhart's law*

Several works rely on known construction procedures of the OOD test split, violating Goodhart's law: "*when a measure becomes a target, it ceases to be a good measure*" (Teney et al.

2020c). The most eloquent illustration is VQA-CP where knowing that the test answer distribution is the inverse of the train distribution allows a model to significantly boost accuracy. Paradoxically, it results in models overfitting this particular OOD setting, without increasing the generalization on unseen distributions. Similarly, in VQA-Introspect, CV/IV-VQA and VQA-LOL, the proposed baselines rely on data augmentation employing the same generation process as the one used to construct the benchmark, without any careful analysis of potential confounders hidden in the generation process.

4.4.2 *Issue in in- and out-of-distribution comparison*

Some benchmarks (such as VQA-CP or CoGenT) do not allow for the possibility to evaluate the performance in both in- and out-of-distribution settings without having to retrain the model on a different set of data. This results in evaluating two copies of the same model, but optimized on different training sets, with different label distributions. However, as demonstrated by Teney et al. (2020c), a method can behave differently depending on the distribution of its training examples. Hence, it ensues in a biased comparison of the in- vs out-of-distribution performance.

4.4.3 *Validating on the test set*

As surprising as it may seem, a majority of the dataset variants does not provide any validation set. Most bias-reduction techniques therefore seem to optimize their hyperparameters on the test split (Cadene et al. 2019; Clark et al. 2019; Ramakrishnan et al. 2018; Wu et al. 2019; Selvaraju et al. 2019), which should be frowned upon, or, alternatively, validate on a subset of train which does not include a shift (Teney et al. 2020b), which is suboptimal. Obviously, selecting hyperparameters on the test split automatically leads to an overestimation of the performances. At the same time, it is worth noticing that having statically separated validation and test splits is not ideal either. Indeed, it is still possible to (slowly) overfit on the test because of multiple evaluations and model comparisons. An interesting direction would be dynamic benchmarks, which evolve through time in order to avoid any potential spurious confounder during the evaluation. Adversarial benchmarks with humans in the loop, such as adVQA or AVVQA are good potential candidates.

4.4.4 *Impact on VQA methods*

These pitfalls of OOD evaluation have a negative impact on the design of new VQA methods. As an illustration, Shrestha et al. (2020) come up with an interesting negative result while analyzing bias-reduction methods based on visual grounding designed on top of the VQA-CP dataset. These methods (Wu et al. 2019; Selvaraju et al. 2019), attempting to supervise a VQA model to attend to visual regions which are relevant to a human considering the question, are very efficient on VQA-CP. Surprisingly, Shrestha et al. (2020) found that simply enforcing the model to attend to random visual regions was at least as much efficient on out- and in-distribution settings. *Why was this negative result not observed*

before? We think that a more profound empirical evaluation of models' behavior would help to better judge and compare the efficiency of new VQA methods.

4.5 CONCLUSION

This chapter draws a comprehensive study of the popular datasets and benchmarks dedicated to the VQA task. We show that several large-scale databases are available, with different settings: synthetic vs natural data, strong annotation, realness, etc. However, we also shed light on potential issues – related to data distribution, linguistic diversity, poor annotation quality, presence of tricky questions, etc.– which could hurt VQA training. Without falling into pessimism (these databases have led to the emergence of powerful models), we think that it is important to be aware of the databases' limitations as it is the root of every deep learning based model.

We then review numerous benchmarks, pointing out the lack of robustness of current VQA models. They confirm the databases' weaknesses, and in particular their inability to accurately evaluate the VQA models. However, we argue that many robustness benchmarks are not trustworthy, preventing them from helping VQA model designers to build more robust models. Most of the criticism introduced is related to wrong practices, in part to the responsibility of the model designers. But it is also a broader issue related to flaws taking root in the current machine learning scientific method, *e.g.* see Forde et al. (2019) or Gorman et al. (2019).

Drawing conclusion from it, we construct (in Chapter 5) our own benchmark – GQA-OOD– dedicated to the evaluation of robustness against distribution shift. We will show that many of the previously designed bias-reduction methods are ineffective in our setting.

Chapter 5

GQA-OOD: EVALUATING VQA IN OOD SETTINGS

5.1 INTRODUCTION

Efforts to learn high-level reasoning from large-scale datasets depend on the absence of harmful biases in the data, which could provide unwanted shortcuts to learning in the form of "*Clever Hans*" effects. Unfortunately, and in spite of recent efforts (Goyal et al. 2017; Hudson et al. 2019b), most VQA datasets remain very imbalanced. Common concepts are significantly more frequent, *e.g.* the presence of a "*red rose*", compared to out of context concepts like the presence of a "*zebra in a city*". This causes the tendency of models to overly rely on biases, hindering generalization (Cadene et al. 2019; Clark et al. 2019). Despite a consensus on this diagnostic, systemic evaluations of error distributions are rare. In particular, overall accuracy is still the major, and often unique, metric used to evaluate models and methods, although it is clearly insufficient. Several questions remain open. *How is error distributed? Are true positives due to reasoning or to exploitation of bias? What is the prediction accuracy on infrequent vs. frequent concepts? How can we validate models in OOD-settings?*

In this chapter we propose a new benchmark and a study of SOTA VQA models, which allows to precisely answer these questions. The proposed new evaluation protocol is complementary to existing ones, but allows a better diagnostic of current VQA performance. In particular, our benchmark can be viewed as an alternative to the VQA-CP (Agrawal et al. 2018) dataset, which has lead to mixed results (see Chapter 4). Our benchmark comprises (i) a new fine-grained reorganization of GQA introducing distribution shifts in both validation and test sets (see Figure 5.1-a); (ii) a set of evaluation metrics; (iii) new evaluation plots illustrating the generalization behavior of VQA models on different operating points. The choice of GQA is motivated by its useful structuring into question groups, which allows capturing biases precisely, to select groups with strong biases and to create distribution shifts tailored to the exact nature of each question (see Figure 5.1-b). It also makes it possible to analyze how errors are distributed over different associations of concepts according to their frequency in the dataset.

CONTRIBUTIONS OF THE CHAPTER

(i) We propose and make public [1] a new fine-grained re-organization of GQA and a set of the respective evaluation metrics allowing to precisely evaluate the reasoning

1. https://github.com/gqa-ood/GQA-OOD

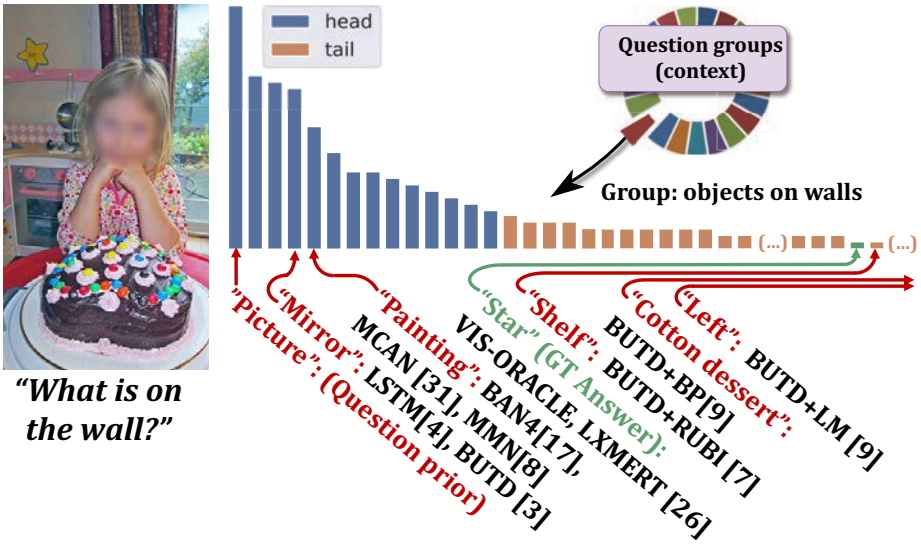


Figure 5.1 – We address bias exploitation in VQA and propose a new benchmark for Out-Of-Distribution evaluation containing distribution shifts tailored to different question groups with highly imbalanced distributions. A new evaluation metric based on rareness inside each question group, here shown for "objects on walls", is experimentally demonstrated to be less prone to bias exploitation. We show that SOTA methods (7 VQA models and 3 bias reduction methods) reproduce biases in training data.

behavior of VQA models and to characterize and visualize their generalization behavior on different operating points w.r.t distribution shifts.

(ii) Compared to competing benchmarks, our dataset features distribution shifts for both, validation and test, allowing to validate models under OOD conditions.

(iii) We experimentally evaluate the usefulness of the proposed metric, showing its behavior on models trained to, more or less, exploit biases.

(iv) In a large study, we evaluate several recent VQA models and show that they struggle to generalize in OOD conditions; we also test several SOTA bias reduction methods and show that there is still room for improvement in addressing bias in VQA.

5.2 GQA-OOD: A BENCHMARK FOR OOD SETTINGS

We introduce a new VQA benchmark named GQA-OOD designed to evaluate models and algorithms in OOD configurations. We here define OOD samples as rare events, in particular measured w.r.t. to a base distribution, *e.g.* a training distribution. These rare events might involve concepts which are also present in the training set. Let's for instance consider the question: *'What color is this rose?'*. If the image represents a rose, then *red* would be a common color, but in an OOD setting, infrequent (correct) test answers would be, for instance, *blue*, requiring models to reason to provide the correct answer. We design a benchmark where this shift is not global but depends on the context. If the context changes, and the flower type is a violet, then a (correct) OOD answer would now be *red* instead of *blue*.

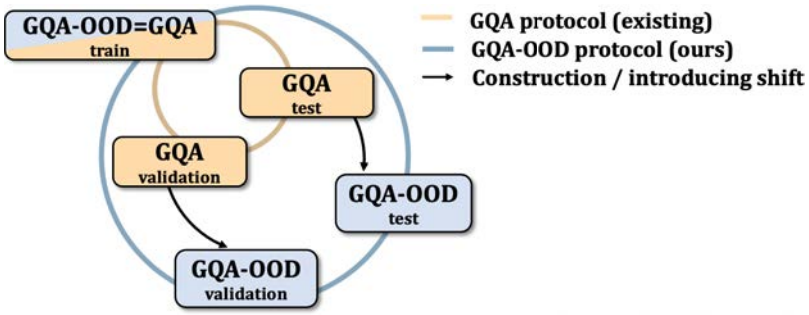


Figure 5.2 – We re-organize GQA (Hudson et al. 2019b) in a fine-grained way: the benchmark contains a distribution shift in validation and test, allowing to validate *and* evaluate in OOD settings.

Dataset	Split	#Quest.	#Groups	#Imgs
GQA-OOD	val	51,045	3,849	9,406
	testdev	2,796	471	388
GQA	val	132,062	36,832	10,234
	testdev	12,578	7,803	398

(a)

GQA-OOD	Subset	#Quest.	#Groups	#Imgs
val	head	33,882	3,849	8,664
	tail	17,163	3,849	6,632
testdev	head	1,733	471	365
	tail	1,063	471	330

(b)

Table 5.1 – Data statistics: (a) GQA-OOD vs. GQA; (b) head vs. tail

5.2.1 *Dataset construction*

The GQA-OOD benchmark consists of a dataset and new evaluation metrics. The dataset itself is based on the existing GQA (Hudson et al. 2019b) dataset [2], which provides more fine-grained annotations compared to competing VQAv2 (Goyal et al. 2017) (the questions in GQA have been automatically generated from scene graphs, which allows better control of the context). Figure 5.2 shows how the proposed protocol compares to the existing GQA protocol: the two share the same (existing) training set, but we introduce fine-grained shifts into both the validation and the test sets applying the process further described below. The shifted subsets have been constructed in 3 steps: (i) dividing questions into groups according to their contexts; (ii) extracting the most imbalanced question groups, considering their answer distributions; (iii) selecting OOD samples among the remaining questions.

QUESTION GROUPS To structure the process introducing distribution shifts, we use the notion of *local groups* provided in the GQA annotation. They allow to precisely define the type of question, *e.g.* *'What color ...?'*, *'Where is ...?'*, etc. They also depend on the concepts related to the question, *e.g.* *'zebra'*, *'violet'*, etc. There is a total of $\approx 37K$ local groups related to $\approx 132K$ questions in the GQA validation split. We use the balanced version of GQA, whose question distribution has been smoothed in order to obtain a more uniform answer distribution. However, this does not impact the imbalanced nature of the dataset, which is often due to real-world tendencies, *e.g.* that *'roses are red'*.

2. We use version 1.2 of GQA (Hudson et al. 2019b).

MEASURING GROUP IMBALANCE We extract a subset of the most imbalanced question groups, as we are interested in evaluating the prediction error specifically in the context, where shifts in distribution are meaningful and strong. We measure balance through Shannon entropy, given as:

$$e(x) = -\sum_{i=0}^{d} p(x_i) \log p(x_i)$$

where $p(x_i)$ is the estimated probability of the class i. As entropy depends on the number of answer classes, which is highly variable between different question groups, we normalize entropy w.r.t. the number d of possible answers in the group:

$$\bar{e}(x) = \frac{e(x)}{\log(d)}$$

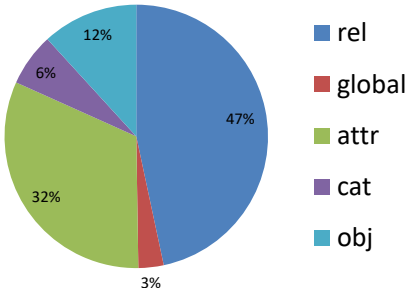
where $\log(d)$ is equal to the entropy of a uniform distribution of size d. Normalized entropy $\bar{e}(x)$ thus measures how close the distribution $p(x)$ is to a uniform distribution of the same dimension. Finally, we keep groups with a normalized entropy smaller than a threshold empirically set to T=0.9. This selects all benchmark's questions, but further work is done in order to select specific answer classes for each group.

5.2.2 *Out-of-distribution setting*

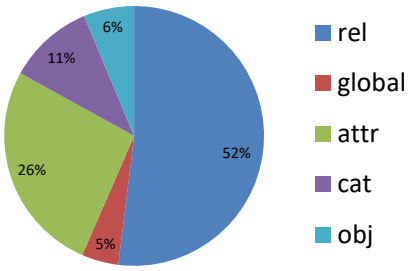
METRICS We introduce a shift in distribution by selecting a subset of answer classes for each question group according to their frequencies, and introduce three different metrics according to which classes are used for evaluation. All these metrics are defined over the aforementioned imbalanced local groups. Figure 5.1 illustrates how the subsets are selected using the example answer histogram of question group *objects on walls*.

- *Acc-tail*: the accuracy on OOD samples, which are the samples of the tail of the answer class distribution, *i.e.* the rarest answers given the context. We define the tail classes as classes i with $|a_i| \leq \alpha\mu(a)$, where $|a_i|$ is the number of samples belonging to the class i and $\mu(a)$ is the average sample count for the group. We empirically set the parameter α=1.2, and in Section 5.3.2 we analyze and illustrate the impact of the choice of α on *Acc-tail*. Figure 5.1 provides an example of such a tail question — we can see that the answer *Star* is rare in this group, therefore it belongs to the tail set like the other answers shown in orange.
- *Acc-head*: the accuracy on the distribution head for each local group, given as the difference between the whole group and its tail (blue answers in Figure 5.1).
- *Acc-all*: the overall (classical) accuracy over all GQA-OOD samples, *i.e.* the in-domain accuracy. In Figure 5.1, this corresponds to the blue and orange answers.

DATASET STATISTICS Table 5.1 provides statistics of the proposed benchmark. We also analyzed the nature, distribution and diversity of the questions w.r.t to GQA, and demonstrate that it preserves the original question diversity. Figure 5.4a and Figure 5.4b show the distribution of question structure type as defined in GQA on the validation split. As one can observe, the process implemented to construct GQA-OOD does not alter

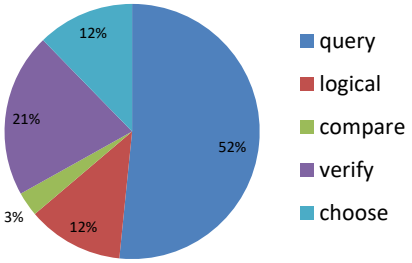


(a) GQA

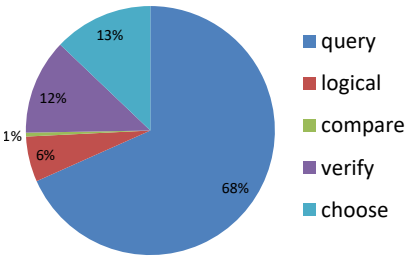


(b) GQA-OOD (tail)

Figure 5.3 – Distribution of the semantic types as defined in GQA. *rel* = relation; *attr* = attribute; *cat* = category; *obj* = object.



(a) GQA



(b) GQA-OOD (tail)

Figure 5.4 – Distribution of the structural types as defined in GQA.

the question diversity of the original split. However, the proportion of open questions – 'query' in Figure 5.3a and Figure 5.3b – has increased in GQA-OOD. Indeed, open questions – such as color questions – generally accept a wider diversity of answer, therefore it is prone to be more imbalanced. On the contrary, other types such as 'choose', 'verify' or 'compare' usually accept only two possible answers and are easier to balance. Figure 5.3a and Figure 5.3b details the distribution of the structure types.

5.2.3 *Discussion and limitations*

DIFFERENCE WITH VQA-CP2 The VQA-CP2 dataset was a first of its kind and paved the way for follow-up work on bias reduction methods in VQA. However, its construction is conceptually different from our work, partially due to the restrictions of the base dataset VQAv2 w.r.t. to GQA, but also due to key design choices. Lacking annotations on group structure in the base dataset, questions are grouped according to their first words and the ground-truth answer. The shift is created by splitting according to types. In contrast, our proposed GQA-OOD dataset allows fine-grained analysis of the generalization behavior of a VQA model by (i) question group, and via (ii) different metrics corresponding to different amounts of shifts (*acc-tail* vs. *acc-head*) in out- and in-distribution settings, and (iii) even through the possibility of continuous evaluation along different operating points (see Figure 5.5). In addition, VQA-CP2 comprises only two splits (train and test), lacking the possibility of validating model hyperparameters (*cf.* Chapter 4). Our GQA-OOD dataset contains a validation set with a shift w.r.t. to the train set, which allows validating

hyperparameters in OOD settings. Finally, unlike VQA-CP, our proposed dataset requires models to be trained on the existing GQA train split. This forces models to reduce bias in their test results while being exposed to natural tendencies and biases captured in the training corpus, favoring work on bias reduction through methodology instead of through cleaning of training data.

LIMITATIONS The proposed benchmark is built on GQA, whose questions have been automatically generated, resulting in a limited vocabulary and a synthetic syntax (*cf.* Chapter 4). While the images are natural and real, one might argue, that the questions are not "*in the wild*". However, the benefits of the synthetic nature of the questions largely out-weight its limitations. In particular, this offers a better control on the data and excludes unmodelled external knowledge, which leads to a better evaluation of reasoning abilities. We made the source code publicly available [3], and we encourage the field to use it to study robustness in OOD settings.

5.3 EXPERIMENTS

In our experiments we used several SOTA VQA models, and we compared the proposed GQA-OOD benchmark to the standard benchmarks VQAv2 (Goyal et al. 2017), GQA (Hudson et al. 2019b) and VQA-CP2 (Agrawal et al. 2018). The line-up includes recent models with object-level attention and two Transformer-based model, as well as two blind baseline models (see Chapter 3 for details). We also evaluate a visual oracle model with a perfect sight, *i.e.* taking as input the question and a set of ground truth objects directly taken from the annotation of GQA [4]. It allows evaluating the performance of a model without the imperfection of the visual extractor. It is based on a compact VL-Transformer architecture (*cf.* Section 3.5).

TRAINING DETAILS All models evaluated on GQA and GQA-OOD have been trained on the balanced training set of GQA, and validated on the validation split. When available, we provide the standard deviation computed over at least four different seeds. For MCAN (Yu et al. 2019) and UpDn (Anderson et al. 2018) we use publicly available implementations at `https://github.com/MILVLG/openvqa`. LSTM (Hochreiter et al. 1997), UpDn, RUBi (Cadene et al. 2019), BP and LM (Clark et al. 2019) are trained during 20 epochs with a batch size equals to 512 and Adam (Kingma et al. 2014) optimizer. At the beginning of the training, we linearly increase the learning rate from $2e^{-3}$ to $2e^{-1}$ during 3 epochs, followed by a decay by a factor of 0.2 at epochs 10 and 12. MCAN is trained during 11 epoch with a batch size equals to 64 and Adamax (Kingma et al. 2014) optimizer. At the beginning of the training, we linearly increase the learning rate from $1e^{-4}$ to $2e^{-1}$ during 3 epochs, followed by a decay by a factor of 0.2 at epochs 10 and 12. For MMN, we use the author's implementation and trained model [5]. LXMERT (Tan et al. 2019) is pre-trained on a corpus combining images and sentences from MSCOCO (Lin

3. `https://github.com/gqa-ood/GQA-OOD`
4. As Ground Truth (GT) annotations (scene-graphs) are only available for the train and validation split, we do not evaluate VIS-ORACLE on the testdev split.
5. Available at `https://github.com/wenhuchen/Meta-Module-Network`.

Model	Baseline benchm.	Proposed benchmark (Acc-tail)		
	Tot. Acc.	α=1.2	α=0.5	α=0.3
UpDn (Anderson et al. 2018) + *bal*	$60.7_{\pm 0.4}$	$45.4_{\pm 0.3}$	$33.8_{\pm 0.5}$	$24.6_{\pm 0.5}$
UpDn (Anderson et al. 2018) + *all*	$59.8_{\pm 0.1}$	$41.9_{\pm 0.1}$	$29.5_{\pm 0.3}$	$18.3_{\pm 0.6}$
Δ (relative):	-1.4%	-7.7%	-12.9%	-25.7%

Table 5.2 – We compare two different VQA models based on UpDn (Anderson et al. 2018), one of which has been trained on a split known to be biased (UpDn (Anderson et al. 2018)+*all*), and evaluate the proposed metric's capacity to detect this bias. All scores in % on the GQA-OOD val split.

et al. 2014) and VisualGenome (Krishna et al. 2017). As GQA is built upon VisualGenome, the original LXMERT pre-training dataset contains samples from the GQA validation split. Hence, we remove those samples before pre-training in order to correctly evaluate on the GQA and GQA-OOD validation split. The VIS-ORACLE model is based on a tiny version of the LXMERT architecture (Tan et al. 2019), where we set the hidden size to 128 and the number of per-layer heads to 4. This perfect-sighted model takes as input objects extracted from the ground-truth GQA annotation (Hudson et al. 2019b). Each object is constructed using one hot vectors encoding its class, its attributes and its in and out scene graph relationships.

5.3.1 *Evaluation of the proposed metric*

We believe that a good evaluation metric satisfies at least two properties: it is easy to interpret, and it provides an estimate for the quality targeted by the evaluation. We argued above on the merits of our proposed tail accuracy (*acc-tail*) as a way of estimating VQA performance less influenced by bias. In what follows, we achieve this by an experimental validation of the metric. To this end, we compared two different VQA models, one of which has been trained in a way known to be biased. In particular, we trained UpDn, known to capture training set biases (Agrawal et al. 2018), on the GQA and GQA-OOD validation splits. The first version, UpDn+*bal*, is trained on the widely used balanced training set of GQA, which we had also used for all other experiments in this paper. This training set had been created by smoothing the question distribution in order to mitigate dataset biases (Hudson et al. 2019b). The second one, UpDn+*all*, is trained on the raw and unbalanced GQA training set, which leads to more spurious biases than the balanced version. As the unbalanced set is ten times bigger than the balanced one, we split it in ten subsets and provide the average score.

Results are given in Table 5.2, comparing two different metrics, namely the classical total accuracy and our GQA-OOD *acc-tail* metric, with three different values for the α hyperparameter. First, we observe that the two versions of UpDn obtain similar scores on GQA overall — the relative difference is only 1.4%. This is not a surprise, the classical metric is influenced by biases. As expected, the two VQA models behave differently on our proposed *acc-tail* metric: the model trained on the unbalanced training set is outperformed by the balanced one by a large margin. Moreover, the score difference

Model	Uses image	acc-all	acc-tail	acc-head	Δ
Quest. Prior	✗	21.6	17.8	24.1	35.4
LSTM (Antol et al. 2015)	✗	30.7	24.0	34.8	45.0
UpDn (Anderson et al. 2018)	✓	46.4±1.1	42.1±0.9	49.1±1.1	16.6
MCAN (Yu et al. 2019)	✓	50.8±0.4	46.5±0.5	53.4±0.6	14.8
BAN4 (Kim et al. 2018)	✓	50.2±0.7	47.2±0.5	51.9±1.0	**9.9**
MMN (Chen et al. 2021)	✓	52.7	48.0	55.5	15.6
LXMERT (Tan et al. 2019)	✓	**54.6**	**49.8**	**57.7**	15.9

Table 5.3 – Comparison of several VQA models on the GQA-OOD testdev split. *Acc-tail*: OOD settings, *Acc-head*: accuracy on most probable answers (given context), scores in %.

increases with decreasing α, (*i.e.* when the metric focuses on the rarer and rarer question-answer pairs, providing valuable evidence that *acc-tail* is indeed well suited for measuring VQA performance undisturbed by bias dependencies.

5.3.2 *Analysis of VQA model error distributions*

The GQA-OOD benchmark allows us to perform an analysis of the error prediction distributions for various VQA models as shown in Table 5.3 and Table 5.4. We provide the three metrics introduced in Section 5.2: *acc-tail*, *acc-head* and *acc-all*. We also measure the difference $\Delta(tail,head) = \frac{acc\text{-}head - acc\text{-}tail}{acc\text{-}tail}$ to illustrate how much is the error prediction imbalanced between frequent and rare answers.

MODELS FAIL ON RARE QUESTION-ANSWER PAIRS We can see that VQA models (dramatically) fail to generalize to infrequent association of concepts. The two blind models (Question Prior and LSTM in Table 5.3) obtain the highest gap between *acc-tail* and *acc-head*, explained by the fact that they uniquely rely on question biases. The Δ score indicates that UpDn, MMN, MCAN, BAN4 and LXMERT also struggle (in a lesser extent) to generalize to the less frequent question-answer pairs. Nevertheless, we observe that the Transformer-based architecture combined with large-scale BERT training, LXMERT, outperforms all models on the *acc-tail* metric, confirming its superiority. This is corroborated by Hendricks et al. (2018), who show that pretrained Transformers improve OOD robustness in NLP.

In contrast to our proposed *acc-tail* metric, the metric *acc-all*, close to the standard VQA metric, does not reflect the true model's performances, since it is mechanically increased by the high scores obtained on the most frequent question-answers. This confirms the need for a two-in-one evaluation: measuring the out- and in-distribution performance scores, as we propose.

VISUALIZING THE GENERALIZATION BEHAVIOR The definition of what constitutes a "rare" answer, i.e. the size of the tail, depends on the parameter α. In Figure 5.5, we analyze how VQA model prediction errors (*acc-tail*) depend on this definition, i.e. how

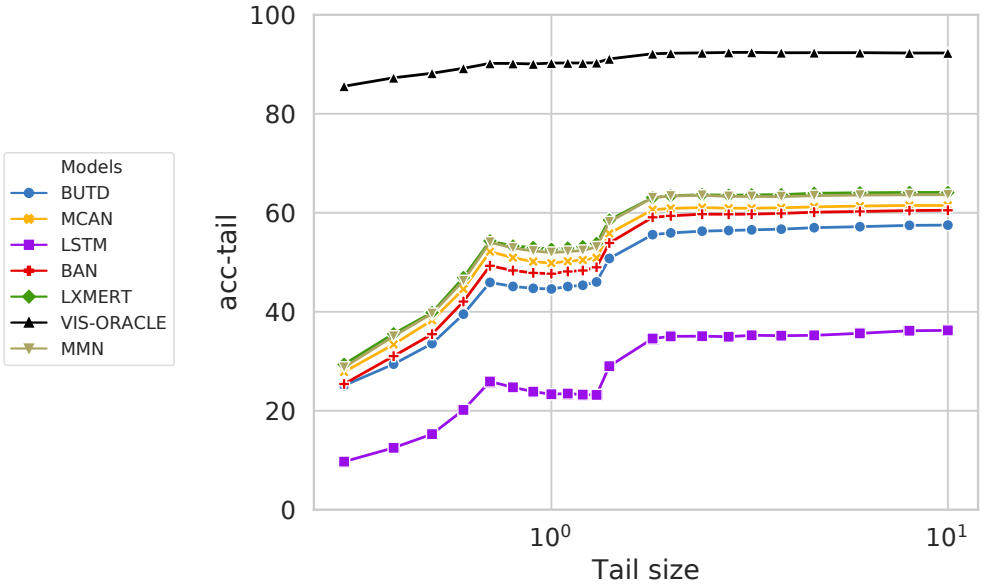


Figure 5.5 – Performance (higher is better) for different definitions of the tail distribution (α parameter values) on the GQA-OOD benchmark. We compare several VQA **models**. The x-axis is in log-scale.

models behave w.r.t. to questions whose answers are more and more rare. Increasing α increases the tail — in the extreme case it is equal to the whole distribution (right side of the plot). With small α, only the most infrequent question-answer pairs are evaluated (left side of the plot). All models follow the same dynamic: starting from a tail size which represents roughly half of the question-answer pairs, tail accuracy starts to linearly decrease until reaching a dramatically low score (about 30 pts lower than the overall accuracy). An exception is VIS-ORACLE: its dynamics is nearly flat, prediction error is almost decorrelated from answer rareness. This provides evidence that a model using perfect visual input is able to learn reasoning with significantly decreased dependency on dataset biases.

We complement this analysis by measuring the confusion between *head* and *tail* as a function of α, shown in Figure 5.6, which provides insights on the causes of the generalization failure observed in Figure 5.5. The confusion corresponds to the proportion of questions where the model predicts a *head* answer with a *tail* GT answer. When plotting the confusion versus α, we decrease the size of the tail set (*i.e.* we keep only the rarest question-answer pairs) while keeping the head set unchanged. For α=1.2, LXMERT confuses answers for 25% of questions, which increases up to 42% for α=0.3. Similar behavior is observed for the other models, but interestingly *not* for VIS-ORACLE, where the curve is nearly flat, again providing evidence for a low dependency on statistical biases in the training set. As a side note, we will show in Chapter 9 that initializing LXMERT weights with VIS-ORACLE allows boosting the accuracy on *acc-tail*.

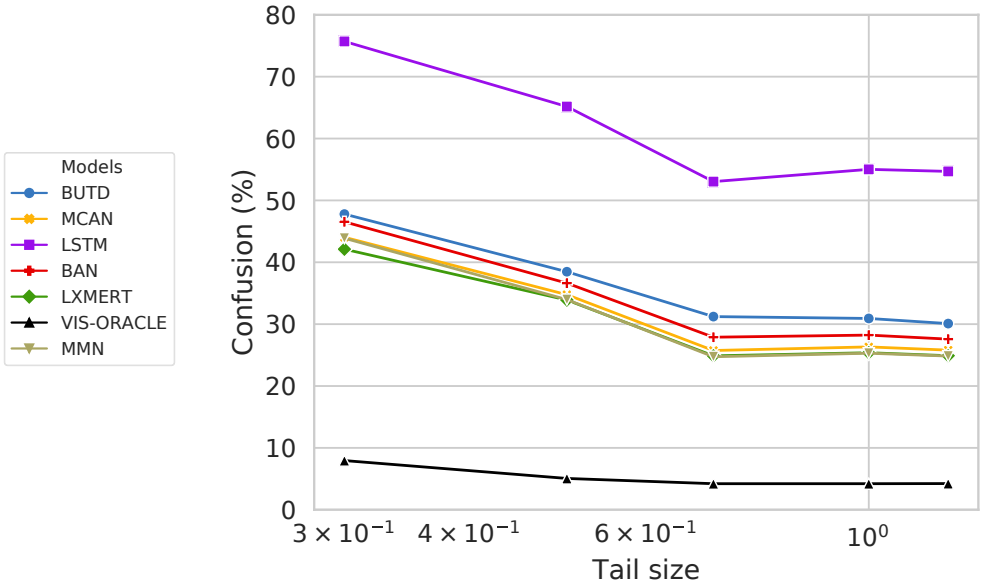


Figure 5.6 – Head/tail confusion (lower is better) for different definitions of the tail distribution (α parameter values) on the GQA-OOD benchmark. We compare several VQA **models**. The x-axis is in log-scale.

EXPLOITING BIASES VS. REASONING It is difficult to assess, whether a model reason or not, in particular since the term *reasoning* has various different definitions (*cf.* Chapter 2). However, it is certain that using statistical biases cannot be considered reasoning, but should rather be denoted as "*educated guesses*" (Hudson et al. 2019b) or *biased* answers. Using the proposed GQA-OOD benchmark, we explore the estimation of three reasoning labels qualifying the mode of operation a model uses for a given input: *bias*, *reason* and *other/unknown*. In absence of GT information, we propose to estimate these labels from proxy rules: a VQA model is estimated to *reason*, when it correctly predicts an answer, which is rare in GT and rare in prediction; it is considered *biased*, when it wrongly predicts an answer, which is *frequent* in its prediction and *rare* in GT.

Figure 5.7-a shows the calculation of these labels based on the distribution of the *head* and *tail* labels of each answer in the predictions (rows) and GT (columns) for LXMERT on the validation split of GQA-OOD. We add a *borderline* label representing the fuzzy frontier between reasoning and bias exploitation [6]. In Figure 5.7-b, we show the distribution of these reasoning labels over the different GQA structural question types: *verify*, *choose*, *compare* and *query*. We observe that LXMERT seems to "*reason*" on the *verify*, *choose* and *logical* questions, which are binary questions, while *compare* [7] and *query* questions are the most prone to bias exploitation. From this, we conclude that future efforts on improvements of model capacities to answer open questions (*e.g* typed as *query*) should be particular fruitful.

6. *head*: α>1.2, *borderline*: 0.7<α<1.2, *tail*: α<0.7.
7. only 1% of the tail questions are typed as *compare*.

	Head		Borderline		Tail	
	C	W	C	W	C	W
Head	30.0%	9.6%	0.0%	3.1%	0.0%	5.3%
Borderline	0.0%	5.5%	6.3%	2.1%	0.0%	3.2%
Tail	0.0%	5.0%	0.0%	0.8%	11.6%	1.3%

C=Correct, W=Wrong

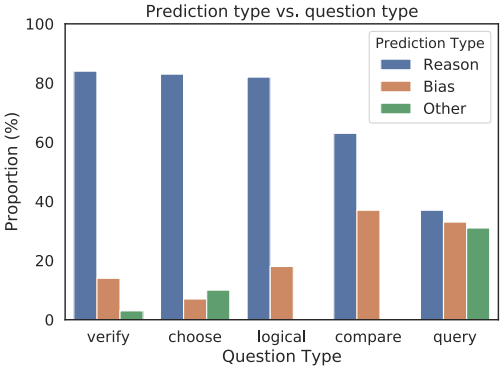
Rows=predicted labels, columns=GT labels

Blue=Model is estimated to reason

Orange=Model is estimated to exploit bias

Green=Unknown label

(a)



(b)

Figure 5.7 – We estimate "*reasoning labels*": the model is estimated to *reason*, when it correctly predicts an answer rare in GT and rare in prediction; it is considered *biased*, when it wrongly predicts an answer, which is *frequent* in its prediction and *rare* in GT. All values are computed over the GQA-OOD validation split. The matrix (a) shows the joint distribution of predicted and GT classes. (b): Distribution the estimated reasoning labels over the GQA (Hudson et al. 2019b) question types for the LXMERT (Tan et al. 2019) model. The model often predicts a biased answer on the *query* and *compare* questions while there is evidence that it may reason on *verify*, *choose* and *logical* questions.

5.3.3 *Re-evaluating bias-reduction methods*

We use the proposed benchmark to re-evaluate several bias-reduction methods, which have been initially designed on the VQA-CP dataset. As these methods were designed to be model-agnostic, we use them together with the UpDn architecture:

RUBI (Cadene et al. 2019) adds a question-only branch to the base model during training to prevent it from learning question biases. This branch is omitted during evaluation. To better analyze bias dependencies, we also study a modified version of RUBi, which we refer to as RUBi+QB below. In this variant, the question-only branch is kept during evaluation.

BP (Clark et al. 2019) is similar to RUBi but differs by directly taking training set statistics to infer question type biases during training [8]. The question type biases are fused with the base model predictions using a product of experts, and removed during testing.

LM (Clark et al. 2019) is an improved version of BP. In this version, the question bias is dynamically weighted by the base model in order to control its influence. In the original setup, an entropy penalty is added to the loss to prevent the model to ignore the bias. Nevertheless, when training on GQA, we obtain better results without this penalty.

8. VQAv2: biases are over question types; GQA: local groups.

Technique	acc-all	acc-tail	acc-head	Δ
UpDn (Anderson et al. 2018)	46.4±1.1	**42.1**±0.9	49.1±1.1	16.6
+RUBi+QB	**46.7**±1.3	**42.1**±1.0	**49.4**±1.5	17.3
+RUBi (Cadene et al. 2019)	38.8±2.4	35.7±2.3	40.8±2.7	14.3
+LM (Clark et al. 2019)	34.5±0.7	32.2±1.2	35.9±1.2	**11.5**
+BP (Clark et al. 2019)	33.1±0.4	30.8±1.0	34.5±0.5	12.0

Table 5.4 – Comparison of several VQA bias reduction techniques on the GQA-OOD testdev split. *Acc-tail*: OOD settings, *Acc-head*: accuracy on most probable answers (given context), scores in %. Bias reduction techniques are combined with UpDn (Anderson et al. 2018) model.

Surprisingly, none of the three bias-reduction methods succeed to improve *acc-tail* (*cf.* Table 5.4). They even deteriorate *acc-head*. This is unexpected as they have been designed to overcome the dependency on question type biases. For further analysis, we evaluate RUBi while keeping the question-only branch during testing (RUBi+QB). As expected, it outperforms RUBi on *acc-head*, indicating it has better captured frequent patterns. However, it also outperforms RUBi on the OOD settings, demonstrating that preventing from learning frequent patterns does not necessarily increase performances on rare samples.

We provide a visualization of the generalization behavior on bias-reduction methods in Figure 5.8. For BP, LM and, to a lesser extent, RUBi, we observe that the right side of the curve has flattened, indicating that overall accuracy, dominated by frequent question-answer pairs, has been reduced by bias-reduction. The left side of the curve, however, corresponding to rare samples, remains almost unchanged, revealing that these methods have somewhat succeeded in preventing the base model from learning dataset biases. As a comparison, the LSTM model in Figure 5.5 performs worse than UpDn but conserves the same frequent/rare imbalance. We observe that RUBi+QB responds the same way as UpDn, confirming the effect of bias-reduction; looking at *head/tail* confusion in Figure 5.9, the result is even more pronounced. In short, we demonstrate the effectiveness of bias reduction methods in preventing the base model from learning salient properties of the training set, and occasionally reducing the dependency toward dataset biases. However, this does not necessarily help the model to learn the subtle distributions, required for generalization and for learning to reason.

5.3.4 *Comparison with other benchmarks*

We compare the proposed GQA-OOD benchmark with the following three standard VQA datasets:

GQA (BALANCED VERSION) (Hudson et al. 2019b) We compare with the overall accuracy and the distribution score on the GQA testdev split. The distribution score is obtained by measuring the match between the true GT answer distribution and the predicted distribution.

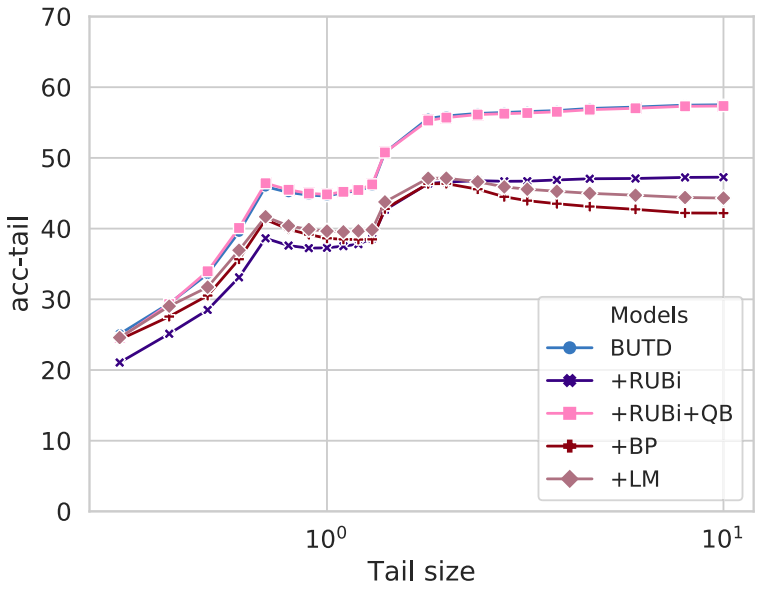


Figure 5.8 – Acc-tail performance, as in Figure 5.5), but for different bias-reduction methods on top of UpDn (Anderson et al. 2018).

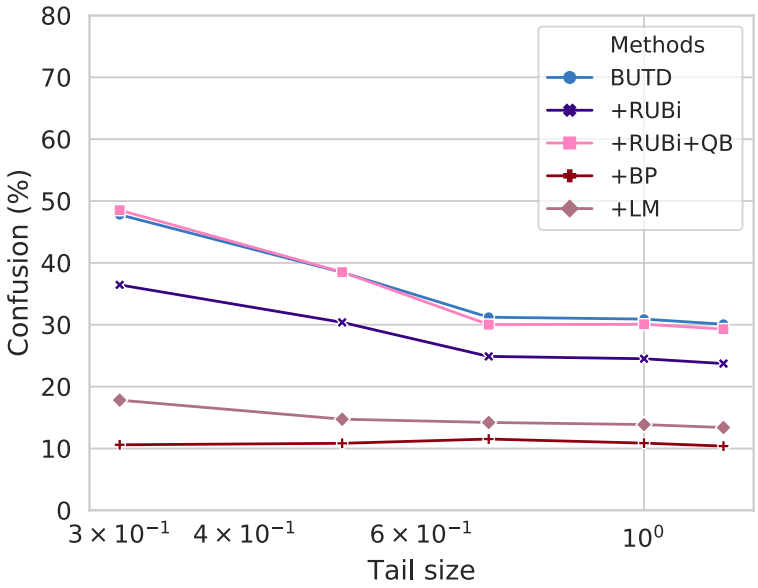


Figure 5.9 – Head/tail confusion, as in Figure 5.6), but for different bias-reduction methods on top of UpDn (Anderson et al. 2018).

Model	VQA2	GQA		VQA-CP2	GQA-OOD
	overall	overall	dist.	overall	acc-tail
Q. Prior	32.1	*27.0*	*55.6*	8.8	*17.8*
LSTM (Antol et al. 2015)	43.0	*39.1*	*3.6*	*22.1*	*24.0*
UpDn (Anderson et al. 2018)	63.5	*51.6*±0.3	*1.8*	40.1	*42.1*±0.9
MCAN (Yu et al. 2019)	***66.1***	*56.3*±0.2	*1.6*	***42.5***	*46.5*±0.5
BAN4 (Kim et al. 2018)	65.9	*54.7*±0.4	*1.6*	40.7	*47.2*±0.5
MMN (Chen et al. 2021)	-	**59.6**	1.8	-	*48.0*
LXMERT (Tan et al. 2019)	**69.9**	***59.6***	***1.5***	-	***49.8***
UpDn (Anderson et al. 2018)	**63.5**	*51.6*±0.3	*1.8*	40.1	***42.1***±0.9
+RUBi+QB	-	***51.9***±1.1	***1.7***	***47.6***±3.7	***42.1***±1.0
+RUBi (Cadene et al. 2019)	61.2	*43.6*±2.0	*1.9*	44.2	*35.7*±2.3
+LM (Clark et al. 2019)	56.4	*39.7*±0.7	*2.1*	52.0	*32.2*±1.2
+BP (Clark et al. 2019)	63.2	*39.6*±0.3	*2.2*	39.9	*30.8*±1.0

Table 5.5 – We compare the proposed *acc-tail* metric with other benchmarks. Results computed on the testdev split of GQA-OOD and GQA, the test split of VQA-CP2 and the VQAv2 validation split. Values in italic: trained and tested by ourselves.

VQAv2 (Goyal et al. 2017) We compare with overall accuracy on the VQAv2 validation split.

VQA-CP2 (Agrawal et al. 2018) We compare with the accuracy on the test split, which has been designed to measure sensitivity to language bias.

COMPARISON WITH GQA AND VQAV2 In Table 5.5, we compare our *acc-tail* score with the other benchmarks. We can see that overall accuracy on GQA and VQAv2 is not sufficient to fully characterize the VQA performances. Our evaluation in OOD settings is the only one to reveal that even SOTA models struggle on infrequent question-answer pairs. The best-performing model LXMERT looses about 10 points in the OOD setting. Our metric also unveils that, despite performing on-par with LXMERT on GQA overall, MMN struggles more on infrequent question-answer pairs. Finally, we argue that *acc-tail* is easier to interpret than the error distribution measure defined in GQA.

COMPARISON WITH VQA-CP2 Comparing *acc-tail* to VQA-CP2 overall accuracy, we observe similar scores on standard VQA models, but a completely different behavior for bias-reduction methods. While they do not improve the scores in the OOD setting (*cf.* Section 5.3.2), they achieve strong performances on VQA-CP2. The score of LM stands out, achieving the highest overall accuracy on VQA-CP2 (52.0%) but one of the lowest *acc-tail* on GQA-OOD (33%), with similar behavior for RUBi and BP. In short, while VQA-CP2 measures to what extent a VQA model struggles to generalize to a specific unseen distributions, the VQA-CP2 evaluation does not reflect the model behaviour on rare question-answer pairs.

5.4 VISUALISING PREDICTIONS

In order to give a better insight about the benchmark's goals and possibilities, we provide additional samples extracted from the GQA-OOD validation split. In Figure 5.10 and Figure 5.11, we show two question-answer pairs belonging to the tail. The histogram represents the answer frequency measured over the set of all questions belonging to the group of the given question. We colored the answers according to their label, head or tail. First, we can observe that the histogram is very imbalanced, which motivates the GQA-OOD approach. Second, in the caption, we provide the predicted answer for each one of the evaluated model. One can notice that the predictions are diverse, showing various degree of bias dependency. However, all models are mostly relying on context biases, as shown in Figure 5.12. Finally, in Figure 5.13, we show a question-answer pair labelled as head, where all models (excepted the blind LSTM) are correct.

5.5 DISCUSSION AND CONCLUSIONS

Going beyond previous attempts to reduce the influence of dataset biases in VQA evaluation, our proposed GQA-OOD benchmark allows to *both* evaluate (1) whether models have absorbed tendencies in the training data, and (2) how well they generalize to rare/unseen question-answer pairs. This was made possible by (i) a thorough choice of imbalanced question groups, (ii) a new set of metrics and finally, (iii) by allowing to control the amount of distribution shift via the hyperparameter α. We have provided evidence that the benchmark and metric measure performance and dependency on dataset bias. Our experiments have also shown that neither conventional SOTA VQA models nor dedicated bias reduction methods succeed in all aspects of the proposed evaluation benchmark. We hope that this sheds light on the current shortcomings in vision and language reasoning, and we hope that GQA-OOD will contribute to the emergence of new models, less prone to learning spurious biases and more reliable in real-world scenarios.

*
* *

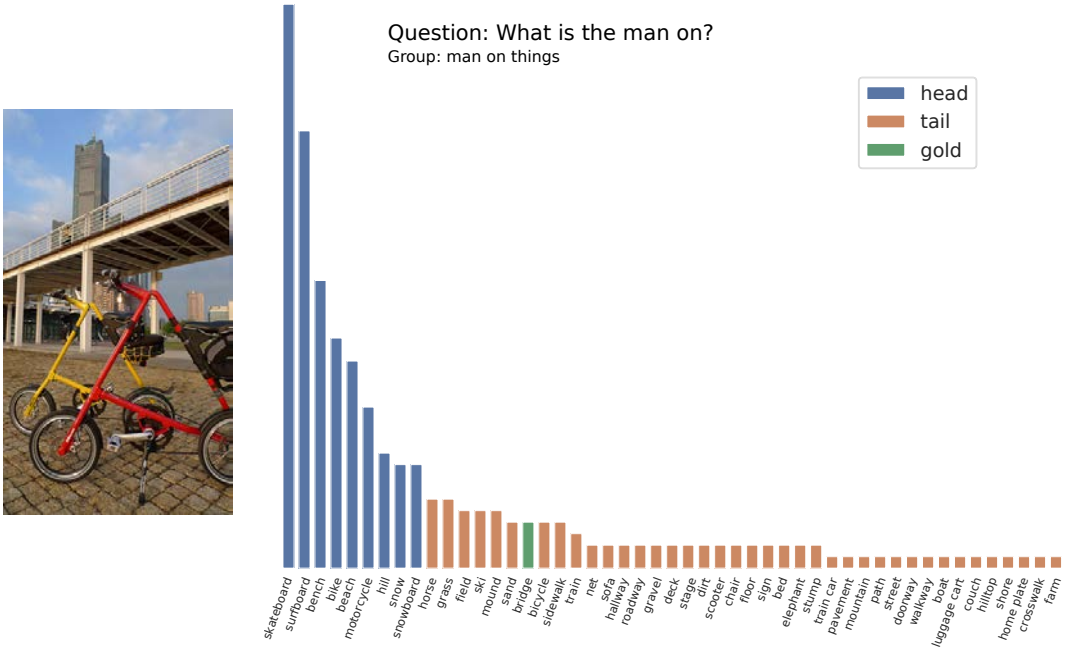


Figure 5.10 – Tail sample from the GQA-OOD validation split. Question:*What is the man on?*. Answer:*bridge*. The evaluated models have predicted: LSTM=*skateboard*; UpDn, MCAN = *bike*; BAN, UpDn+LM, MMN, UpDn+RUBI, UpDn+BP = *bicycle*; LXMERT, ORACLE-VIS = *bridge*. The histogram represents the answer frequency measured over the set of all questions belonging to the question group.

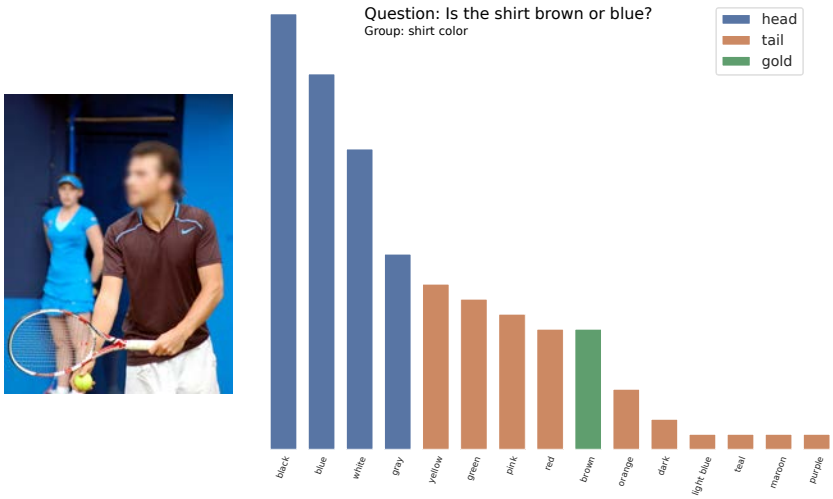


Figure 5.11 – Tail sample from the GQA-OOD validation split. Question:*Is the shirt brown or blue?*. Answer:*brown*. The evaluated models have predicted: LSTM, BAN, UpDn, UpDn+LM = *blue*; UpDn+RUBI, = *light blue*; MCAN, LXMERT, ORACLE-VIS, MMN, UpDn+BP = *brown*. The histogram represents the answer frequency measured over the set of all questions belonging to the question group.

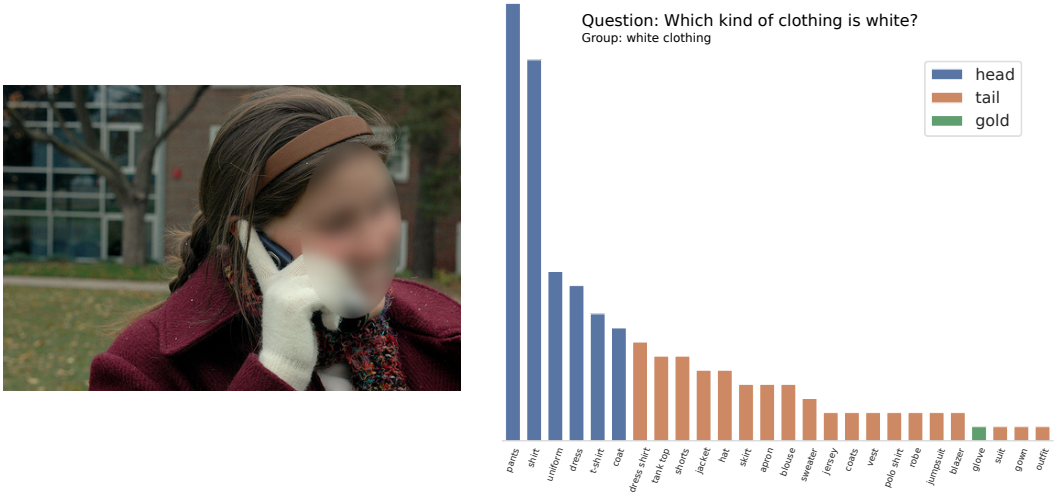


Figure 5.12 – Tail sample from the GQA-OOD validation split. Question:*Which kind of clothing is white?*. Answer:*glove*. The evaluated models have predicted: LSTM = *shirt*; LXMERT, UpDn, BAN, MMN, UpDn+RUBI = *coat*; MCAN = *jacket*; UpDn+LM, UpDn+BP = *long sleeved*; ORACLE-VIS = *glove*. The histogram represents the answer frequency measured over the set of all questions belonging to the question group.

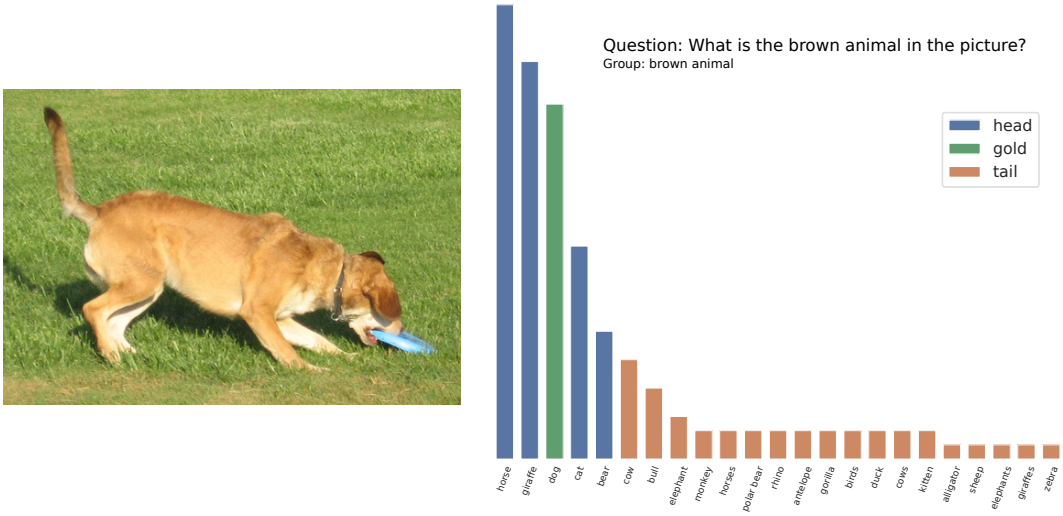


Figure 5.13 – Head sample from the GQA-OOD validation split. Question:*What is the brown animal in the picture?*. Answer:*dog*. The evaluated models have predicted: LSTM = *horse*; BAN, UpDn, UpDn+LM, UpDn+RUBI, MCAN, LXMERT, ORACLE-VIS, MMN, UpDn+BP = *dog*. The histogram represents the answer frequency measured over the set of all questions belonging to the question group.

Part III

ANALYSE: IN SEARCH OF REASONING PATTERNS

INTRODUCTION

Part II has shown that current VQA models are prone to exploiting harmful biases in the data, which can provide unwanted shortcuts to learning in the form of "*Clever Hans*" effects. We demonstrated (in Chapter 5) the necessity to define new ways of evaluating VQA models, going beyond the standard overall accuracy. But more importantly, we highlighted the fact that this bias-related issue is difficult to identify and measure. Indeed, although the VQA bias dependency has already been largely studied and analyzed (*e.g.* see Agrawal (2019)), the issue persists. SOTA VQA models are still bias dependent, and datasets initially designed to measure bias dependency become quickly insufficient. VQA-CP (Agrawal et al. 2018) is probably the most eloquent example: while it allowed to reveal the strong bias dependency of VQA models, it is currently at the origin of new types of biases leading to negative results (in a few words, de-bias methods designed on top of VQA-CP tend to overfit on its specific setup, *cf.* Chapter 5 for details). Summarizing, it appears that designing one (or many) benchmark(s) is not sufficient to solve the bias issue.

BUT THEN, WHY IS THE STUDY OF BIAS SO DIFFICULT? A pessimistic answer would be that designing bias reduction methods generally boils down to a trade-off between seemingly incompatible goals: reaching high accuracy on standard benchmarks (which are biased), or being robust against biases. Thereby, working on bias-reduction would not be an attractive choice because it could steer efforts away from the classic SOTA competition metrics. But this hypothesis is not completely satisfying. First, reaching SOTA accuracy on standard (biased) benchmarks while being robust against biases is, theoretically, not incompatible, but arguably, hard. As an illustration, we observed in GQA-OOD (Chapter 5) that models performing the best in in-distribution settings are also the best performing in out-of-distribution ones (which measure bias-robustness). Second, there is a large amount of work trying to address (with more or less success) bias robustness in VQA (*cf.* Part II), suggesting that this issue is a topic perceived as relevant by the field.

Actually, we rather think that the origin of the problem is simpler. Working on bias robustness is difficult precisely because it is hard to correctly diagnose bias dependencies. Why? Because of the lack of interpretability of VQA models. As models are not interpretable enough, experts have to build their own interpretation of models' predictions, at the risk of overestimating their reasoning capabilities (à la "*Clever Hans*"), and so ignoring bias issues. In a nutshell, it is hard to solve a problem that we do not understand well.

VQA INTERPRETABILITY In the line of recent work in AI explainability (Lipton 2016; Ribeiro et al. 2016), and data visualization (Hohman et al. 2018; Vig 2019a; DeRose et al. 2020), we aim at improving our understanding of VQA model predictions. More precisely, we propose to borrow tools and methods from AI explainability to draw a better picture of the bias issue in VQA, in a complementary way with the benchmark-based evaluations discussed in Part II. For this purpose, we develop a tool (VISQA) and conduct in-depth

analyses of attention mechanisms at work in VQA models, providing cues to answer the following questions: *When is the model relying on biases? What did it learn? What are the conditions for the emergence of reasoning?*

This work on VQA interpretability is directly related to the discussion on VQA evaluation done in Part II, and contributes to designing new VQA methods introduced in Part IV. *The whole part is the result of a collaboration between experts in visual analytics (in particular, Théo Jaunet), and experts in Visual Question Answering systems and Machine Learning.* Part III is organized as follows:

Chapter 6 aims at improving the interpretability of a VL-Transformer VQA model using VisQA, an instance-level visual analytics tool for VQA, developed in collaboration with Théo Jaunet. In particular, we show that analyzing the attention mechanism at work in the VQA model help experts to better judge when it is reasoning or exploiting shortcuts. This work has resulted in an online interactive tool, publicly available at `https://visqa.liris.cnrs.fr`.

Chapter 7 extends the VisQA analysis, conducted at an instance-level, to get a broader view of the behavior of the VL-Transformer at a dataset level. In particular, we focus on the emergence of reasoning patterns at work in the attention layers of the model. We experimentally demonstrate that the reasoning patterns emerge when the training conditions are favorable enough, and in particular when the uncertainty in the visual part is reduced.

This part has led to the publication of the following conference papers:

- Theo Jaunet, Corentin Kervadec, Romain Vuillemot, Grigory Antipov, Moez Baccouche, and Christian Wolf (2021). "VisQA: X-raying Vision and Language Reasoning in Transformers". In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)*;
- Corentin Kervadec, Theo Jaunet, Grigory Antipov, Moez Baccouche, Romain Vuillemot, and Christian Wolf (2021c). "How Transferable are Reasoning Patterns in VQA?". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*;

Chapter 6

INVESTIGATING ATTENTION IN TRANSFORMERS

6.1 INTRODUCTION

Attention is at the heart of the VL-Transformer architecture (*cf.* Section 3.5 in Chapter 3). While conceptually simple, it makes possible learning very complex relationships between input items. Making sense of the learned attention and verifying its inner workings is a difficult problem, which is addressed by VisQA. More precisely, VisQA is an instance-based visual analytic tool designed to help domain experts investigate how information flows in a VL-Transformer architecture and how the model relates different items of interest to each other in vision and language reasoning. In this chapter, we propose to use VisQA in order to elucidate *if the so-called attention is informative enough to provide insights on the emergence of reasoning or bias exploitation* in the context of VQA.

For this purpose, we explore the different attention maps, represented as heatmaps, generated by the VL-Transformer for a given question-image pair. The exploration is guided by color codes that convey the intensity of each attention head, *i.e.* whether they focus attention narrowly on specific items, or broadly over the full input set. Complementary dataset-wide statistics are provided for each selected attention head, either globally, or with respect to specific task functions, *e.g. "What is", "Where is", "What color"* etc. (this aspect will be discussed in more detail in Chapter 7). While VisQA is post-hoc, it is also interactive and allows certain modifications to the internal structure of the model. At any time, attention maps can be pruned to observe their impact on the output answer.

In a first part, we motivate the need of an attention visualization in a detailed case study. We show that VisQA improves the interpretability of the VL-Tranformer and helps to better understand the reason of its failure. In a second part, we ask different experts in deep learning, who were not involved in the project nor its design, to evaluate the feasibility of using attention analysis to identify bias exploitation and reasoning in the model. We answer positively, and report experiments with qualitative interviews and results in Section 6.4.

This chapter results from a collaboration with visual analytics experts. The scope of this thesis is the study of bias vs reasoning in VQA and not data visualization itself, hence we let the reader refer to our associated paper (Jaunet et al. 2021) to get all the details on the design of VisQA. However, we encourage the reader to watch the short

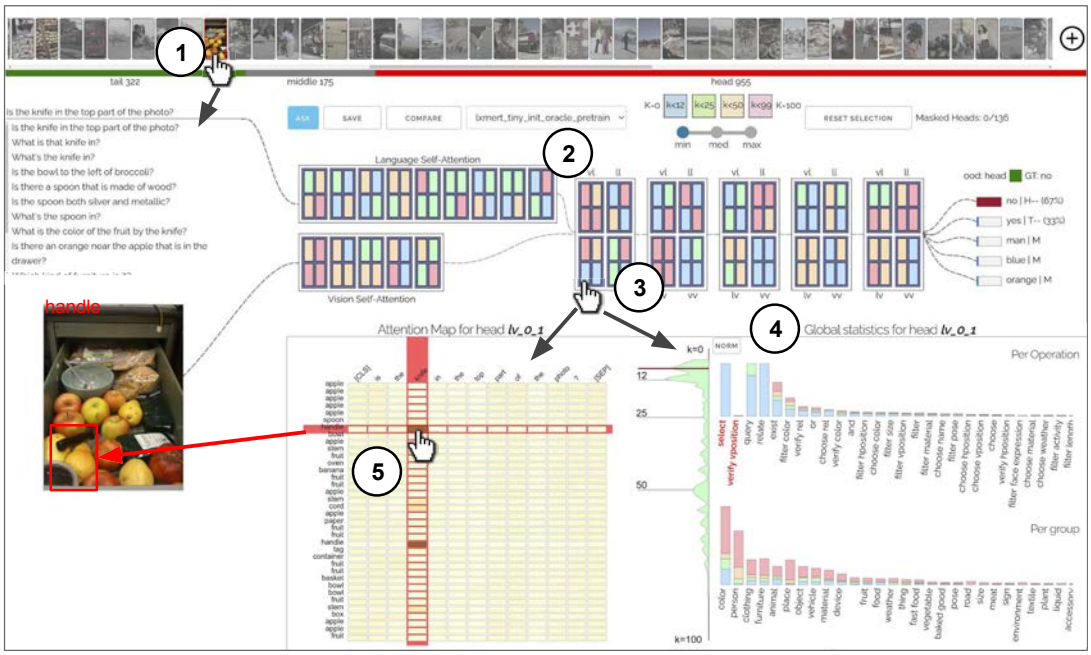


Figure 6.1 – Opening the black box of neural models for vision and language reasoning: given an open-ended question and an image ①, VisQA enables to investigate whether a trained model resorts to reasoning or to bias exploitation to provide its answer. This can be achieved by exploring the behavior of a set of attention heads ②, each producing an attention map ⑤, which manage how different items of the problem relate to each other. Heads can be selected ③, for instance, based on color-coded activity statistics. Their semantics can be linked to language functions derived from dataset-level statistics ④, filtered and compared between different models.

introductory video provided at https://visqa.liris.cnrs.fr/static/assets/demo.mp4, in order to familiarize with VisQA.

CONTRIBUTIONS OF THE CHAPTER

(i) We participate in the conception of VisQA, an interactive visual analytics tool developed by Théo Jaunet, which helps experts to explore the inner workings of transformers models for VQA by displaying models' attention heads in an instance-based fashion.

(ii) We provide a set of visualizations to address bias in VQA systems, by exploring models' performances in real-time with altered attention, and/or by asking free-text questions.

(iii) We conduct an evaluation with domain experts, resulting in insights on the emergence bias in transformers for VQA.

VisQA is available online as an interactive prototype: https://visqa.liris.cnrs.fr, and our code and data are available as an open-source project: https://github.com/Theo-Jaunet/VisQA .

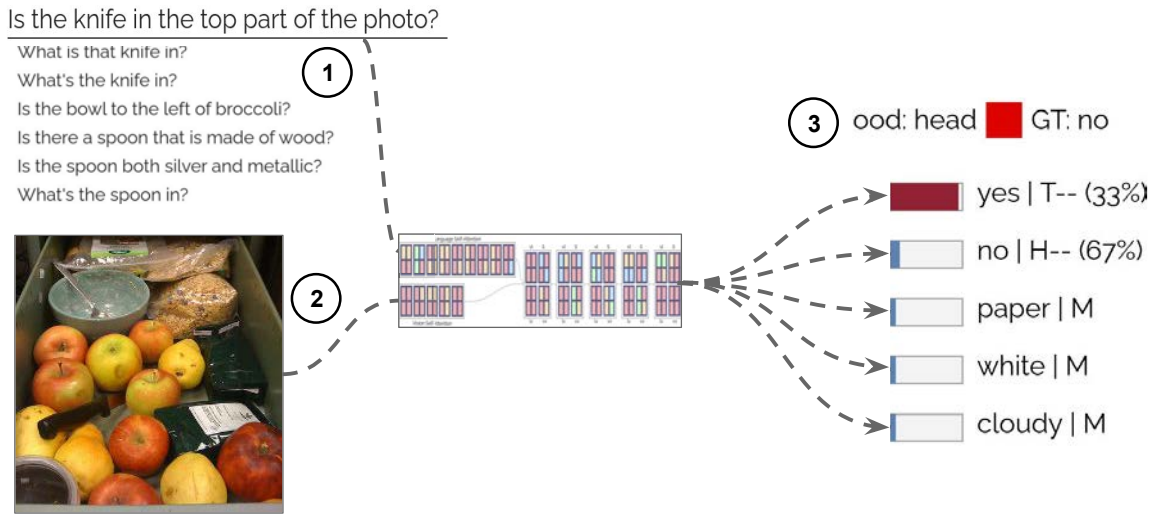


Figure 6.2 – When asked *"Is the knife in the top part of the photo"* ① the VL-Transformer model, with the image of a knife at the bottom ②, incorrectly outputs *"yes"* ③ with more than 95% confidence. While an exploitation of bias can be considered, we can observe that the answer *"yes"* represents only 33% of answers of similar questions over the complete dataset. Thus in-depth analysis of the attention of the model may be required to grasp what led to such a mistake.

6.2 A SHORT INTRODUCTION TO VISQA

6.2.1 *A visual analytics tool for interpretability of DL*

Our work is related to building visual analytics tools for interpretability of DL. DL models are *white boxes* [1], which are generally hardly interpretable. Prior work focused on the analysis of image processing models, known as CNN, by exposing their gradients over the input images (Zeiler et al. 2014). This approach, enhanced with visual analytics (Liu et al. 2016), and provided glimpses on how the neurons of those models are sensitive to different patterns in the input. More recently, CNN have been analyzed through the prism of attribution maps in works such as Activation-atlas (Carter et al. 2019) and attribution graphs (Hohman et al. 2020).

On the other side, NLP with recurrent neural networks, have also been explored through static visualization (Karpathy et al. 2015b) which provided insights, among others, on how those models can learn to encode patterns in sentences beyond their architectures in capacities. Interactive visual analytics works such as LSMTViz (Strobelt et al. 2017), and RetainVis (Kwon et al. 2018) have also addressed the interpretability of those models through visual encoding of their inner parameters, which can then be filtered and completed with additional information. Those parameters are collected during forward pass on models, as opposed to RNNbow (Cashman et al. 2018), which has the

1. Contrary to a black box, all operations conducted in a *white box* are observable.

particularity to focus on visualizing gradients of those models through back-propagation during training.

More recently, models with attention (Vaswani et al. 2017) increasingly gained popularity due to their improvement of state-of-the-art performance, and their attention mechanisms which may be more interpretable than CNN and RNNs. The interpretability of attention models similar to the transformer models used in this work, initially designed for NLP, has also been addressed by visual analytics contributions. Commonly, in works such as (Strobelt et al. 2018; Olah et al. 2016; Vig 2019b), the attention of those models is presented, in instance-based (Hohman et al. 2019) interfaces as graphs with bipartite connections that can be inspected to grasp how input words are associated with each other. Attention Flows (DeRose et al. 2020) addresses the influence of BERT pre-training on model predictions by comparing two transformers models applied to NLP. Similar to VisQA, such a tool displays an overview of each attention head with a color encoding their activity. Those methods are specific to NLP tasks. In this work, we address the challenges provided by the bi-modality of vision and language reasoning, and expand the interpretability of VQA systems which can rely on visual cues or dataset biases. Current practices of VQA visualization include attention heatmaps of selected VL heads based on their activation (Li et al. 2020a) to highlight word/key-object associations, global overview heatmaps of attention heatmaps towards a specific token (Cao et al. 2020), and guided backpropagation (Goyal et al. 2016) to highlight the most relevant words in questions. Following those works, VisQA provides a visualization of every head's attention heatmaps and word/object associations, along with an overview of their activations.

This work is complementary with an alternative direction of work, proposing to increase the interpretability through explanation generation. Thus, Hendricks et al. (2016) and Park et al. (2018) directly supervise their model to generate an explanation in addition to the task-related answer, resulting in an explainable vision-and-language model. Beside, we also propose a complementary approach to Manjunatha et al. (2019), which run rule mining algorithms to explicitly discover VQA shortcuts.

6.2.2 *A tool for investigating hypothesis on bias vs reasoning*

VisQA have been designed to investigate hypotheses on the presence of bias vs reasoning in a VL-Transformer based VQA model (we let the reader refer to Section 3.5 for a detailed review of the aforementioned architecture). In particular, VisQA focuses on attention maps, which are a key feature of transformer-based neural models, as they fully determine relationships between input items. We recommend the reader referring to Jaunet et al. (2021) to familiarize itself with VisQA, or watching the short introductory video available at `https://visqa.liris.cnrs.fr/static/assets/demo.mp4`. We nevertheless briefly recall its main features.

FREE-FORM QUESTIONS By default, VisQA loads the GQA dataset to provide images and questions. But at any time, we can type and ask free-form open-ended questions. Such an interaction allows investigating the model's bias exploitation. For instance, when

asked the following question from the GQA dataset *"Is this a mirror or a sofa"*, the model correctly outputs "*mirror*". However, when asked the following question *"Is there a mirror in this image?"*, the model fails and outputs *"no"*. This suggests that the model might have exploited biases when it answered the first question, which is supported by the fact that in the GQA dataset, *"mirror"* is the correct answer to the question *"Is this a mirror or a sofa"* in 85% of all cases.

VISUAL SUMMARY VisQA allows us to explore the attention maps generated by the VL-Transformer attention heads for a given question-image pair. In order to scope with the relatively high dimension of attentions maps, making them hardly interpretable in a reasonable amount of time, we rely on summarizing each of them to a single scalar. Such a scalar, referred to as *k-number* (Ramsauer et al. 2020), represents the normalized amount of tokens per row summed up to reach a threshold of 90% of energy. A *k-number* close to 0 indicates that the corresponding row has peaky attention focusing on only one column (as seen in Figure 6.3 ①), and a high *k-number* encodes a uniform attention (as in Figure 6.3 ②). Then we combine each of those *k-number* together using either *min*, *median*, or *max* functions. Such functions can be selected in VisQA, depending on the attention maps intensity they want to investigate. VisQA provides this interaction because for a head to have a low *k-number*, the majority of its rows needs to be highly activated. This can shadow attention maps with less than half of their rows with peaky attention. In VisQA, the *k-number* is discretized and color encoded in 4 categories as it follows: k<12 k<25 k<50 k<99. In addition, for each head, we provide global (*i.e.* dataset-level) statistics which will be the subject of the next Chapter 7.

HEAD INTERACTIONS VisQA let the possibility to dynamically interact with the model. More precisely, it makes possible to filter and prune attention head. By clicking on a row, column, or cell, we can filter attention heads to only keep the ones in which the corresponding clicked element has attention above a threshold. This facilitates seeking for heads in which a specific association is expected *e.g.* a word in the question with an object of the image required to answer. It is also possible to prune selected attention heads. Pruning here means that the attention head does not perform any focused attention, but uniformly distributes attention over the full set of items (objects or words). Each row of a pruned attention map is thus the equivalent of an average calculation. This can be used in order to test hypotheses on attention head interpretations, as explored in Section 6.4. The benefit of such pruning is that it preserves the amount of energy in the head, at contrary to an alternative approach where the attention head output is simply zeroed.

SETUP VisQA is based on the VL-Transformer architecture described in Chapter 3. We use the compact version, with an embedding size equals to d=128 and a number of heads per-layers set to h=4. As a recall, in addition to the VQA objective, we train the model parameters also on MS-COCO (Lin et al. 2014) and Visual-Genome (Krishna et al. 2017) images following the semi-supervised BERT (Devlin et al. 2019)-like strategy introduced in (Tan et al. 2019). In particular, the model is trained to perform simple tasks such as recognizing masked words and visual objects, or predicting if a given sentence matches

the question. After pre-training on these auxiliary tasks, the model is fine-tuned on the GQA (Hudson et al. 2019b) dataset with the VQA objective.

6.3 MOTIVATING CASE STUDY

We illustrate the advantages and the power of instance-level visualizations with VisQA on the following case study. It is based on the following input instance, *i.e.* the image given in Figure 6.2(②), and associated question *"Is the knife in the top part of this photo?"* ①. The correct ground truth answer is of course *"No"*, but the model incorrectly answers *"Yes"* ③. We see the frequency of the different possible answers provided in the interface, and observe that the wrong answer *"Yes"* is not the most frequent one for this kind of question as *"No"* is the correct answer 67% of the time, which does not provide evidence for bias exploitation. The objective is to use VisQA to dive deeper into the inner workings of the model.

A first step is to analyze whether the model is provided with all necessary information. While the input image itself does contain all the information required to find the answer, the neural transformer model reasons over a list of objects detected by a first object detection and recognition module – Faster R-CNN (Ren et al. 2015) –, the outputs of which may be erroneous.

IS THE KNIFE DETECTED BY THE VISION MODULE? VisQA provides access to the bounding boxes of the objects detected by the input pipeline. Each bounding box can be displayed superimposed over the input image along with the corresponding object label predicted by the object recognition module. We can observe that the key object *"knife"* lacks a suitable bounding box or class label, it has not been detected. Since this object is required to answer the question for this image, the model cannot predict a coherent answer. However, the question remains why the wrong answer is *"yes"*, corresponding to the presence of a knife.

CAN ATTENTION MAPS PROVIDE CUES FOR REASONING? For the example above, we are interested in checking the correspondence between the question word *"knife"* and the set of bounding boxes, which should provide us with evidence whether the model was capable of associating the concept with the visual object in the scene, which is, of course, not sufficient for correctly answering, but a necessary step. This verification is non-trivial, however, since the model is free to perform this operation in any of the inter-modality layers and heads. VisQA allows to select the different heads, and we could observe that none of the heads provides a correct association. As an example, we can see the behavior of a head in Figure 6.3 ①, which associates the word *"knife"* to various objects, mostly fruits. No other head is found, indicating a more promising relationship.

IS COMPUTER VISION THE BOTTLENECK? From the example above, as well as similar observations in other instances, we conjecture that the computer vision input pipeline (notably, the imperfect object detector) is one of the main bottlenecks in preventing correct reasoning. To validate this hypothesis, we explored training an *Oracle* model

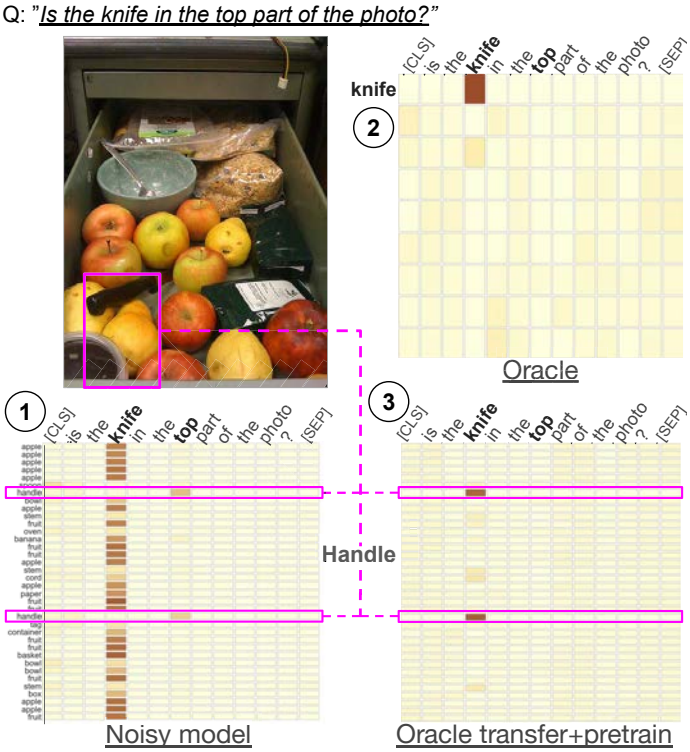


Figure 6.3 – Visualization of a selected vision-to-language head and attention map for two different models. ① the baseline model associates the *"knife"* word with a large number of different objects, including fruit. ② the oracle model learns a perfect association between the word *"knife"* and the *"knife"* object. Head selections are not comparable between models, we therefore checked for permutations.

with perfect sight, which thus takes as input the ground truth objects provided by human annotation instead of the noisy object detections by a trained neural model. This considerably improves the performance of the model, reaching $\approx 80\%$ accuracy on the difficult questions with rare ground-truth answers, compared to $\approx 20\%$ for the standard model reasoning on noisy input. This particularly high difference in performance for questions with rare answers suggests a higher performance in correct reasoning of the oracle model. By loading this model into VisQA, we observe in Figure 6.3 ②, that there exists an attention map which associates the word *"knife"* to a visual object *"knife"*, which, as the reader might recall, is an object indicated through human annotation. This correct association is reassuring, but by itself does not yet guarantee correct reasoning — further exploration is possible, but we will now concentrate on this problem of finding correspondences between words and visual objects and explore this question further. In Section 7.2, we will provide a statistical (dataset-level) analysis of the vision bottleneck.

6.4 Evaluation with domain experts

In order to evaluate the usability and convenience of analyzing attention to get insights on the emergence of reasoning or biases in a VL-Transformer model, we conduct

an experimental study with 6 experts. They have experience in building deep neural networks, but were not involved in the project or the design process of VisQA. We report on their feedback using VisQA to evaluate the decision process of the *Oracle transfer model* (this model will be introduced in Chapter 9), which obtains 57.8% accuracy on GQA, as well as insights they received from this experience. Hypotheses drawn from single instances cannot be confirmed or denied, but as illustrated in the following sections, such a fine-grained analysis aims to provide cues (often unexpected) that will be later explored through statistical evidence in Chapter 7.

6.4.1 *Evaluation Protocol*

For each expert, we conducted an interview session lasting on average two hours. Sessions were organized remotely and began with a training on VisQA, showing step-by-step how to analyze attention maps. During this presentation, experts were able to ask questions. The study then began with questions on 6 problem instances, *i.e.* image/question pairs loaded into VisQA in a browser window on participants' workstations. Those instances were balanced between the prediction failures and successes, head or tail distributions of question rarity as described in GQA-OOD (*cf.* Chapter 5), as well as our estimation on whether the model resorts to bias for this instance grasped using VisQA. VisQA, configured as conditioned during evaluations is accessible online at: `https://theo-jaunet.github.io/visqEval/`. The model outputs were hidden, and the experts were asked to use VisQA to provide an estimate for two different questions: (1) will the model predict a correct answer, (2) what will it be?, and (3) does it exploit biases for its prediction, or does it reason correctly? During this part of the interview, experts were asked to explain out loud what lead them to each decision. Once those questions were completed, post-study questions were asked, such as "*Which part of* VisQA *is the least useful?*", and "*What was the hardest part to understand?*".

RESULTS The ability of experts to predict failures and specific answers of VQA systems has already been addressed through evaluation (Chandrasekaran et al. 2018) under different conditions. The experiment closest to ours is question+image attention (Lu et al. 2016) with instant feedback — similarly to ours, experts were asked to estimate whether a model will predict a correct answer when provided with attention visualizations of the model, and reaching a similar score of $\approx$ 75% accuracy. The difference is that in Lu et al. (2016) attention is overlaid over the visual input, whereas our attention maps allow to inspect reasoning in a more detailed and fine-grained manner, and not necessarily tied to the visual aspects. The similarity in results changes when experts are asked to provide the specific answer predicted by the model: this accuracy drops to 61% in our case, and to 51% in (Chandrasekaran et al. 2018). While our results are promising, they cannot be directly compared to their results due to the different pool and amount of experts. Future work will address studies on a larger number of human experts.

More importantly, our work focuses on qualitative results of bias estimation in which experts obtained a precision of 75% on whether the model exploited any bias. We extracted the ground truth estimate by comparing the rarity of the question, following Chapter 5. These results are encouraging, as they provide a first indication that the

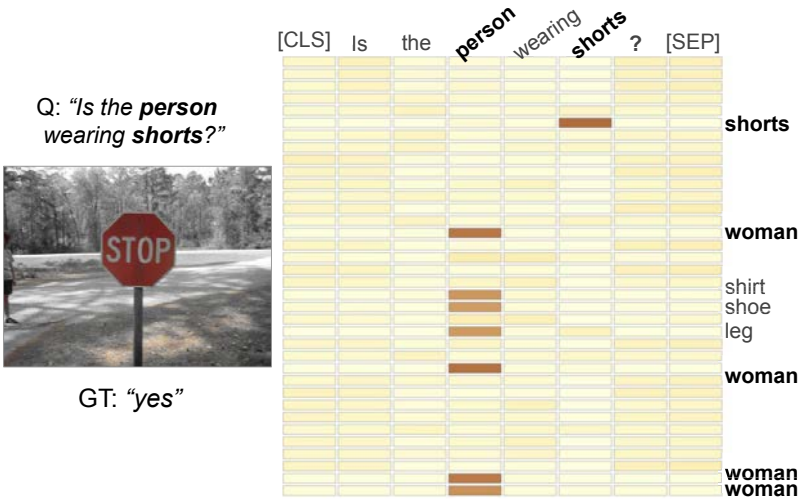


Figure 6.4 – When asked *"Is the person wearing shorts?"*, the *oracle transfer* model successfully answers *"yes"*. It can be observed in its first Language-to-Vision attention maps, that the word *"shorts"* (column) is strongly associated with the object *"shorts"* (row). The same phenomenon is also observed for the word *"person'*, strongly associated with objects labeled as *"woman"* among others.

reasoning behavior of VL models can be examined and estimated by human experts with VisQA. While 75% of performance reasoning vs. bias is not a perfect score, it is also far away from the random performance of 50%, which is important given the large capacity of these models, which contain millions of trainable parameters.

6.4.2 *Object Detection and Attention*

To provide an answer, a model must first grasp which objects from the image are requested and thus are essential to focus on. Such an association needs to occur early in the model as those objects are needed for further reasoning. The experts widely observed high intensity in the first language-to-vision (LV) layer. As illustrated in Figure 6.4, when asked *"Is the person wearing shorts?"*, the attention map *LV_0_1* has peaky activations in the columns *"person"* and *"shorts"*. This can be interpreted as the model correctly identifying with its self-attention for language that those two words are essential to answer the given question. In Figure 6.4, the word *"person"* is associated with the bounding boxes labeled as *"woman"*,*"shirt"*,*"shoe"*,*"leg"*, while the word *"shorts"* is associated with the *"shorts"* bounding box. Based on this observation, all experts concluded that the model correctly sees the required objects, and more broadly over the evaluation instances, that the first LV layer might be responsible for the recognition of objects with respect to the question. One of the experts mentioned that therefore, *"if we don't see a good word/bounding-box association here, the model can hardly cope with such a mistake and might exploit dataset biases"*. In order to verify such a statement, we pruned the four heads in this LV layer, to observe how the model would behave with no association in them. From such pruning, we observe that the following vision-to-language (VL) layers have lower attention distributions, close uniform in some cases. In addition, after pruning, the model's prediction wrongly switched from *"Yes"*, a rare answer (in Tail), to *"No"*, the most frequent one.

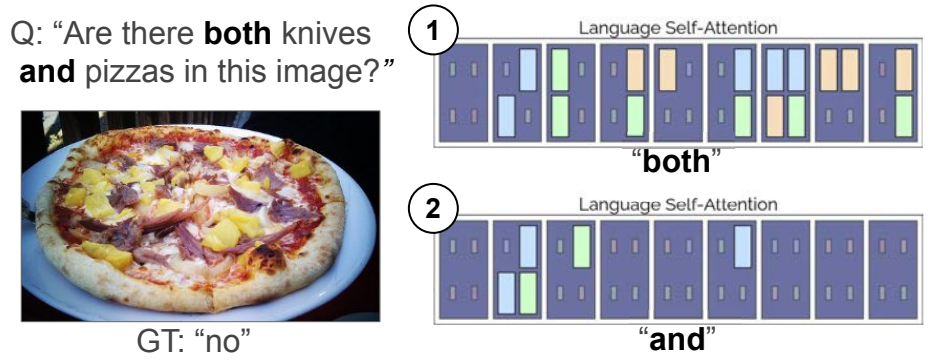


Figure 6.5 – When asked *“Are there both knives and pizzas in this image?”*, the *oracle transfer* model fails and answers *“yes”*. By filtering heads associated with a selected word, we can observe that language self-attention heads are more responsive to the word *“both”* ①, as opposed to the word *“and”* ②.

6.4.3 *Questions with Logical Operators*

During the evaluation, experts were shown two instances with questions containing the word *“and”*. Such instances are interesting because, as one of the experts mentioned, *“this word has a lot of importance is this question”*. To answer correctly, the model needs to grasp that it must analyze the image over two different aspects. With the image, illustrated in Figure 6.5, and asked *“Are there both knives and pizzas in this image?”*, the model fails and answer *“yes”*, the most frequent answer despite having no knife in the picture nor provided bounding-boxes. However, when asked *“Are there knives in this image?”* the model correctly answers *“no”*. This suggests that the model failed to grasp the meaning of the keyword *“and”*, and thus that the self-attention language heads might associate wrong words. Also, swapping the terms *“knives”* and *“pizzas”* in the question, yields the correct answer, *i.e.* *“no”*. This may indicate that the model ignores the first term when questions contain the operator *“and”*. Using the head-filtering interaction, we can observe that in attention heads, the word *“and”* has little to no attention. Instead, the word *“both”* has peaky attention scattered across most of self-language layers, and some language-to-language heads. Pruning those 19 heads makes the model correctly yield *“no”*, regardless of the order the words *“knives”* and *“pizzas”* are in the question. Such a behavior can be observed over our evaluation dataset, in which 34 questions have the keyword *“and”*. On those questions the model, without pruning, can provide a correct answer 62% of cases, up to 64% with the two words around *“and”* are swapped. In opposition, while having the 19 attention-heads with peaky attention for the word *“both”* pruned, the model reached an accuracy of 76%, down to 74% with words around *“and”* swapped. In the worst case, this pruning of the 19 attention heads illustrated in Figure 6.5 is responsible for an improvement of 10% on question contain the operator *“and”*.

6.4.4 *Vision to Vision Contextualization*

When asked *“What is the woman holding?”*, with the image in Figure 6.6, the model fails and outputs *“remote-control”*, a frequent answer, instead of *“hair dryer”*. This could be interpreted as bias exploitation. However, in such a dataset, *“remote-control”* is not among

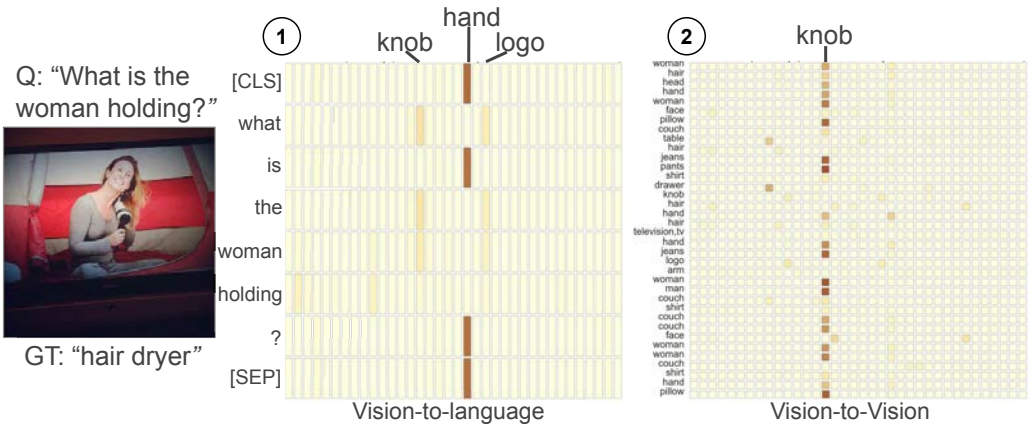


Figure 6.6 – Without any *"hair dryer"* provided by the object detector, the *oracle transfer* associates in its vision-to-language ① the object *"hand"* with the words *{"[CLS]","is","?","[SEP]"}*. While vision-to-vision focuses on a *"knob"* object ②.

the 10 most common answers to this question. This raises the question of what leads the model to output such an answer. During evaluation on this instance, experts noticed that the object detector failed to provide a *"hair dryer"* object. Similar to the use case given in Section 6.3, such a mistake forces the model to draw its attention towards other bounding boxes related to the missing object. In this case, as observed by experts, a majority of the vision-to-language reached their highest association between the word *"holding"* and bounding boxes labeled as *"hands"*. Such an association is expected as held objects are directly related to hands, and no *"hair dryer"* bounding box is provided. Among those bounding boxes, we can observe the presence of one labeled as *"television"*, and another as *"knob"* which are associated to *"holding"* and *"woman"* in both vision-to-vision_2_2 and early vision-to-language layers. This suggests that those heads might have influenced the model's predictions towards *"remote-control"* instead of the most common dataset bias. This can be confirmed by pruning those heads which yields a more frequent answer: *"cell phone"*. One of the experts also highlighted that those attention heads had a high association with the tokens *"[CLS]"*, *"is"*, *"?"*, and *"[SEP]"*. Which the expert interpreted as *"the model correctly transferred the context of the question"*.

6.5 CONCLUSION

We introduced how VisQA – an interactive visual analytics tool designed to perform instance-based in-depth analyses of the attention – helps to better understand reasoning behavior in transformer neural networks for vision and language reasoning. Thanks to its multiple features – displaying attention head intensities; inspecting attention distributions; pruning attention heads; asking free-form questions – VisQA allows us to conduct a qualitative study of bias vs reasoning in a VL-Transformer model. Our quantitative evaluations are encouraging, providing the first evidence that we can obtain indications on the reasoning behavior of a neural network using its attention maps, *i.e.* estimates on whether it correctly predicts an answer, and whether it exploits biases. Finally, VisQA provides us interesting cues about VQA VL-Transformer models' behavior, that will be explored in a broader statistical analysis in Chapter 7.

Chapter 7

ON THE EMERGENCE OF REASONING PATTERNS IN VQA

7.1 INTRODUCTION

In this chapter, we continue to study the capabilities of VQA models to "*reason*". As a recall, while an exact definition of this term is difficult, we refer to (Bottou 2014) and define it as "*algebraically manipulating words and visual objects to answer a new question*" (*cf.* Chapter 2). In particular, we interpret reasoning as the opposite of exploiting spurious biases in training data. We argue, and will provide evidence, that learning to algebraically manipulate words and objects is difficult when visual input is noisy and uncertain compared to learning from perfect information about a scene. When objects are frequently missing, detected multiple times or recognized with ambiguous visual embeddings wrongly overlapping with different categories, relying on statistical shortcuts may be a simple and tempting alternative for the optimizer.

In Chapter 6, we introduced VISQA, a tool designed to help researchers analyzing the reasoning and biases at work in Transformer based VQA models. This interactive tool provides a fine-grained understanding of the reasoning and bias mechanisms learned by the model. However, this study is limited to be at instance level. This is at the same time a benefit and a drawback. A benefit, as it lets the user inspect attention layers without alteration due to statistical aggregations. Thus, it provides cues and intuitions on what has been learned by the VQA model. At the same time, instance level visualization is not sufficient to discover how the model behaves at a dataset level. Therefore, in this chapter, we conduct a complementary analysis, focusing on a large scale statistical analysis of the attention mechanisms learned by the same VL-Transformer VQA model.

More precisely, drawing conclusion from Chapter 6, we propose an in-depth analysis of attention mechanisms in Transformer-based models and provide indications of the patterns of reasoning employed by models of different strengths. We visualize different operating modes of attention and link them to different sub tasks ("*functions*") required for solving VQA. In particular, we use this analysis for a comparison between perfect-sighted (oracle) models and standard models processing noisy and uncertain visual input, highlighting the presence of reasoning patterns in the former and less so in the latter. Indeed, we show that a perfect-sighted oracle model learns to predict answers while significantly less relying on biases in training data. Therefore, we claim that once the noise has been removed from visual input, replacing object detection output by Ground

Truth (GT) object annotations, a deep neural network can more easily learn the *reasoning patterns* required for prediction and for generalization.

In addition to improving our understanding of VL-Transformer decisions, this large scale analysis will serve as a basis for enhancing VQA training methods. In particular, we will explore a method for transferring the reasoning patterns learned by the oracle (trained with GT visual input) to the standard settings where visual inputs are extracted using an (imperfect) object detector (*cf.* Chapter 9).

CONTRIBUTIONS OF THE CHAPTER

(i) A study of the visual bottleneck in VQA, *i.e.* we explore how the visual uncertainty (caused by imperfect object detectors) affects VQA performance.

(ii) An in-depth analysis of reasoning patterns at work in Transformer-based models, including: (a) visualizations of attention modes; (b) an analysis of the relationships between attention modes and reasoning; and (c) an exploration of the impact of attention pruning on reasoning.

(iii) A comparison of oracle vs. noisy (standard) models, where we show that the former more easily learns *reasoning patterns*.

7.2 VISION IS THE BOTTLENECK

VisQA study made us wonder: *is computer vision the bottleneck?* We conjecture that difficulties in the computer vision pipeline are the main cause preventing VQA models from learning to reason well, and which leads them to exploit spurious biases in training data. Most of these methods use pre-trained off-the-shelf object detectors during training and evaluation steps. But in a significant number of cases, the visual objects necessary for reasoning are misclassified, or even not detected at all, as indicated by detection rates of SOTA detectors on the Visual Genome dataset (Krishna et al. 2017), for instance. In that context, Under these circumstances, even a perfect VQA model is unable to predict correct answers without relying on statistical shortcuts. In the context of a collaboration with Pierre Marza, we propose two experiences shedding light on the visual bottleneck and its potential consequences.

OBJECT DETECTION QUALITY We evaluate the quality of the objects detected by Faster RCNN (Ren et al. 2015) for the VQA task. In particular, we ask: *are important objects (given the question) correctly detected by the detector?* Therefore, for each question of the GQA-OOD validation split, we measure the proportion of object correctly detected and required for answering the question. Results are shown in Table 7.1. In our setup, an object is correctly detected if it sufficiently overlaps (measured with IoU, Intersection over Union) with the ground truth. It is worth to notice that it only provide an underestimation of the detector capabilities, as we do not consider the predicted label associated to the image region (*e.g.* in some case, a region can be falsely detected). We observe that the detection is not accurate enough for VQA, as many important objects are not detected (especially when the IoU threshold is set to 0.8).

GQA-OOD val. split	R@0.2	R@0.5	R@0.8
Head	89.7%	77.1%	12.7%
Tail	89.0%	75.8%	12.6%

Table 7.1 – Are important objects correctly detected? We report R-CNN recall (R) on objects required for answering the question with various IoU tresholds (R@0.8 means that a ground-truth object is considered as correctly detected if it has an IoU greater than 0.8 with one of the Faster R-CNN objects). We observe that, on both head and tail GQA-OOD splits, the object detection is not accurate and only few important objects are perfectly detected.

Table 7.2 – Impact of object detection quality (embeddings and BBs) on the UpDn VQA model (Anderson et al. 2018), evaluated as comparison with oracles (GQA balanced validation set).

GT BB boxes	GT embeddings (1-in-K class)	Perturbed B. boxes	Perturbed Embeddings	Accuracy	Binary	Open
–	–	–	–	60.01	72.22	48.56
✓	–	✓	✓	59.58	76.75	43.48
✓	–	✓	–	69.21	82.15	57.08
✓	–	–	–	69.21	82.18	57.06
✓	✓	–	–	83.29	82.93	83.62

IMPACT OF THE VISUAL QUALITY Table 7.2 indicates that more than 20 accuracy points are gained when both object features and bounding boxes are taken from a perfect (oracle) object detector. This confirms our intuition that there is a large room for improvement on the object detection side of VQA. Moreover, analyzing the gain brought by perfect selection of bounding boxes alone, one may notice that it can bring more than +9 pts of improvement for VQA. We also measure to what extent the exact regression of the object coordinates (bounding boxes) is essential for VQA, evaluating the scores under the perturbation of the GT coordinates. For each GT bounding box coordinate, we sample random translations from a uniform distribution over $[-\frac{l}{2}; +\frac{l}{2}]$, where l is the size of the bounding box along the axis at hand. The results are shown in the 3rd row of Table 7.2 and paint a clear picture: given the rather strong amplitude of the coordinate perturbations, the drop in performance is surprisingly small. On the contrary, if in addition to the bounding box coordinates perturbations, we also perturb the detector's feature embeddings, the VQA performance drastically drops (the 2nd row in Table 7.2). This corroborates the intuition that answering questions in current applications and datasets requires a rather coarse knowledge of where objects are mostly restricted to their spatial relationships with other objects (*left, right, above, under, below*, etc.), but a quite precise knowledge of the type of objects involved is necessary. In other words, it is important to coarsely select the objects required for answering the question, but the precise regression of their bounding box coordinates is not important. This result (hopefully) tones down a bit the observation made in Table 7.1, even thought the visual uncertainty remains a main bottleneck for VQA.

7.3 VISUAL NOISE VS. MODELS WITH PERFECT-SIGHT

7.3.1 *Oracle: a perfect-sighted model*

To further explore this working hypothesis, we propose to compare the learned attention of a VL-Transformer (*cf.* Chapter 3) trained with two different settings: *oracle* and *noisy*. *Oracle* setting consists in training a VL-Transformer model with perfect sight, *i.e.* a model which receives perfect visual input. It receives the GT objects from the GQA annotations, encoded as GT bounding boxes and 1-in-K encoded object classes replacing the visual embeddings of the classical model. All GT objects are fed to the model, not only objects required for reasoning. *Noisy* settings corresponds to the classical model, based on the same VL-Transformer as *oracle*, but taking as input objects features detected by an object detector (FasterRCNN (Ren et al. 2015)). We call it *noisy* because of the uncertainty in the vision part.

EXPERIMENTAL SETUP All analyses in this chapter have been performed with a hidden embedding size $d = 128$ and a number of per-layer heads $h = 4$. This corresponds to the compact version of the VL-Transformer architecture (*cf.* Chapter 3). Therefore, "*compact-LXMERT*" corresponds to the VL-Transformer architecture plus BERT-like (LXMERT) pre-training. Unless specified otherwise, objects have been detected with Faster R-CNN (Ren et al. 2015). Visualizations are done on GQA (Hudson et al. 2019b) (validation set) as it is particularly well suited for evaluating a wide variety of reasoning skills. However, as GQA contains synthetic questions constructed from pre-defined templates, the dataset only offers a constrained VQA environment. Additional experiments might be required to extend our conclusions to more natural setups.

7.3.2 *Does the oracle "*reason*"?*

We study the capabilities of both models – the oracle model and the classical one – to "*reason*" (see our definition in Chapter 2). Following Chapter 5 we measure the reasoning capabilities of a VQA model as the capacity to correctly answer questions, where the GT answer is rare *w.r.t.* the question group, *i.e.* the type of questions being asked. In particular, we evaluate the models on the GQA-OOD benchmark designed for OOD evaluation (*cf.* Chapter 5).

OOD EVALUATION Figure 7.1 illustrates the model behavior in different situations. At the extreme case (left side of the plot), the model is evaluated on the rarest samples only, while on the right side all samples are considered. We observe that the performance of the classical model taking noisy visual (compact-LXMERT) drops sharply for (image, question) pairs with rare GT answers, which is an indication of a strong dependency on dataset biases. We would like to insist that in this benchmark the rarity of a GT answer is determined *w.r.t.* the question type, which allows measuring biases taking into account language. The oracle model, on the other hand, obtains performances which are far less dependent on the answer rarity, providing evidence for its ability to overcome statistical biases. As a consequence, we conjecture that the visual oracle is closer to a real "*reasoning*

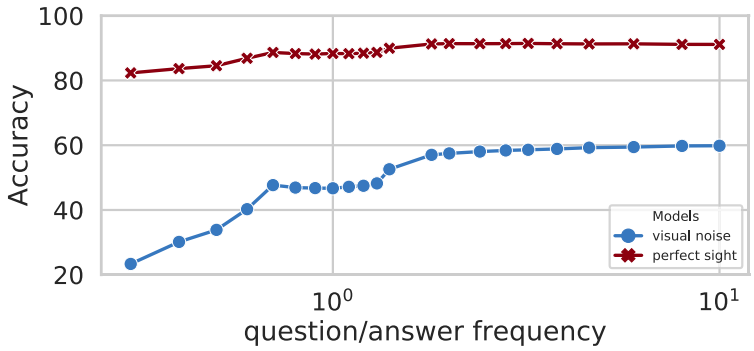


Figure 7.1 – Uncertainties and noise in visual input dominate the difficulties in learning reasoning: comparison of the out-of-distribution generalization between two different VQA Models. A perfect-sighted oracle model and a standard noisy vision based model trained on the GQA-OOD benchmark (Kervadec et al. 2021a). For the classical model, accuracy drops for questions where the GT answer is rare (left side) compared to frequent answers (right side), indicating probable bias exploitation. In contrast, the oracle obtains high performance also on rare answers. Both models are compact-LXMERT.

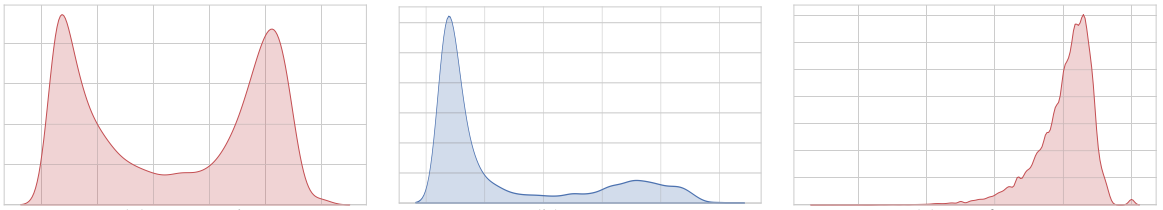
process", by predicting the answer resulting from a manipulation of words and objects, rather than by having captured statistical shortcuts. In the absence of GT on reasoning, we admit that there is no formal proof to this statement, but we believe that the evidence above is sufficient.

7.4 ATTENTION MODES IN VL-TRANSFORMERS

7.4.1 *Defining and estimating the attention modes*

Attention modes, or distributions, are at the heart of the VL-Transformer. They are not directly supervised during training, their behavior emerges from training the different VQA objectives, *i.e.* the discriminative loss as well as the eventual additional BERT-like objectives (Tan et al. 2019). Their definition as a strength of association between different items makes them a prime candidate for visualization of inner workings of deep models. We analyze attention, and in particular we observe different attention modes in trained VQA models.

k-DISTRIBUTION We use the technique previously introduced in Chapter 6. As a recall, it consists in visualizing the distribution of attention energy associated with each Transformer head in multi-headed attention, following (Ramsauer et al. 2020). For each attention map, associated with a given head for a given sample, we calculate the number k of tokens required to reach a total sum of 90% of the distribution energy. A low k-number is caused by peaky attention, called *small meta-stable state* in (Ramsauer et al. 2020), while a high k-number indicates uniform attention, close to an average operation (*very large meta-stable state*). For each head, and over a subset of validation samples, we plot the



(a) Bimorph (b) Dirac (c) Uniform

Figure 7.2 – Attention modes learned by the oracle model. Following (Ramsauer et al. 2020), for each head we plot the distribution of the number k of tokens required to reach 90% of the attention energy (GQA-val). X-axis (from 0 to 100%): ratio of the tokens k *w.r.t.* the total number of tokens. Plots are not attention distributions, but distributions of indicators of attention distributions. We observe three major modes: (a) "*bimorph*" attention, unveil two different types of attention distribution for the same head; (b) Dirac attention with high k-median, *i.e.* small meta stable state; (c) uniform attention, with low k-median, *i.e.* very large meta stable state.

distribution of k-numbers, and for some experiments we summarize it with a median value taken over samples and over tokens.

7.4.2 *Diversity in attention modes*

In this experiment, we focus on the oracle VL-Transformer, where we observed a high diversity in attention modes. We also observed that some layers' heads, especially those processing the visual modality (t_{-}^{V} or $t_{\times}^{V \leftarrow L}$) [1] are mainly working with close-to-average attention distributions (*very large meta-stable states* (Ramsauer et al. 2020)). On the other hand, we observed smaller meta-stable states in the language layers (t_{-}^{L} or $t_{\times}^{L \leftarrow V}$). This indicates that the reasoning process in the oracle VL-Transformer is in large part executed by the model as a transformation of the language features, which are successively contextualized (i.e. influenced) by the visual features (and not the opposite).

BIMORPH ATTENTION MODE In contrast to the attention modes reported in (Ramsauer et al. 2020), we also observed bi-modal k-number distributions, shown in Figure 7.2-a, which are a combination of a Dirac (Figure 7.2-b) and uniform (*cf.* Figure 7.2-c) attention modes. We call these modes "*bimorph*" attention, since they reveal the existence of two different shapes of attention distribution: for some samples, a Dirac activation is generated, while other samples lead to uniform attention (averaging over tokens) [2].

ORACLE'S HEADS ARE MORE DIVERSE Besides, in Figure 7.3, we compare attention mode diversity between the noisy visual model and the oracle $t_{\times}^{L \leftarrow V}$ heads, where we observe higher diversity for the oracle. In particular, "*bimorph*" attention is mostly performed by the oracle.

1. As a recall, the annotation is introduced in Section 3.5.
2. We remind that these plots are distributions of indicators of distributions: uniform behavior does not show up as a flat plot, but as plot with a peak on the right side — it may in these plots look like a Dirac.

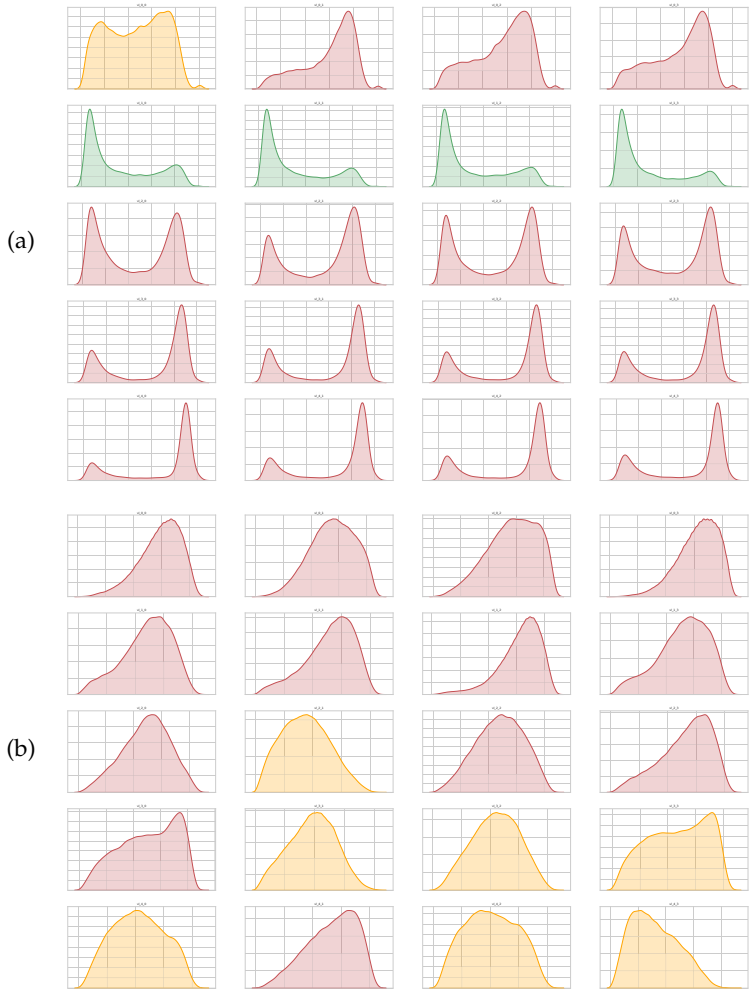


Figure 7.3 – Comparison of k-distribution of $t_{\times}^{L\leftarrow V}$ attention heads for two different models: (a) oracle; (b) noisy visual input. Rows indicate different $T_{\times}^{L\leftarrow V}$ layers. Heads are colored according to the median of the k-number.

7.5 ATTENTION MODES AND TASK FUNCTIONS

In this experiment, we study the relationships between attention modes and question types, which correspond to different functions of reasoning required to solve the problem instance. In other words, we explore to what extent the neural model adapts its attention distribution to the question at hand. We group the set of questions according to functions using the GQA (Hudson et al. 2019b) annotation, using 54 different functions such as *e.g.* "*filter color*", "*verify size*", etc..[3]

3. There is limited overlap between functions, *e.g.* "*filter*" contains, among others, the "*filter color*" and "*filter size*".

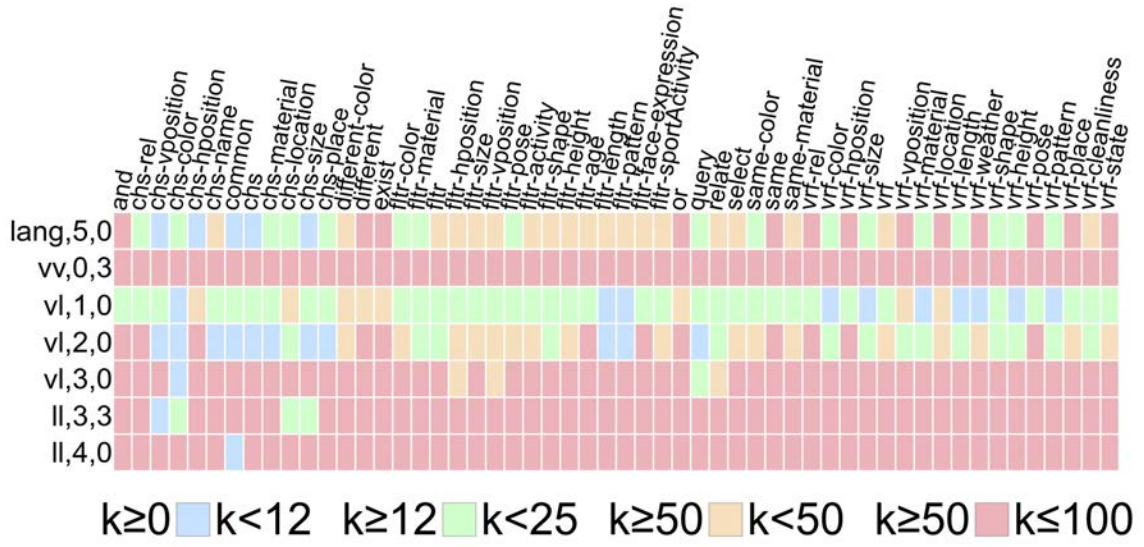


Figure 7.4 – Attention modes for selected attention heads (rows) related to functions required to be solved to answer a question (columns). The head's notation x, i, j refers to the head j of the i-th Transformer layer of type x: 'lang'/'ll'=$t^L_-(\cdot)$, 'vis'/'vv'=$t^V_-(\cdot)$, 'vl'=$t^{L\leftarrow V}_\times(\cdot)$, 'lv'=$t^{V\leftarrow L}_\times(\cdot)$. The VL-Transformer's architecture is presented in Chapter 3. The color encodes the attention mode, *i.e.* median of the k-number (Ramsauer et al. 2020). We observe (1) attention heads behave differently depending on the function; (2) a given function causes different attention modes for different heads.

7.5.1 *Attention vs. function in oracle setting*

We link functions to the attention modes introduced in Section 7.4. In Figure 7.4 we show functions in columns and a selection of attention heads in rows, while the color encodes the median k-number for the oracle model.

RELATION BETWEEN ATTENTION AND TASK FUNCTION We observe a certain dependency between functions and the attention modes. Certain functions, *e.g.* the majority of the "*choose X*" functions, tend to cause the emergence of small meta-stable states. In these modes, the attention mechanism is fundamental, as it allows the model to attend to specific token combinations by detecting specific patterns. On the other hand, some functions requiring to attend to very general image properties, such as "*choose location*" or "*verify weather*", seem to be connected to very large meta-stable states. We conjecture, that to find general scene properties, a large context is needed. In these modes, the attention mechanism is less important, and replacing it with a simple averaging operation is likely to keep performance — an experiment we explore in Section 7.6. Similarly, when focusing on heads instead of functions, we observe that a majority of heads typed as $t^{V\leftarrow L}_\times(\cdot)$ or $t^V_-(\cdot)$ tends to behave independently of the question functions, and they generally show close-to-uniform attention.

EMERGENCE OF SPECIALIZED HEADS On the other hand, the $t^L_-(\cdot)$ and $t^{L\leftarrow V}_\times(\cdot)$ heads are highly dependent on the question functions. As shown in Figure 7.4 and Figure 7.5,

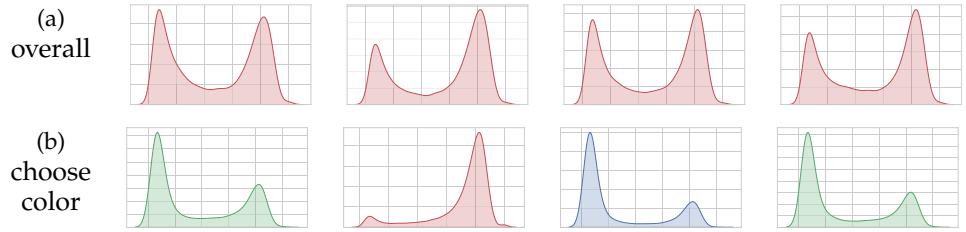


Figure 7.5 – Influence of the question on oracle's "*bimorph*" attention heads. We compare attention modes of the third layer of $T_{\times}^{L \leftarrow V}$ heads as a distribution of the k-numbers (Ramsauer et al. 2020) over (a) samples of all functions, and (b) samples with questions involving the "*choose color*" function, and observe a clear difference. The function "*choose color*" seems to cause the activation (*i.e.* emergence of a small meta-stable state) of the 1^{st}, 2^{nd} and 4^{th} head, and the desactivation of the 3^{rd} one, further indicating task dependence of attention head behavior.

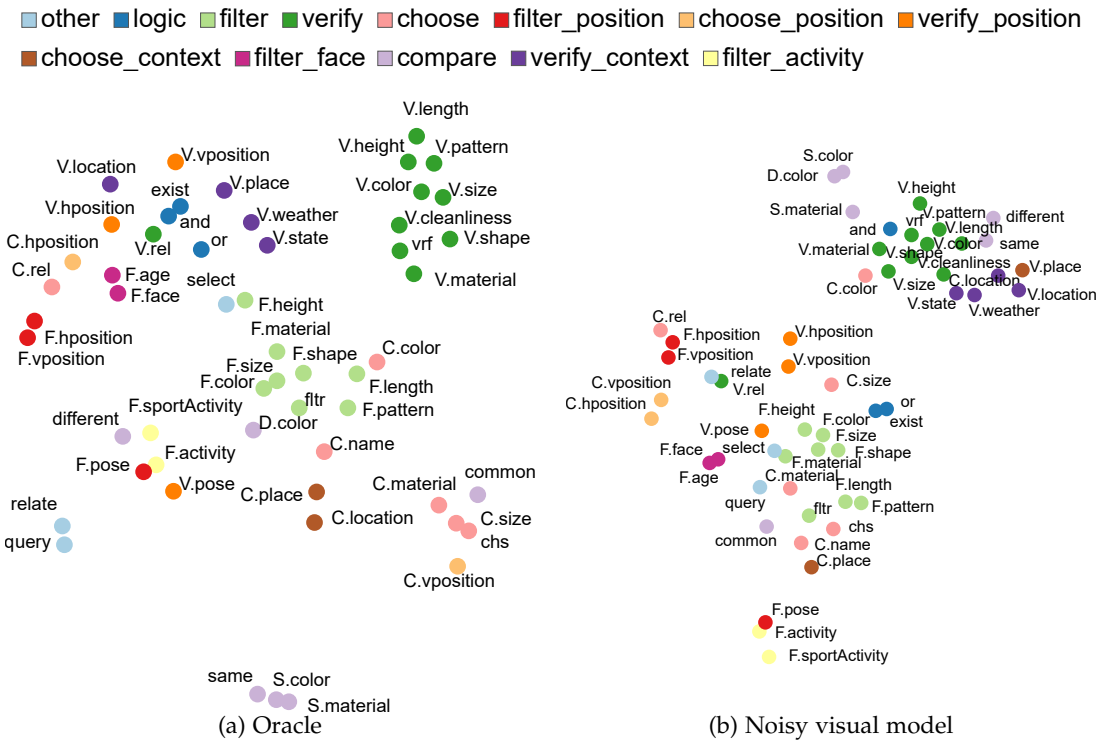
these heads does not behave in the same way and are not "*activated*" (*i.e.* have a smaller metastable-state) for the same combination of functions. This provides some evidence for modularity of the oracle VL-Transformer, each attention head learning to specialize to one or more functions.

ILLUSTRATION WITH `choose color` In addition, in Figure 7.5, we visualize the difference in oracle attention modes between two different function configurations: Figure 7.5-a is the distribution of median k-numbers over *all* samples, *i.e.* involving all functions, whereas Figure 7.5-b shows the distribution over samples involving the "*choose color*" function. We show the 3rd $T_{\times}^{V \leftarrow L}$ Transformer layer heads. Over all functions, these heads show "*bimorph*" behavior, whereas on questions requiring to choose a color, these same heads show either dirac or uniform behavior.

7.5.2 *Oracle vs. Noisy Input*

In the next experiment, we explore the difference in behavior between the perfect-sighted oracle and the classical model taking noisy visual input. For each input sample, we create a 80-dimensional representation describing the attention behavior of the model by collecting the k-numbers of the 80 cross-attention heads into a flat vector, taking the median over the tokens for a given head.

STANDARD MODEL FAILS TO RELATE ATTENTION TO FUNCTION Figure 7.6 shows two different t-SNE projections of these attention behavior space, one for the oracle model and one for the noisy model. While the former produces clusters regrouping functions according to their general type, the function representation of the noisy model is significantly more entangled. We conjecture, that the attention-function relationship provides insights into the reasoning strategies of the model. VQA requires handling a large variety of reasoning skills and different operations on the input objects and words. Question-specific manipulation of words and objects is essential for correct reasoning. In contrast to the oracle one, the t-SNE plot for the noisy visual model paints a muddier picture, and does not show clear relationships between attention modes and functions.



(a) Oracle (b) Noisy visual model

Figure 7.6 – t-SNE projection of the attention mode space, *i.e.* the 80-dim representation median k-numbers, one per head of the model. Colors are functions, also provided as overlaid text. We compare projections of (a) the oracle, and (b) the noisy visual model, and observe a clustering of functions in the attention mode space for the oracle, but significantly less for the noisy input model.

Furthermore, when analyzing the special case of the `choose color` task function (in Figure 7.8), we do not observe any evidence of a relation between attention and function for the noisy setting.

CAVEAT visualizing attention modes does not provide any indication of the attention operation itself, only about the shape of the operation. In particular, an attention head might result in the same low k-number for two different input samples, showing Dirac attention, but could attend to quite different objects or words in both cases.

7.6 ATTENTION PRUNING

We further analyze the role of attention heads by evaluating the effect of pruning heads on model performance. As reported by Voita et al. (2019) and Ramsauer et al. (2020), specific attention heads may be useful during training, but less useful after training. In the same lines, for specific heads we replace the query-key attention map by a uniform one, "*pruned*" heads will therefore simply contextualize each token by an averaged representation of all other tokens, as a head with large meta-stable state would have done.

Pruned attentions	n/a	L	V	L←V	V←L
Accuracy	91.5	37.9	91.4	52.8	68.1

Table 7.3 – Impact of pruning different types of attention heads of the trained oracle model. We observe that 'vision' and 'language→vision' Transformers are hardly impacted by pruning, in contrast to 'language' and 'vision→language'. Accuracies (in %) on the GQA validation set.

7.6.1 *Pruning different types of attention heads*

In Table 7.3 we report the effect of pruning on GQA validation accuracy according to different attention categories and observe that the oracle model is resilient to pruning of the $t_{-}^{V}(\cdot)$ and $t_{\times}^{V\leftarrow L}(\cdot)$ heads, but that pruning of $t_{-}^{L}(\cdot)$ and $t_{\times}^{L\leftarrow V}(\cdot)$ heads results in sharp drops in performance. This indicates that the bulk of reasoning occurs over the language tokens and embeddings, which are contextualized from the visual information through $t_{\times}^{L\leftarrow V}(\cdot)$ cross-attention. We can only conjecture why this solution emerges after training — we think that among reasons are the deep structure of language and the fact that in current models the answer is predicted from the CLS language token.

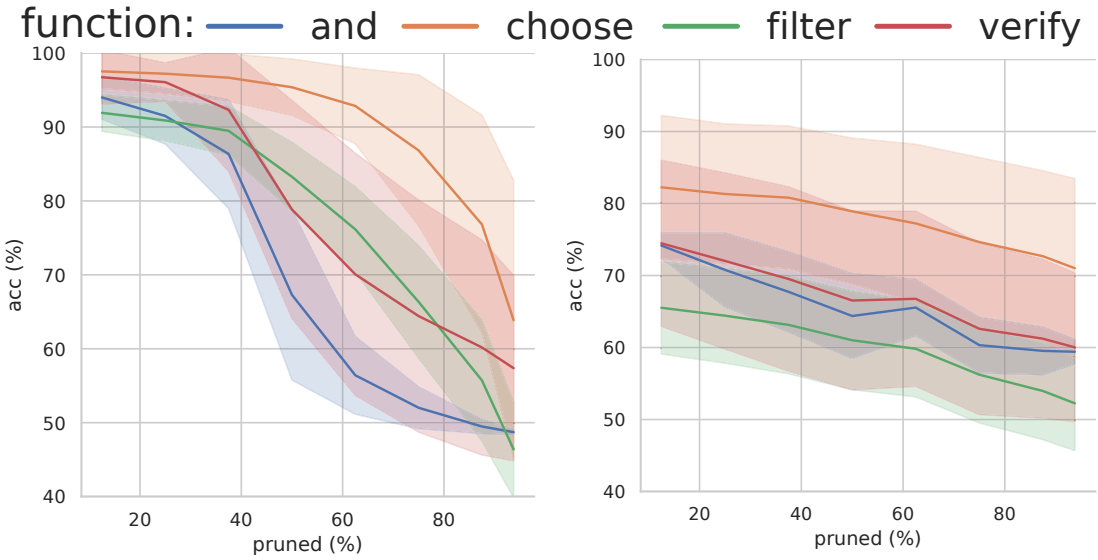
7.6.2 *Impact on functions*

We study the impact of pruning on the different task functions by randomly pruning n cross-attention heads and measuring accuracy for different function groups, n being varied between 0% (no pruning) to 100% (all heads are pruned), as shown in Figure 7.7 for the oracle and noisy vision-based models. For the sake of clarity only 4 different functions are shown, additional results are provided in Figure 7.9. For the perfect-sighted oracle (Figure 7.7-a), we first observe that the pruning has a different impact depending on the function. Thereby, while *filter* and *choose* are dominated by negative curvature where performance drops only when a large number of heads are pruned, *verify* and *and*, are characterized by a sharp inflection point and an early steep drop in performance. This indicates that the model has learned to handle functions specifically, resulting in various degrees of reasoning distribution over attention heads. For the noisy vision-based model, on the other hand, the effect of head pruning seems to be unrelated to the function type (Figure 7.7-b).

7.7 CONCLUSION

In this chapter, we have provided a deep analysis and visualizations of several aspects of deep VQA models linked to reasoning on the GQA dataset. We have shown, that oracle models produce significantly better results on questions with rare GT answers than models on noisy data, that their attention modes are more diverse and that they are significantly more dependent on questions. We have experimentally measured a pronounced difference in attention modes between the perfect-sighted oracle and a noisy vision based model. More importantly, the oracle model shows a strong relationship between attention mode

and task function, which we interpret as the capability of adapting reasoning to the task at hand. The classical model significantly lacks these abilities, suggesting a strategy of transferring patterns of reasoning from an oracle model pre-trained on visual GT to a model taking noisy visual input. This *oracle transfer* method will be studied in Chapter 9.



(a) Oracle (b) Noisy visual

Figure 7.7 – Impact of random pruning of varying numbers of attention heads in cross-modal layers on GQA-validation accuracy. (a) For the oracle, the impact is related to the nature of the function, highlighting its modular property. (b) For the noisy-vision-based model, pruning seems to be unrelated to function types.

*
* *

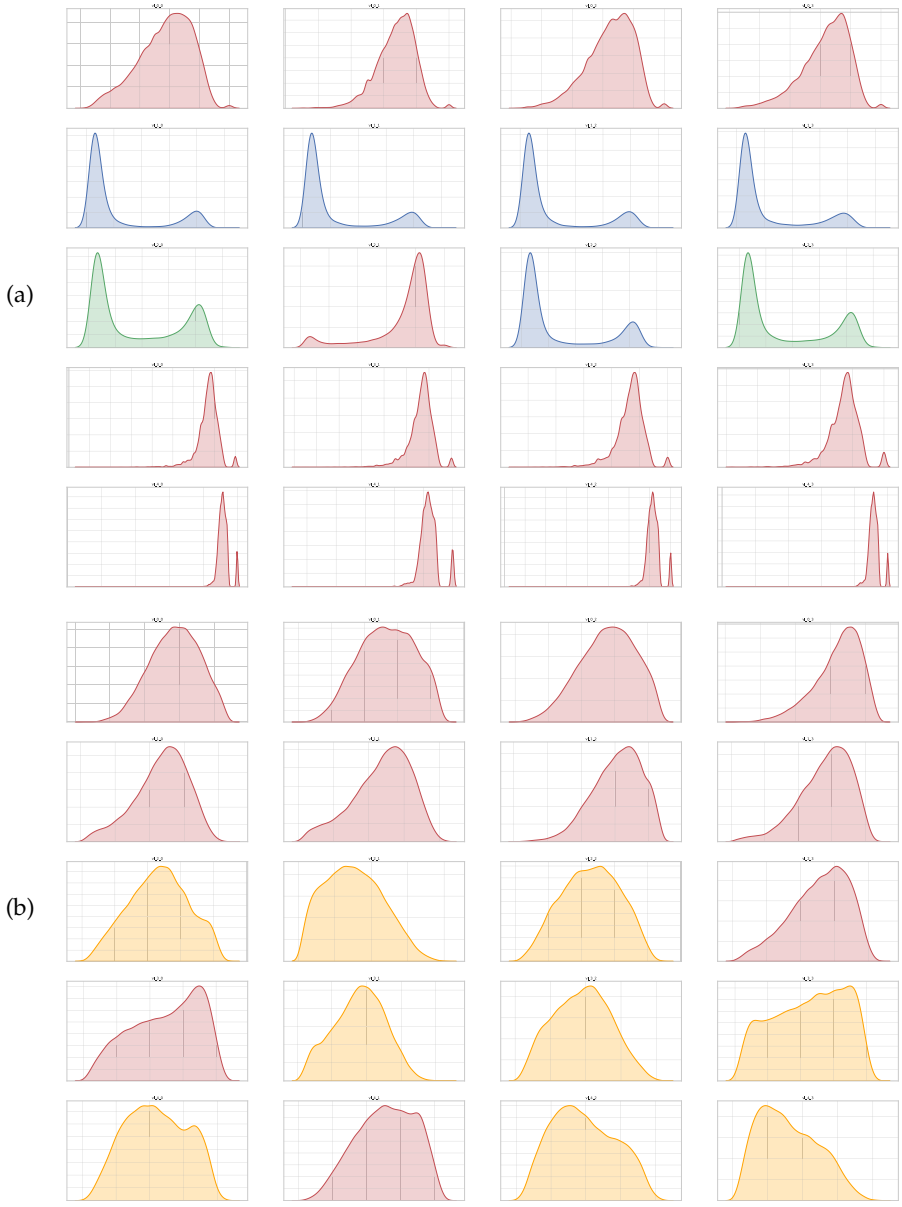
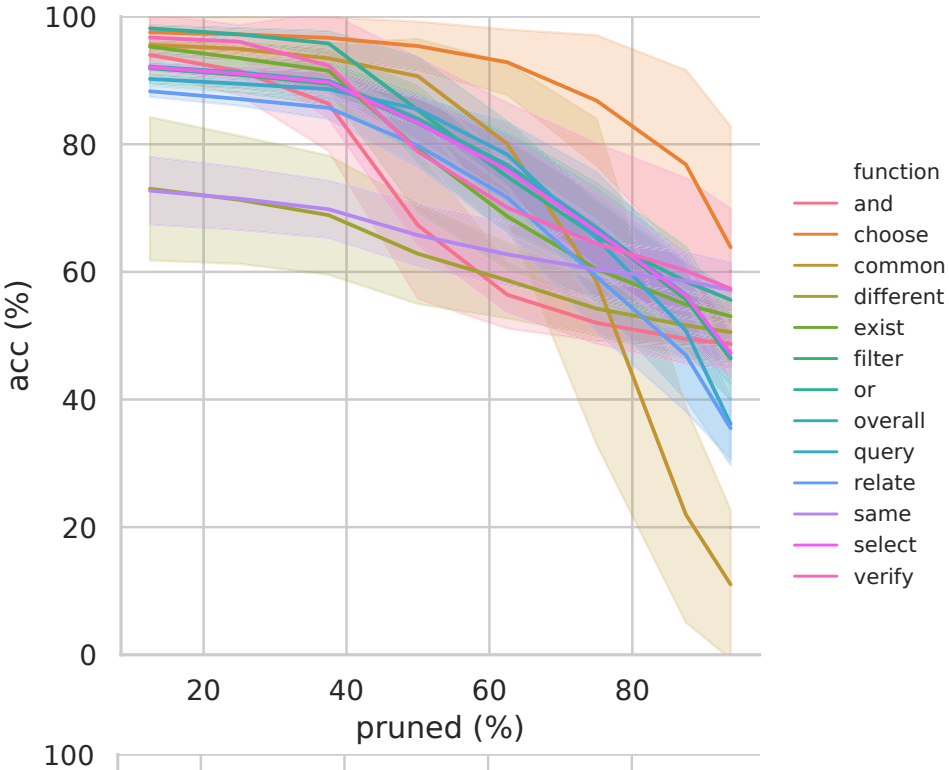


Figure 7.8 – Comparison of k-distribution of VL-attention heads for two different models for the function *choose color*: (a) oracle (5 first rows); (b) noisy visual input (5 last rows). Heads are colored according to their k-number median. As a recall, for each head we plot the distribution of the number k of tokens required to reach 90% of the attention energy (GQA-val). The x-axis represents in % the number of tokens k relatively to the total number of token, it goes from 0% to 100%.

(a) Oracle



(b) Noisy visual

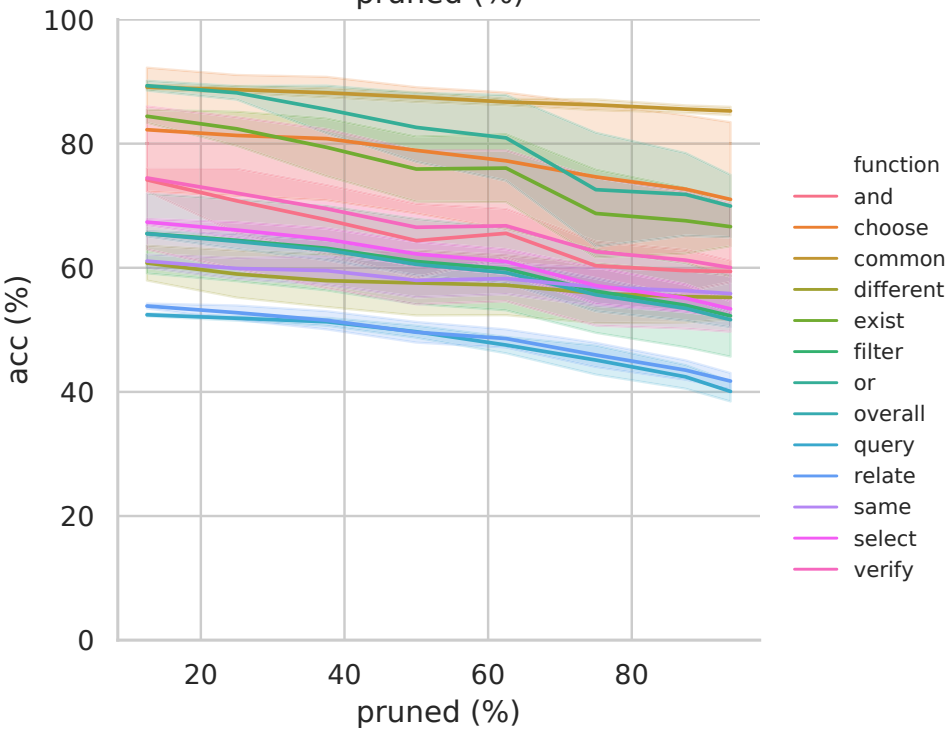


Figure 7.9 – Full visualization of the impact of random pruning of varying numbers of attention heads in cross-modal layers on GQA validation accuracy. (a) For the oracle, the impact is related to the nature of the function, highlighting its modular property. (b) For the noisy-vision-based model, pruning seems to be unrelated to function types.

Part IV

IMPROVE: A NEW HOPE

INTRODUCTION

In the year 2021 AD, 2 liters of coffee and 100 pages later:

– *Sir?*
– You, again?
– *Now I understand everything! We are call VQA a "*visual turing test*", but it turns out that, rather than reasoning, VQA models only learn shortcuts...*
– Don't be so pessimistic. In some cases, VQA models still succeed in reasoning. Slowly, but surely, we are moving towards models more apt to reason.
– *Do you mean that there is sill hope?*
– Exactly, and I have some ideas!

In Part II and Part III we experimentally demonstrate that SOTA VQA models tend to leverage dataset biases and shortcuts in learning rather than performing reasoning, leading to lack of generalization. Using VISQA (*cf.* Chapter 6) to inspect the attention maps learned by a VL-Transformer taught us that it struggles to detect the fined-grained interactions between language and vision. Furthermore, when it succeeds, it is often dominated by shortcuts. As an illustration, when asking "*What is the woman holding?*" with the picture in Figure 7.10 to the tiny-LXMERT (Tan et al. 2019) model, it wrongly predicts "*a banana*". Yet, the attention map *VL_3_0* informs us that the model has correctly grounded "*woman*" and "*holding*" to the woman's face and the glove, respectively. This suggests that the reasoning process, leading to the answer prediction, does not rely on the right cues (here, it relies on shortcuts rather than on word-object alignment). Thus, it appears as a necessity to develop methods preventing shortcut learning in VQA.

IMPROVING THE REASONING PROCESS Drawing conclusion from our evaluations (Part II) and analyses (Part III), we now propose to improve the performance of VQA models. In the VQA literature, the reasoning ability is frequently assumed to be implicitly learned during training from application-specific losses, mostly cross-entropy for classification, or the use of inductive biases in the model's architecture. We conjecture that it is not so obvious, and explore two alternatives, improving the VQA accuracy and reducing the impact of biases on the prediction. The first one aims at guiding the reasoning process during training, leveraging a weak supervision of the object-word alignment or a supervision of the operations steps required to answer the question. The second one consists in pre-training the VQA model on perfect (oracle) input, in order to learn the reasoning patterns observed in Chapter 7, and transfer them to the standard settings, where vision is uncertain. The underlying intuition is that it is easier to learn a shortcut-free reasoning when the training conditions are favorable enough (*i.e.* when the uncertainty in the input is reduced). Part IV is organized as follows:

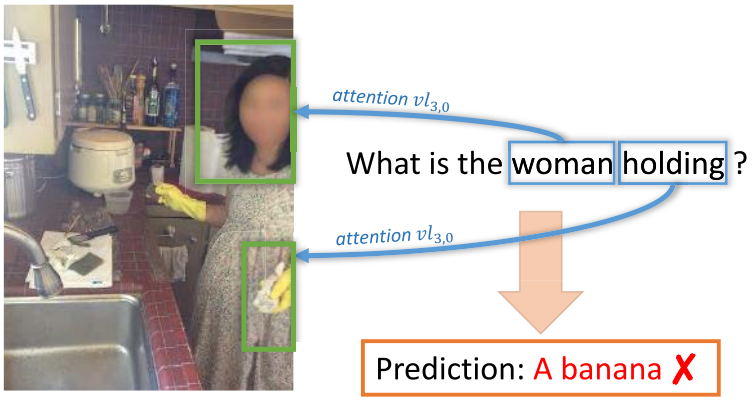


Figure 7.10 – To the question "*What is the woman holding?*", the tiny-LXMERT (Tan et al. 2019) model answers "*a banana*". Looking at the attention maps generated by attention head *VL_3_0*, we observe that it has correctly found the relationships between words "*woman holding*" and visual regions. However, these relationships are dominated by shortcuts, as the final prediction is "*a banana*" (a yellow fruit, like the gloves).

Chapter 8 addresses the question of the reasoning supervision. While the GT reasoning signal is not observable, it is still possible to approximate it through proxy losses. Thus, we propose a weak supervision of the word-object alignment during the training of VL-Transformer, in order to better ground its reasoning to the vision-language relationships. Furthermore, we borrow results from PAC-learning and provide theoretical cues on the benefits brought by this reasoning supervision.

Chapter 9 explores an alternative method, directly related to the analyses conducted in Chapter 7. We propose to transfer the reasoning patterns learned by a visual oracle, trained with perfect visual input, to a standard VQA model with imperfect visual representation. In a second part, we combine this method with the reasoning supervision, through program prediction, and show that the latter can be used as a catalyst for the transfer of reasoning patterns.

This Part has led to the publication of the following conference papers:

- Corentin Kervadec, Grigory Antipov, Moez Baccouche, and Christian Wolf (2019). "Weak Supervision helps Emergence of Word-Object Alignment and improves Vision-Language Tasks". In: *European Conference on Artificial Intelligence (ECAI)*;
- Corentin Kervadec, Theo Jaunet, Grigory Antipov, Moez Baccouche, Romain Vuillemot, and Christian Wolf (2021c). "How Transferable are Reasoning Patterns in VQA?". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*;
- Corentin Kervadec, Christian Wolf, Grigory Antipov, Moez Baccouche, and Madiha Nadri (2021d). "Supervising the Transfer of Reasoning Patterns in VQA". in: *Advances in Neural Information Processing Systems (NeurIPS)*.

Chapter 8

A PROXY LOSS FOR SUPERVISING REASONING

8.1 INTRODUCTION

High-capacity deep neural networks trained on a large amount of data currently dominate methods addressing problems involving either vision or language, or both of these modalities jointly (Tan et al. 2019). Examples for vision-language tasks are image retrieval task (Karpathy et al. 2015a) – retrieve an image given a query sentence –, image captioning (Lin et al. 2014) – describe the content of an input image in one or more sentences –, or VQA. These tasks require different forms of reasoning, among which we find the capacity to analyze instructions – *e.g.* the question in VQA –, or the ability to fuse modalities. Additionally, they often require different levels of understanding, from a global image-text comparison to fine-grained object-word matching. In this context, a wide panoply of high-performing models adopt self-attention architectures (Vaswani et al. 2017) and BERT-like (Devlin et al. 2019) training objectives, which complement the main task-related loss with other auxiliary losses correlated to the task (see VL-Transformer in Chapter 3). The common point of this large body of work is the large-scale training of unified vision-language encoders on image-sentence pairs.

However, despite their impressive success in standards benchmarks, we have shown in Chapter 5 that these models – in particular, LXMERT (Tan et al. 2019) – are prone to learn shortcuts instead of reasoning. More precisely, when analyzing the attention maps learned by a VL-Transfomer with VisQA (*cf.* Chapter 6), we observe that, despite its ability to model interactions unique to one modality (*i.e. intra*-relationships), it tends to struggle to identify fine-grained object-word relationships (*inter*-relationships, or cross-modality relationships). Yet, these relationships are essential in visual reasoning, which can be illustrated in the example of VQA (*cf.* Figure 8.1): answering a question given an input image requires the detection of certain objects in the image, which correspond to words in the question, and possibly the detection of more fine-grained relationships between visual objects, which are related to entities in the sentence.

In this chapter, we claim that the word-object alignment does not necessarily emerge automatically, but rather requires explicit supervision. Therefore, we design a training signal, aiming at supervising the model to learn a fine-grained matching between question's words and visual objects. This takes the form of an additional pre-training supervision, which can be viewed as a proxy loss for guiding the model to learn reasoning.

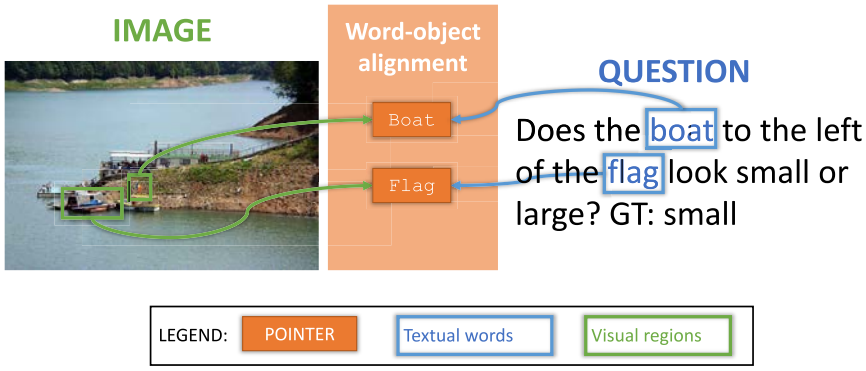


Figure 8.1 – Answering a question posed over an image requires grounding the question's words in the image. In this specific illustration, it is important to understand what are the image regions corresponding to the *boat* and the *flag*. We propose to supervise the VQA model to learn this alignment.

Our experiments show the benefit of this approach on VQA. Moreover, we also test its generalization to another task requiring to reason over images, namely the language driven comparison of images.

In a second part, we conjecture that such reasoning supervision in itself leads to a simpler learning problem. Indeed, the underlying reasoning function is decomposed into a set of tasks, each of which is easier to learn individually than the full joint decision function. Following recent works in PAC-learning (Xu et al. 2020), we back up this claim through a theoretical analysis showing decreased sample complexity under mild hypotheses.

CONTRIBUTIONS OF THE CHAPTER

(i) a weakly supervised word-object alignment objective for vision language reasoning;

(ii) a theoretical analysis of the benefit of supervising reasoning in VQA deriving bounds on sample complexity.

8.2 SUPERVISING WORD-OBJECT ALIGNMENT

In the literature, the alignment or matching of words to visual objects is generally assumed to be implicitly learned from application-specific losses — mostly cross-entropy for classification — thanks to the inductive biases provided by the encoder's architecture, *i.e.* the possibility of the model to *represent* this kind of matching. In this section, we experimentally show that (1) modality alignment does not necessarily emerge automatically and (2) that adding weak supervision for alignment between visual objects and words improves the quality of the learned models on tasks requiring visual reasoning. We therefore propose to add a *vision-language alignment decoder* on top of the VL-Transformer architecture (*cf.* Chapter 3), which directly supervises the word-object alignment.

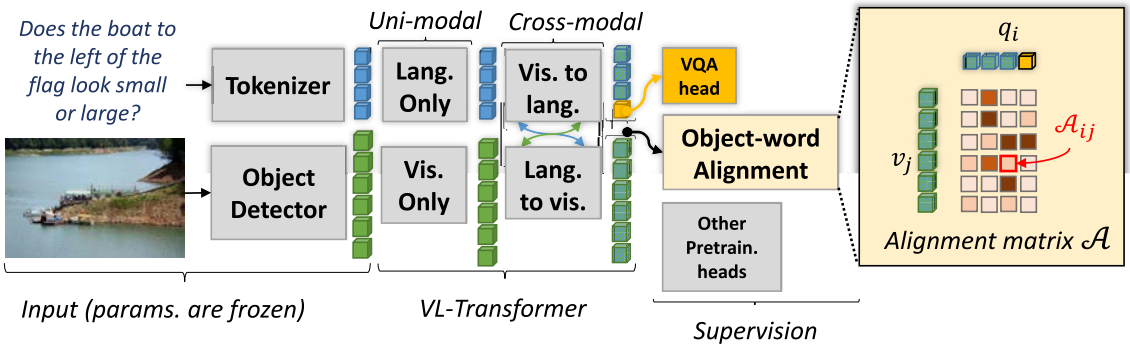


Figure 8.2 – We propose to add a word-object alignment module (*on the right*) on top of the VL-Transformer architecture, in order to supervise the fine-grained alignment between vision and language.

8.2.1 *Vision-Language Decoder*

The overall architecture of our model is presented in Figure 8.2. It is based on the VL-Transformer described in Chapter 3.

VL-TRANSFORMER As a recall, VL-Transformer is a vision-language encoder based on a succession of self- and guided-attention layers. The former are used to process uni-modal interactions (inside one modality) while the latter process the cross-modal interactions (between vision and language). On the input side, VL-Transformer is fed with the tokenized question and the image, which is represented as a set of objects extracted by an object detector (here, the Faster-RCNN (Ren et al. 2015)). Following Tan et al. (2019), the encoder is trained on a large image-sentences corpus with BERT-like losses adapted to the vision-language understanding: vision and language masking, image-sentence matching and VQA. We propose to augment this set with a word-object alignment loss, in order to improve the reasoning.

8.2.2 *Vision-Language Alignment Decoder*

As shown in Figure 8.2, we propose to add a vision-language alignment decoder on top of the VL-Transformer.

VISION-LANGUAGE ALIGNMENT DECODER The whole model is supervised to predict the object-word alignment matrix $\mathcal{A}$ from the VL-Transformer's outputs (v', q'). First, (v', q') are projected into a joint space using a feed-forward layer with layer normalization (Ba et al. 2016) and residual connection. We obtain $(\hat{v}, \hat{q})$, from which we compute $\mathcal{A}$:

$$\mathcal{A} = \frac{\hat{q} \otimes \hat{v}}{\sqrt{d}} \tag{8.1}$$

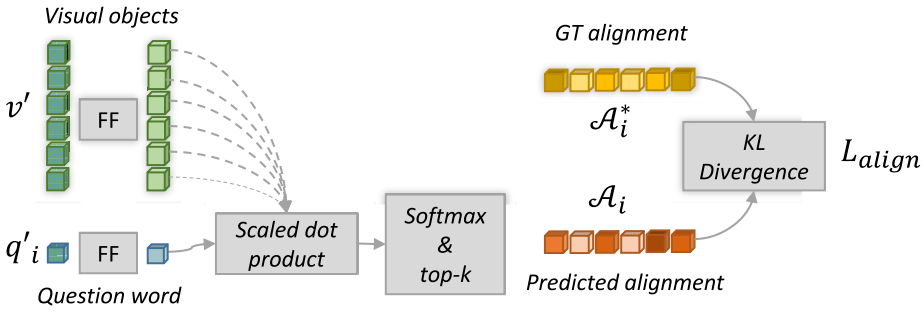


Figure 8.3 – The proposed vision-language alignment decoder and the respective weakly-supervised loss. In this illustration, we present the alignment prediction $\mathcal{A}_i$ between one word q'_i and the visual objects v'. FF stands for feed-forward layers.

where $\otimes$ is the outer product. In other words, the alignment scalar $\mathcal{A}_{ij}$ is computed as the scaled-dot-product between object-word pair (v_i, q_j), as shown in Figure 8.3:

$$\mathcal{A}_{ij} = \frac{\hat{q}_i . \hat{v}_j^T}{\sqrt{d}} \tag{8.2}$$

For each word q_i we only keep the top-k highest predictions and apply a softmax:

$$\mathcal{A}_i = softmax_j(top_k(\mathcal{A}_{ij})) \tag{8.3}$$

In this work, we empirically set $k = 3$. This way, we compute from each word a probability distribution $\mathcal{A}_i$ over the set of visual objects detected by Faster-RCNN. A high probability $\mathcal{A}_{ij}$ means word q_i and object v_j refer to the same high-level entity. The dedicated loss L_{align} is defined using Kullback-Leibler (KL) divergence:

$$L_{align} = KL(\mathcal{A}^*, \mathcal{A}) \tag{8.4}$$

where $\mathcal{A}^*$ is the GT alignment.

SOFT ALIGNMENT SCORE: APPROXIMATING $\mathcal{A}^*$ Let's suppose we have the ground truth object-word pair $(q_i, b^*_{q_i})$. This pair is composed of a word or group of words q_i taken from the input sentence and a bounding box $b^*_{q_i}$ indicating the position of the respective object in the image (provided in GQA). However, we cannot directly use this supervision because both ground truth object-word annotations and the object detector are imperfect. More precisely, (1) the ground truth visual-object annotation is often misaligned with the object detection's bounding box prediction, or (2) the annotated object can simply be not detected at all. To address this issue, we set up a soft-alignment score taking into account both the detection-annotation misalignment and the object detector imperfection. To this end, we consider two criteria: the position one and the semantic one.

POSITION CRITERION For each ground truth object-word pair $(q_i, b^*_{q_i})$, we compute Intersection over Union (IoU) between the object detector's predicted bounding box b_{v_j} and the ground truth object's bounding box $b^*_{q_i}$:

$$P\mathcal{A}^*_{ij} = IoU(b^*_{q_i}, b_{v_j}) \tag{8.5}$$

A high IoU leads to a high position criterion value. Therefore, this criterion allows giving more importance to objects detected in the same image region as the GT object.

SEMANTIC CRITERION Since we cannot only rely on positional information, we also have to take into account the semantics of the object detector's prediction. This would avoid aligning a word with a well-localized but a semantically-different object (according to the detector). Therefore, we define the semantic criterion which computes the semantic similarity between a word q_i and the object's class c_{v_j} – and attribute a_{v_j} – predicted by the detector:

$$SA^*_{ij} = \frac{3}{4} S(q_i, c_{v_j}) + \frac{1}{4} S(q_i, a_{v_j}) \tag{8.6}$$

where $S(\cdot, \cdot)$ compute the cosine similarity between the GloVe embeddings of the class/attribute names. We bias the similarity toward object class, as we empirically found it more relevant than the attribute prediction.

Finally, we combine the two criteria in order to obtain a soft alignment score for each object-word pair in the annotation:

$$\mathcal{A}^*_{ij} = \frac{norm_j(PA^*_{ij}) + norm_j(SA^*_{ij})}{2} \tag{8.7}$$

The resulting soft-alignment scores are normalized over the objects such as:

$$\sum_{j}^{n_{objects}} \mathcal{A}^*_{ij} = 1 \tag{8.8}$$

Hence the ground truth soft alignment score $\mathcal{A}^*_i$ of word q_i is a probability distribution over the set of visual objects detected by the object detector. The soft alignment score defined in this chapter is by construction incomplete and approximate. It is for this reason that we refer to the designed supervision signal as weak, according to the definition of "*weak supervision*" in (Zhou 2018).

8.2.3 *Experimental evaluation*

We now study in what extent the weak supervision of the object-word alignment improve the reasoning. For this purpose, we evaluate the encoder on the VQA task, and in particular, on the GQA dataset. In order to further evaluate the generalization of our conclusions on other tasks requiring reasoning, we also conduct an evaluation on the language-driven comparison of images task, using the Natural Language for Visual Reasoning (NLVR2) dataset (Suhr et al. 2019). The latter is composed of triplets $(img_1, img_2, sentence)$ where img_1 and img_2 are two images and $sentence$ is a sentence describing one or both images. The goal is to predict if the sentence is true. It is worth noticing that NLVR2 data is not viewed during the encoder training, therefore it truly evaluates the generalization capacity of our method.

Table 8.1 – Evaluation of the proposed object-word alignment weak supervision on the GQA (Hudson et al. 2019b) dataset. The presented results are calculated on the dataset's test-std split. The GQA's accuracy is presented in the last column. The exact definitions of all other (auxiliary) metrics can be found in (Hudson et al. 2019b). † means that the model relies on the supervision of the scene graph predictor. B=Binary; O=Open; V=Validity; P=Plausibility; C=Consistency; D=Distribution; Acc=Overall accuracy.

Models	B	O	V	P	C	D	Acc.
Human (Hudson et al. 2019b)	91.2	87.4	98.9	97.2	98.4	-	89.3
UpDn (Anderson et al. 2018)	66.6	34.8	96.2	84.6	78.7	6.0	49.7
MAC (Hudson et al. 2018)	71.23	38.9	96.2	84.5	81.6	5.3	54.1
LCGN (Hu et al. 2019)	73.7	42.3	**96.5**	**84.8**	84.7	4.7	57.0
LXMERT (Tan et al. 2019)	77.2	45.5	96.4	84.5	89.6	5.7	60.3
NSM (Hudson et al. 2019a) †	**78.9**	**49.3**	96.4	84.3	**93.3**	**3.7**	**63.2**
ours	*76.9*	*46.1*	*96.3*	*84.7*	*89.7*	*5.3*	*60.5*

8.2.3.1 *Setup*

DATASET Following Tan et al. (2019), we train our encoder on the concatenation of several corpuses: MSCOCO (Lin et al. 2014), Visual Genome (Krishna et al. 2017), VQAv2 (Goyal et al. 2017), GQA (Hudson et al. 2019b) and VG-QA (Krishna et al. 2017). Consequently, our dataset is composed of 9.18M image-sentence pairs (a sentence can be either a caption or a question).

The GT object-word alignment scores are calculated based on the annotations extracted from GQA and Visual Genome. In GQA dataset, salient question words and answers are annotated with visual pointers. A visual pointer consists of a bounding box corresponding to the visual region described by the words composing the question or the answer. Nevertheless, as GQA represents only 12% of the dataset, the use of the GQA pointers would have been insufficient.

To alleviate this issue, we augment the pointer annotation with visual grounded annotations from Visual Genome. Every Visual Genome image is accompanied by visual region descriptions forming *(description, bounding box)* pairs. Unlike in GQA, descriptions are full descriptive sentences and not small groups of words. Therefore, the so obtained pointer is less discriminative towards the language part. Thus, we choose to combine these descriptions in order to obtain sentences with one, two or three pointers. For instance, the two descriptions "*the cat playing near the tree*" and "*the yellow bird*" become "*the cat playing near the tree and the yellow bird*", with the associated bounding boxes.

All in all, by combining annotations from GQA and Visual Genome, we gather roughly 6M image-sentence pairs annotated with pointers. In other words, about 70% of the total number of the image-sentence pairs in the dataset have fine-grained object-word alignment annotations.

Note: thus research was conducted prior to the creation of our GQA-OOD dataset (introduced in Chapter 5.)

Table 8.2 – Impact of the proposed object-word alignment weak supervision on the VQA task. The presented results are calculated on the GQA (Hudson et al. 2019b) test-std split.

Models	Consistency	Accuracy
ours (w/o alignment supervision)	79.5	54.9
ours (with alignment supervision)	**89.7**	**60.5**

ARCHITECTURE We use the VL-Transformer architecture defined in Chapter 3. We use the original version defined in Tan et al. (2019), with a hidden size $d = 768$ and a multi-head number h=12.

PRE-TRAINING DETAILS We train our vision language encoder using the Adam optimizer (Kingma et al. 2014) during 20 epochs. However, the VQA supervision is only added after 10 epochs, following Tan et al. (2019). We set the learning rate to 10^{-4} with warm starting and learning rate decay. The batch size is 512. Training is done on four P100 GPUs.

FINE-TUNING DETAILS For NLVR2 (Suhr et al. 2019), we use the same fine-tuning strategy as in Tan et al. (2019). Thus, we concatenate the two encoder's output $[CLS]$ embeddings – obtained with $(img_1, sentence)$ and $(img_2, sentence)$ pairs – and pass them through a feed-forward layer. We then use a binary cross-entropy loss. We fine-tune during 4 epochs using Adam optimizer (Kingma et al. 2014). The learning rate is set to $5 * 10^{-5}$ and the batch size is 32. We only supervise with the task-specific binary objective, *i.e.* we drop all the supervision signals used for encoder training. For the GQA result, we directly evaluate our pre-trained model without any fine-tuning step.

8.2.3.2 *Results*

VISUAL QUESTION ANSWERING Table 8.1 compares the results of applying our vision-language encoder on the VQA task versus the recent published works. As one may observe, our model obtains the 2nd-best SOTA result [1], just after the NSM model (Hudson et al. 2019a). The latter is fundamentally different from our approach (contrary to NSM, our approach does not rely on the supervision of the scene graphs predictor). Moreover, it is important to highlight that, unlike previous work (Tan et al. 2019; Lu et al. 2019), our model has not been fine-tuned on the target dataset after the main training step – *i.e.* we kept the same encoder and prediction head used in the pre-training step – making the obtained result even more significant.

In order to quantify the impact of our object-word alignment weak supervision on the VQA task, we evaluate the two versions of our model, with and without the proposed loss, on the GQA dataset. The results are reported in Table 8.2. One may observe that the proposed weak supervision boosts the accuracy with +5.6 points. Moreover, when we focus on the consistency metric, our weakly-supervised alignment allows gaining more

1. At the time of writing the related publication, in December 2019.

Table 8.3 – Evaluation of the proposed object-word alignment weak supervision on the NLVR2 evaluation splits. Models marked with * have been ran by the authors of (Suhr et al. 2019).

Models	Dev.	Test-P
MAC* (Hudson et al. 2018)	50.8	51.4
FiLM* (Perez et al. 2018)	51.0	52.1
CNN+RNN* (Suhr et al. 2019)	53.4	52.4
MaxEnt (Suhr et al. 2019)	54.1	54.8
LXMERT (Tan et al. 2019)	74.9	74.5
ours	**75.8**	**75.5**

Table 8.4 – Impact of the proposed object-word alignment weak supervision on the Visual Reasoning grounded by Language task. The presented results are calculated on the Test-P set of the NLVR2 dataset.

Models	Test-P	Unbalanced	Balanced
ours (w/o alignment sup.)	74.5%	76.0%	73.1%
ours (with alignment sup.)	**75.5%**	**77.2%**	**74.5%**

than +10 points. This demonstrates that, by enforcing the model to explicitly align words with visual objects, we obtained a finer multimodal representation.

NATURAL LANGUAGE FOR VISUAL REASONING (NLVR2) As shown in Table 8.3, our method outperforms the published SOTA accuracy on NLVR2 with a gain of +1 point [2]. Furthermore, we have performed the same ablation analysis as for the VQA task (*i.e.* with and without the object-word alignment weak supervision), and the obtained results are summarized in Table 8.4. These results are coherent with those calculated on the VQA task confirming the advantage of the proposed supervision. Note that the scores in Table 8.4 are reported both for unbalanced and balanced subsets of the NLVR2 dataset. This split takes into account the visual biases present in the dataset. The benefit of our fine-grained alignment supervision method is constant between both subsets, showing that the gain is not caused by learning shortcuts.

8.2.3.3 *Visualizing Reasoning*

In Figure 8.4, we inspect the attention maps inside the inter-modality transformers, which illustrates the information flow between the two modalities (vision and language) [3]. Generally, attention maps convey information on the importance that a neural map poses

2. At the time of writing the related publication, in December 2019.

3. As a side note, this visualization has been generated before the conception of VISQA. It is worth noticing that VISQAis perfectly suited for this type of analysis.

Figure 8.4 – Visualization of the attention maps of the penultimate (=4th) inter-modality transformer. Word-object alignment does not emerge naturally for the baseline (without object-word alignment supervision), whereas our model with the proposed weakly-supervised objective learns to pay strong cross-attention on co-occurring combinations of words and objects in the scene. In the attention maps, rows represent words and columns represent visual objects. For the sake of visibility, we display the bounding box of the detected object with the highest activation regarding the selected word. The predicted answer (underlined) is written after the question. Its corresponding language token is $[CLS]$, *i.e.* the first row in attention maps.

on local areas in input or activations. In the particular case of our model, the inter-modality attention map visualizes how modalities are fused by the model, as they give weight to outputs for a given word as a function of a given object (or vice-versa).

The effectiveness of the new object-word alignment objective is corroborated by attention units which are higher for object-word pairs referring to the same entity in our model. We observe a radically different behavior in the baseline's attention maps, where attention is less-fine grained: roughly uniform attention distributions indicate that the layer outputs of all words attend to roughly the same objects.

CAVEAT We do not want to imply, that the exact word-object alignment in the inter-modality layer is indispensable for a given model to solve a reasoning task, as a complex neural network can model relationships in the data in various different layers. However, we do argue, that some form of word-object alignment is essential for solving vision-language tasks, as the model is required to query whether concepts from the question are present in the image, and possibly query their relationships to other concepts. Inductive bias has been added to the model for this type of reasoning in the form of inter-modality layers, and it is therefore natural to inspect whether this cross-attention emerges at this

exact place. We would also like to point out that we do not force or favor word-object alignment at a specific layer, as our proposed supervision signal is injected through a new module attached to the inter-modality layer (see Figure 8.2). The attention maps show that the supervision signal is successfully propagated from the new alignment head to the inter-modality layer.

8.3 SAMPLE COMPLEXITY OF REASONING SUPERVISION

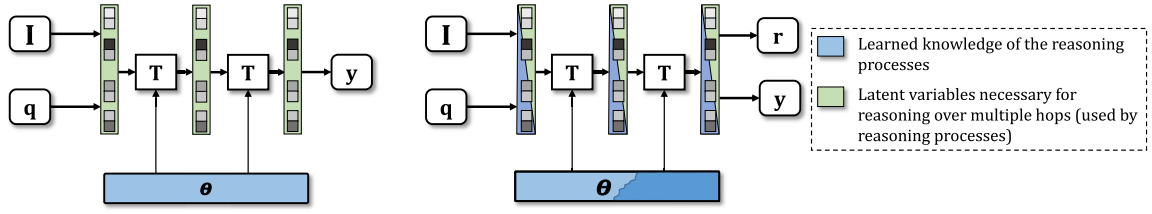
In the previous section, we experimentally show the benefit of guiding the reasoning process through supervision. *Why does this additional supervision help to learn reasoning ?*.

In this section we focus on supervising reasoning programs, i.e. we suppose that the exact logical function computing the output answer from input is known during training time (not during testing). We provide a theoretical analysis indicating that the prediction and supervision of reasoning can improve learnability in vision and language reasoning under some assumptions. We back up this claim through a theoretical analysis showing decreased sample complexity under mild hypotheses. This will be experimentally confirmed in chapter 9.

8.3.1 *Measuring complexity of learning problems*

Measuring complexity of learning problems and thus generalization, has been a goal of theoretical machine learning since the early days, with a large body of work based on PAC-Learning (Valiant 1984; S. Shalev-Shwart et al. 2014). Traditionally, bounds have been provided ignoring data distributions and focusing uniquely on hypothesis classes (network structures in neural network language), *e.g.* as measured by VC-dimension. Surprising experimental results on training networks on random samples have seemingly contradicted learning theory (Zhang et al. 2017), in particular Rademacher Complexity. To cope with this, we use the modern estimators of sample complexity developed for the deep learning era (see Belkin (2021) for an overview), which provide the possibility of calculating tighter bounds under the assumption that learning is performed by over-parametrized deep networks and stochastic gradient descent. These estimators are data-dependent and as such more powerful.

Within this framework, in particular the work of Arora et al. (2019), sample complexity is linked to the functional form of the decision function directly. If the functional form is simpler, learning it requires fewer samples. Arora et al. (2019) provides a direct way to estimate sample complexity, if the functional form is known, or through its estimation from training data in the form of a stochastic Gram matrix. Algorithmic alignment between neural network structures and the decomposition of underlying reasoning functions has been studied in Xu et al. (2020), with a focus on algorithms based on dynamic programming. Our theoretical contribution in Section 8.3.2 builds on the latter two methodologies and extends this type of analysis to intermediate supervision of reasoning programs.



(a) Classical training, CE loss on answers **y**

(b) CE loss on **y** + reasoning supervision **r**

Figure 8.5 – VQA takes visual input $\boldsymbol{v}$ and a question $\boldsymbol{q}$ and predicts a distribution over answers $\boldsymbol{y}$. (a) Classical discriminative training encodes the full reasoning function in the network parameters θ, while the network activations contain latent variables necessary for reasoning over multiple hops. (b) Additional reasoning supervision requires intermediate network activations to contain information on the reasoning process, simplifying learning the reasoning function g. Under the hypothesis of its decomposition into multiple reasoning modes, intermediate supervision favors separately learning the mode selector and each individual mode function. This intuition is analyzed theoretically in Section 8.3.2.

We here briefly recall the notion of sample complexity, in the context of PAC-learning (Valiant 1984), which characterizes the minimum amount (=M) of samples necessary to learn a function with sufficiently low $(=\epsilon)$ error with sufficiently high $(=\delta)$ probability:

DEFINITION 8.3.1 (SAMPLE COMPLEXITY). *Given an error threshold ϵ>0; a threshold on error probability δ; a training set $S = \{x_i, y_i\}$ of M i.i.d. training samples from $\mathcal{D}$, generated from some underlying true function $y_i = g(\boldsymbol{x}_i)$, and a learning algorithm $\mathcal{A}$, which generates a function f from training data,* e.g. *$f = \mathcal{A}(S)$; Then g is (M, ϵ, δ)-learnable by $\mathcal{A}$ if*

$$\mathbb{P}_{x\sim D}\left[||f(x) - g(x)|| \leq \epsilon\right] \geq 1 - \delta \tag{8.9}$$

8.3.2 *Reasoning supervision reduces sample complexity*

In what follows, we denote with g "true" (but unknown) underlying reasoning functions, and by f functions approximating them, implemented as neural networks. The goal is to learn a function g able to predict a distribution $\boldsymbol{y}$ over answer classes given an input question and an input image, see Figure 8.5a. While in the experimental part we use state-of-the-art Transformer based models, in this theoretical analysis, we consider a simplified model, which takes as input the two vectorial embeddings $\boldsymbol{q}$ and $\boldsymbol{v}$ corresponding to, respectively, the question and the visual information (image), for instance generated by a language model and a convolutional neural network, and produces answers $\boldsymbol{y}^*$ as:

$$\boldsymbol{y}^* = g(\boldsymbol{q}, \boldsymbol{v}) \tag{8.10}$$

We restrict this analysis to two-layer MLPs, as they are easier to handle theoretically than modern attention based models. The reasoning function g is approximated by a neural network f parametrized by a vector θ and which predicts output answers $\boldsymbol{y}$ as:

$$\boldsymbol{y} = f(\boldsymbol{q}, \boldsymbol{v}, \theta) \tag{8.11}$$

Our analysis uses PAC-learning (Valiant 1984) and builds on recent results providing bounds on sample complexity taking into account the data distribution itself. We here briefly reproduce Theorem 3.5. from Xu et al. (2020), which, as an extension of a result in (Arora et al. 2019), provides a lower bound for sample complexity of overparametrized MLPs with vectorial outputs, *i.e.* MLPs with sufficient capacity for learning a given task:

THEOREM 8.3.2 (SAMPLE COMPLEXITY FOR OVERPARAMETRIZED MLPS). *Let $\mathcal{A}$ be an overparametrized and randomly initialized two-layer MLP trained with gradient descent for a sufficient number of iterations. Suppose $g : \mathbb{R}^d \rightarrow \mathbb{R}^m$ with components $g(x)^{(i)} = \sum_j \alpha_j^{(i)} (\boldsymbol{\beta}_j^{(i)T} x)^{p_j^{(i)}}$, where $\beta_j^{(i)} \in \mathbb{R}^d$, $\alpha^{(i)} \in \mathbb{R}$, and $p_j^{(i)} = 1$ or $p_j^{(i)} = 2l, l \in \mathbb{N}_+$. The sample complexity $\mathcal{C}_{\mathcal{A}}(g, \epsilon, \delta)$ is*

$$\mathcal{C}_{\mathcal{A}}(g, \epsilon, \delta) = O\left(\frac{\max_i \sum_j p_j^{(i)} |\alpha_j^{(i)}| \cdot ||\mathbf{fi}_j^{(i)}||_2^{p_j^{(i)}} + \log(\frac{m}{\delta})}{(\epsilon/m)^2} \right) \tag{8.12}$$

We use the following *Ansatz*: since each possible input question requires a potentially different form of reasoning over the visual content, our analysis is based on the following assumption.

ASSUMPTION 1. *The unknown reasoning function $g()$ is a mixture model which decomposes as follows*

$$\boldsymbol{y}^* = \sum_r \boldsymbol{\pi}_r \boldsymbol{h}_r = \sum_r \boldsymbol{\pi}_r g_r(\boldsymbol{v}), \tag{8.13}$$

where the different mixture components r correspond to different forms of reasoning related to different questions. The mixture components can reason on the visual input only, and the mixture weights are determined by the question $\boldsymbol{q}$, i.e. *the weights $\boldsymbol{\pi}$ depend on the question $\boldsymbol{q}$,* e.g. *$\boldsymbol{\pi} = g_\pi(\boldsymbol{q})$.*

We call $g_\pi(.)$ the *reasoning mode estimator*. One hypothesis underlying this analysis is that learning to predict fine-grain alignment or reasoning programs (*cf.* Chapter 9) allows the model to more easily decompose into the form described in Equation 8.13, *i.e.* that the network structure closely mimics this decomposition, as information on the different reasoning modes r is likely to be available in the activations of intermediate layers, *cf.* Figure 8.5. This will be formalized in Assumption 3 and justified further below.

Considering the supposed "true" reasoning function $\boldsymbol{y}^* = g(\boldsymbol{q}, \boldsymbol{v})$ and its decomposition given in Equation 8.13, we suppose that each individual reasoning module g_r can be approximated with a multi-variate polynomial, in particular each component $\boldsymbol{h}_r^{(i)}$ of the vector $\boldsymbol{h}_r$, as:

$$\boldsymbol{h}_r^{(i)} = g_r(\boldsymbol{v}) = \sum_j \alpha_{r,j}^{(i)} (\beta_{r,j}^{(i)T} \boldsymbol{v})^{p_{r,j}^{(i)}} \quad \text{with params.} \quad \omega = \left\{ \alpha_{r,j}^{(i)}, \beta_{r,j}^{(i)}, p_{r,j}^{(i)}. \right\} \tag{8.14}$$

A trivial lower bound on the complexity of the reasoning mode estimator $g_\pi(.)$ is the complexity of the identity function, which is obtained in the highly unlikely case where the question embeddings $\boldsymbol{q}$ contain the 1-in-K encoding of the choice of reasoning mode r. We adopt a more realistic case as the following assumption.

ASSUMPTION 2. *The input question embeddings $\boldsymbol{q}$ are separated into clusters according to reasoning modes r, such that the underlying reasoning mode estimator g_π can be realized as a NN classifier with dot-product similarity in this embedding space.*

Under this assumption, the reasoning mode estimator can be expressed as a generalized linear model, *i.e.* a linear function followed by a soft-max σ:

$$\boldsymbol{\pi} = g_\pi(\boldsymbol{q}) = \sigma\left(\left[\gamma_0^T \boldsymbol{q},\ \gamma_1^T \boldsymbol{q}, ...\right]\right) \tag{8.15}$$

where the different γ_r are the cluster centers of the different reasoning modes r in the question embedding space. As the softmax is a monotonic non-linear function, its removal will not decrease sample complexity [4], and the complexity can be bounded by the logits $\boldsymbol{\pi}_r = \gamma_r^T \boldsymbol{q}$. Plugging this into Equation 8.13 we obtain that each component $\boldsymbol{y}^{*(i)}$ of the answer is expressed as the following function:

$$\boldsymbol{y}^{*(i)} = \sum_r \left(\gamma_r^T \boldsymbol{q}\right) \sum_j \alpha_{r,j}^{(i)} (\beta_{r,j}^{(i)T} \boldsymbol{v})^{p_{r,j}^{(i)}} \tag{8.16}$$

We can reparametrize this function by concatenating the question $\boldsymbol{q}$ and the visual input $\boldsymbol{v}$ into a single input vector $\boldsymbol{x}$, which are then masked by two different binary masks, which can be subsumed into the parameters γ_r and $\beta_{r,j}^{(i)}$, respectively:

$$\boldsymbol{y}^{*(i)} = \sum_r \sum_j (\gamma_r^T \boldsymbol{x}) \alpha_{r,j}^{(i)} (\beta_{r,j}^{(i)T} \boldsymbol{x})^{p_{r,j}^{(i)}} \tag{8.17}$$

Extending Theorem 3.5. from Xu et al. (2020), we can give our main theoretical result as the sample complexity of this function, expressed as the following theorem.

THEOREM 8.3.3 (SAMPLE COMPLEXITY FOR MULTI-MODE REASONING FUNCTIONS). *Let $\mathcal{A}$ be an overparametrized and randomly initialized two-layer MLP trained with gradient descent for a sufficient number of iterations. Suppose $g : \mathbb{R}^d \to \mathbb{R}^m$ with components $g(x)^{(i)} = \sum_r \sum_j (\gamma_r^T \boldsymbol{x}) \alpha_{r,j}^{(i)} (\beta_{r,j}^{(i)T} \boldsymbol{x})^{p_{r,j}^{(i)}}$ where $\gamma_r \in \mathbb{R}^d$, $\beta_{r,j}^{(i)} \in \mathbb{R}^d$, $\alpha_{r,j}^{(i)} \in \mathbb{R}$, and $p_{r,j}^{(i)} = 1$ or $p_{r,j}^{(i)} = 2l, l \in \mathbb{N}_+$. The sample complexity $\mathcal{C}_\mathcal{A}(g, \epsilon, \delta)$ is*

$$\mathcal{C}_\mathcal{A}(g, \epsilon, \delta) = O\left(\frac{\max_i \sum_r \sum_j \pi p_{r,j}^{(i)} |\alpha| \cdot ||\gamma_r||_2 \cdot ||\beta_{r,j}||_2^{p_{r,j}^{(i)}} + \log(m/\delta)}{(\epsilon/m)^2}\right).$$

The proof of this theorem is given in Section A.1.

Theorem 8.3.3 provides the sample complexity of the reasoning function $g()$ under classical training. In the case of program supervision, our analysis is based on the following assumption (see also Figure 8.5b):

4. In principle, there should exist special degenerate cases, where an additional softmax could reduce sample complexity; however, in our case it is applied to a linear function and thus generates a non-linear function.

ASSUMPTION 3. *Supervising reasoning encodes the choice of reasoning modes r into the hidden activations of the network f. Therefore, learning is separated into several processes,*

(a) learning of the reasoning mode estimator $g_\pi()$ approximated as a network branch $f_\pi()$ connected to the program output;

(b) learning of the different reasoning modules $g_r()$ approximated as network branches $f_r()$ connected to the different answer classes $\boldsymbol{y}_r$; each one of these modules is learned independently.

We justify Assumption 3.a through supervision directly, which separates $g_\pi()$ from the rest of the reasoning process. We justify Assumption 3.b by the fact that different reasoning modes r will lead to different hidden activations of the network. Later layers will therefore see different inputs for different modes r, and selector neurons can identify responsible inputs for each branch $f_r()$, effectively switching off irrelevant input.

We can see that these complexities are lower than the sample complexity of the full reasoning function given in Theorem 8.3.3, since for a given combination of i, r, j, the term $||\gamma_r||_2 \cdot ||\beta_{r,j}||_2^{p_{r,j}^{(i)}}$ dominates the corresponding term $||\beta_{r,j}||_2^{p_{r,j}^{(i)}}$. Let us recall that the different vectors γ correspond to the cluster centers of reasoning modes in language embedding space. Under the assumption that the language embeddings $\boldsymbol{q}$ have been created with batch normalization, a standard technique in neural vision and language models, each value $\gamma_r^{(i)}$ follows a normal distribution $\mathcal{N}(0,1)$. Dropping indices i, r, j to ease notation, we can then compare the expectation of the term $||\gamma||_2 \cdot ||\beta||_2^p$ over the distribution of γ and derive the following relationship:

$$\mathbb{E}_{\gamma^{(i)} \sim N(0,1)} ||\gamma||_2 \cdot ||\beta||_2^p \;=\; C||\beta||_2^p \;=\; \sqrt{2}\frac{\Gamma(\frac{m}{2}+\frac{1}{2})}{\Gamma(\frac{m}{2})}||\beta||_2^p \tag{8.18}$$

where Γ is the Gamma special function and m is the dimension of the language embedding γ. We provide a proof for this equality in A.2.

8.3.2.1 *Discussion and validity of our claims*

The difference in sample complexity is determined by the factor C in Equation 8.18, which monotonically grows with the size of the embedding space m, which is typically in the hundreds. For the order of m=512 to m=768 used for state-of-the-art LXMERT models (Tan et al. 2019), complexity grows by a factor of around $\sim$20.

We would like to point out, that this analysis very probably under-estimates the difference in complexity, as the difference very much depends on the complexity of the reasoning estimator $\boldsymbol{\pi}$, which we have simplified as a linear function in Equation 8.15. Taking into account just the necessary soft-max alone would probably better appreciate the difference in complexity between the two methods, which we leave for future work. Our analysis is also based on several assumptions, among which is the simplified model (an over-parametrized MLP instead of an attention based network), as well as assumptions of Theorem 8.3.3 from Xu et al. (2020) and Arora et al. (2019), on which our analysis is based.

Lastly, we would like to comment on the fact that we compare two different bounds: (i) the bound on sample complexity for learning the full multi-modal reasoning given in Theorem 8.3.3, and (ii) the bound for learning a single reasoning mode given by Theorem 8.3.2. While comparing bounds does not provide definitive answers on the order of models, both bounds have been derived by the same algebraic manipulations, and we claim that they are comparable.

We also provide an experimental evaluation of the sample complexity of both variants, with and without program supervision, in section 9.3.2.2, Figure 9.6.

8.4 CONCLUSION

In this chapter, we have demonstrated that it is possible to improve the reasoning abilities of VQA models by designing an additional supervision loss. In particular, we propose to guide the learning of reasoning during the VL-Transformer training through the weak supervision of the fine-grained word-object alignment. We experimentally show that our method improves the performance on GQA, and generalizes well to the language-driven comparison of images, another visual reasoning task. Furthermore, our experiments are supported by a theoretical analysis, providing cues on the benefit of this additional supervision. More precisely, we leverage theorems from PAC-learning to demonstrate that program supervision can decrease sample complexity, under reasonable hypothesis. In the next Chapter 9, we will show how this reasoning supervision can be used as a catalyst for transferring reasoning patterns learned on perfect training conditions.

Chapter 9

TRANSFERRING REASONING PATTERNS

9.1 INTRODUCTION

"*Learning to reason*", what does it mean? As already stated in Chapter 2, providing a general definition of "*reasoning*" is difficult. Following Bottou (2014), we define "*reasoning*" as "*algebraically manipulating previously acquired knowledge in order to answer a new question*". We also specify that, in the context of ML, "*reasoning*" can be defined as the opposite of shortcut learning (Geirhos et al. 2020). As such, we can assess that a VQA model performs reasoning if it has learned decision rules which perform well on the training set, in-distribution and all relevant OOD test sets. In Chapter 7, we provided evidence that deep neural networks can learn to reason, when training conditions are favorable enough, *i.e.* when uncertainty and noise in visual inputs is reduced. In particular, we highlighted the existence of *reasoning patterns* at work in the attention layers learned by a Transformer-based VQA model trained on perfect (oracle) visual input. On the contrary, when comparing this visual oracle with a standard VQA model (*i.e.* with uncertain visual input), we discover that the observation does not hold.

In this chapter, we wonder: *are reasoning patterns transferable?* In other words, is it possible to transfer, or adapt, the ability of reasoning (modularity, generalization, etc.) learned in favorable conditions to a less favorable setup where the vision in uncertain? We first propose a naive approach, by fine-tuning the perfectly-sighted oracle model on the real noisy visual input (see Figure 9.1). Using the same analysis and visualization techniques as in Chapter 6, we show that attention modes, absent from noisy models, are transferred successfully from oracle models to deployable[1] models. We report improvements in overall accuracy and OOD generalization.

While this *oracle transfer* method provides strong empirical results and insights on the bottlenecks in problems involving learning to reason, it still suffers from significant loss in reasoning capabilities during the transfer phase, when the model is required to adapt from perfectly clean visual input to the noisy one. We conjecture that reasoning on noisy data involves additional functional components, not necessary in the clean case, due to different types of domain shifts: (1) a *presence shift*, caused by imperfect object detectors, leading to missing visual objects necessary for reasoning, or to multiple

1. In this thesis, we define the term *deployable* as a model that *does not* use GT visual inputs. It is not related to deployment to production.

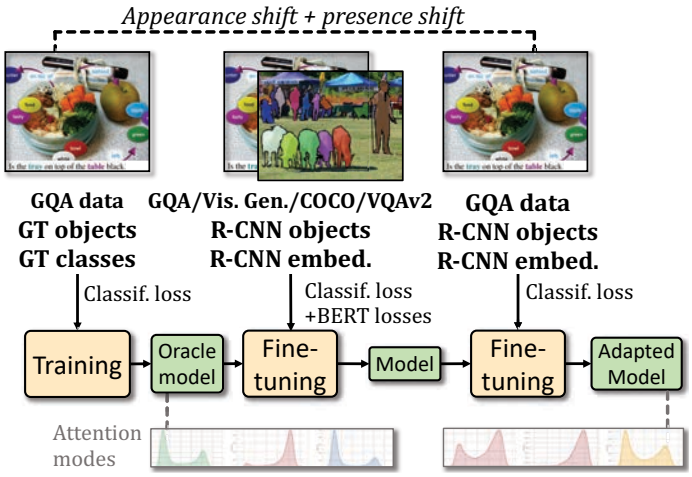


Figure 9.1 – We argue that noise and uncertainties in visual inputs are the main bottleneck in VQA preventing successful learning of reasoning capacities. In a deep analysis, we show that oracle models with perfect sight, trained on noiseless visual data, tend to depend significantly less on bias exploitation. We exploit this by training models on data without visual noise, and then transfer the learned reasoning patterns to real data. We illustrate successful transfer by an analysis and visualization of attention modes.

(duplicate) detections; and (2) an *appearance shift* causing variations in object embeddings (descriptors) for the same class of objects due to different appearance.

We then propose to enhance *oracle transfer* by adding a regularization term, minimizing loss of the reasoning capabilities during transfer. In particular, we address this problem through program prediction as an additional auxiliary loss, *i.e.* supervision of the sequence of reasoning operations along with their textual and/or visual arguments. This additional supervision is directly related to Chapter 8, where we demonstrated, experimentally and theoretically, that guiding the reasoning process during training (through supervision) helps to improve the predictions of the VQA model. Therefore, to maintain a strong link between the learned function and its objective during the knowledge transfer phase, when inputs are switched from clean oracle inputs to noisy input, the neural model is required to continue to predict complex reasoning programs from different types of inputs. In an experimental study, we demonstrate the effectiveness of this *guided oracle transfer* on GQA and show its complementarity when combined to BERT-like self-supervised pre-training.

CONTRIBUTIONS OF THE CHAPTER

(i) an *oracle transfer* method allowing to transfer, through fine-tuning, the knowledge learned from perfect visual input to a deployable setting where the visual representation is uncertain.

(ii) an augmented *guided oracle transfer*, leveraging results from Chapter 8 in order to improve the transfer of reasoning patterns by adding a program supervision loss.

(iii) we experimentally demonstrate the efficiency of the reasoning pattern transfer and show that it increases VQA performance on both in- and out-of-distribution sets, even when combined with BERT-like pre-training.

9.2 TRANSFERRING REASONING PATTERNS FROM ORACLE

Our purpose is to transfer the *reasoning patterns* learned by a visual oracle, when the uncertainty in the visual input is reduced, to a standard model taking as input the imperfect image representation extracted by an object detector. Therefore, we propose a method called *oracle transfer*. It consists in first pre-training the VQA model on the oracle (perfect) visual input, and then further training on the standard (noisy) data. In Chapter 7, we conjecture that the uncertainty in vision is one of the major cause leading to shortcut learning. Therefore, we argue that the first optimization steps are crucial for the emergence of specific attention modes, and claim that such oracle pre-training puts the model in favorable condition for avoiding learning shortcuts.

9.2.1 *Method: oracle transfer*

ORACLE TRANSFER As shown in Figure 9.1, training proceeds as follows:

1. Training of a perfectly-sighted oracle model on GT visual inputs from the GQA annotations, in particular a *symbolic* representation concatenating the 1-in-K encoded object class and attributes of each object.
2. Initializing a new model *with the oracle parameters*. This new model is taking noisy visual input in a form of the *dense* representation (2048-dim feature vector extracted by Faster-RCNN (Ren et al. 2015) fused with bounding-boxes). The first visual layers (T^V_{-}) are initialized randomly due to the difference in nature between dense and symbolic representations.
3. Optionally and complementary, *continue training* with large-scale self-supervised objectives (LXMERT (Tan et al. 2019)/BERT-like) on combined data from Visual Genome (Krishna et al. 2017), MS COCO (Lin et al. 2014), VQAv2 (Goyal et al. 2017).
4. *Fine-tuning* with the standard VQA classification objective on the target dataset (GQA or VQAv2).

9.2.2 *Experimental evaluation*

9.2.2.1 *Setup*

DATASET Our models are trained on the balanced GQA (Hudson et al. 2019b) training set (∼1M question-answer pairs). LXMERT pretraining is done on the on a corpus gathering images and sentences from MSCOCO (Lin et al. 2014) and VisualGenome (Krishna et al. 2017). Note that, as the GQA dataset is built upon VisualGenome, the original LXMERT pre-training dataset contains samples from the GQA validation split. Therefore, *we removed these validation samples from the pre-training corpus*, in order to be able to validate on the GQA validation split. We evaluate on the GQA, our own GQA-OOD (*cf.* Chapter 5) and VQAv2 (Goyal et al. 2017) datasets.

ARCHITECTURE We use the same compact VL-Transformer architecture as defined in Chapter 3.

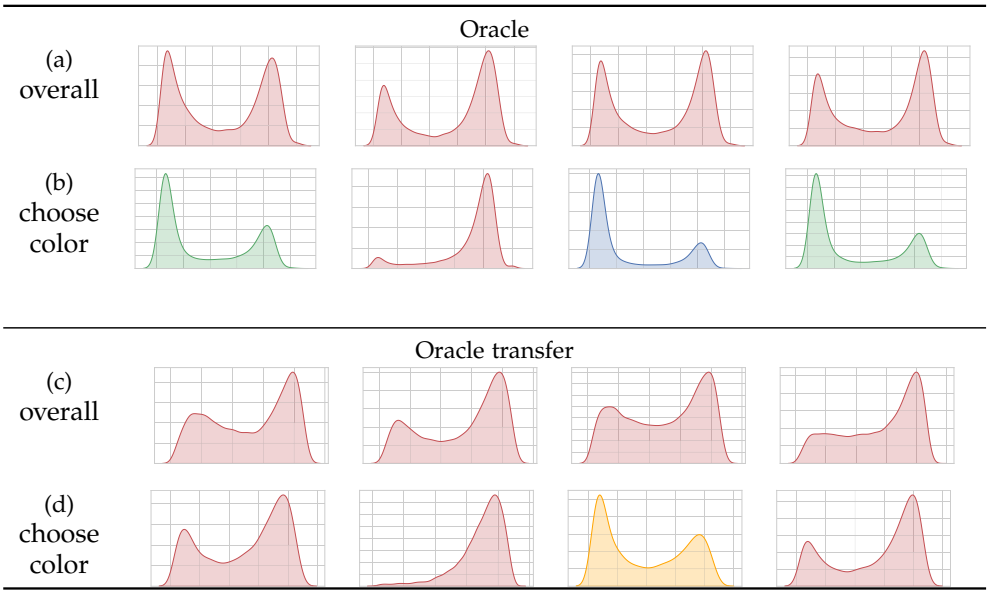


Figure 9.2 – We reproduce Figure 7.5, (a) and (b), with our VL-Transformer + dense *Oracle Transfer* (same heads/layers). As we can see in (c), the attention heads have retained their "*bimorph*" property, although their shape is distorted by the noisy visual training. In addition, when we measure the attention mode on questions involving the `choose color` function, in (d), we observe that the attention heads are still function-dependant, although in a lesser extent.

TRAINING DETAILS All models were trained with the Adam optimizer (Kingma et al. 2014), a learning rate of 10^{-4} with warm starting and learning rate decay. Training was done on one P100 GPU. Two P100 GPUs were used for BERT/LXMERT (Tan et al. 2019) pre-training. For the oracle, the batch size was equal to 256. We train during 40 epochs and select the best epoch using accuracy on validation. The oracle transfer follows exactly the same procedure, except when using LXMERT pretraining. In that case, BERT/LXMERT pretraining is performed during 20 epochs max with a batch size of 512. All pretraining losses are added from the beginning, including the VQA one. After pre-training, we fine-tune either on GQA or VQAv2. For GQA, we fine-tune during 4 epochs, with a batch size of 32 and a learning rate equal to 10^{-5}. For VQAv2, we fine-tune during 8 epochs, with a batch size of 32 and a learning rate equal to 10^{-5}. Hyperparameters are selected either on the testdev (for VQAv2) or validation (for GQA-OOD and GQA) sets

9.2.2.2 *Results*

EVALUATING TRANSFER We evaluate the impact of *Oracle Transfer* on three different benchmarks in Table 9.1, observing that transferring knowledge from the oracle significantly boosts accuracy. We also evaluate the effect of *Oracle Transfer* on bias reduction and benchmark on GQA-OOD (*cf.* Chapter 5), reporting gains in OOD settings — rare samples, "*acc-tail*" — by a large margin, which suggests improved generalization ability. Our experiments show that *Oracle Transfer* is complementary to large-scale vision-language self-supervised objectives of type LXMERT/BERT-like pretraining as introduced in (Tan et al. 2019). An overall gain of about +1 accuracy points is observed from models (c) to (d)

Model	Pretraining		GQA-OOD		GQA	VQAv2
	Oracle	LXMERT	acc-tail	acc-head	overall	overall
(a) Baseline			42.9	49.5	52.4	-
(b) *Oracle transfer* (ours)	✓		**48.5**	**55.5**	**56.8**	-
(c) Baseline (+LXMERT)		✓	47.5	54.7	56.8	69.7
(d) *Oracle transfer* (ours) (+LXMERT)	✓	✓	**48.3**	**55.2**	**57.8**	**70.2**

Table 9.1 – Quantitative evaluation of the proposed knowledge transfer from oracle models. All listed models are deployable and based on the same compact VL-Transformer architecture (*cf.* Chapter 3), no GT input is used for testing. Models: (c)+(d) are pretrained with LXMERT (Tan et al. 2019)/BERT-like objectives after *Oracle Transfer*. All scores are obtained on GQA-OOD-testdev (*cf.* Chapter 5); GQA-testdev; VQAv2-test-std. Training hyperparameters selected on respective validation sets.

Method	Input train	Input test	Acc.
(a) Baseline	Dense	Dense	61.7
(b) Transf. w/o retrain	1-in-K GT	1-in-K pred.	58.8
(c) Transf. w/ T^V_- retrain	1-in-K GT	Dense	61.7
(d) Transf. w/ retrain	1-in-K GT	Dense	66.3

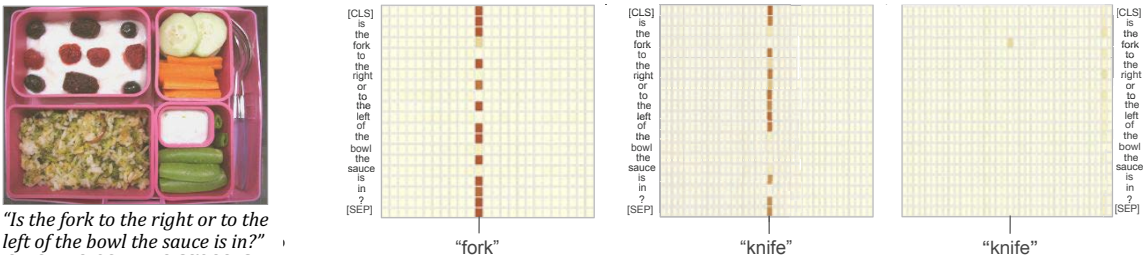
Table 9.2 – Impact of different types of transfer, GQA (Hudson et al. 2019b) val. accuracy. All models are deployable (no GT used for testing).

in Table 9.1, attributed to *Oracle Transfer*. As a comparison, LXMERT/BERT pretraining alone does not improve "*acc-tail*" on GQA-OOD.

CROSS-DATASET TRAINING We explore whether the effects of oracle knowledge generalize beyond the GQA dataset, and evaluate training the oracle on GQA GT annotations, performing LXMERT/BERT pretraining, and transferring to a model trained on the VQAv2 dataset. We improve VQAv2 accuracy by a significant margin, suggesting positive transfer beyond GQA (Table 9.1).

TRANSFER ABLATION STUDIES We evaluate different variants of knowledge transfer, shown in Table 9.2, on the GQA validation set only. We explore a direct transfer from the oracle to a deployable model without retraining, by making visual input representations comparable. To this end, the deployable model receives 1-in-K encoded class information, albeit not from GT classes but taking classes from the Faster R-CNN detector (Table 9.2-b). While inferior to the baseline, its performance is surprisingly high, suggesting that the oracle learns knowledge which is applicable in real/noisy settings. Performance gains are, however, only obtained by finetuning the model to the uncertainties in dense visual embeddings. Retraining only the visual block (Table 9.2-c), performances are on par with the baseline, retraining the full model (Table 9.2-d) gains +4.6 points.

COMPARISON WITH SOTA *Oracle Transfer* allows improving performance of the tiny-LXMERT model both in-distribution and OOD settings (Table 9.8, bottom part). Further-



(a) Oracle (b) Oracle transfer (c) Baseline

Figure 9.3 – Example for the difference in attention in the second $T_{\times}^{L \leftarrow V}$ layer. The oracle drives attention towards a specific object, "*fork*", also seen after transfer but not in the baseline (we checked for permutations). The transferred model overcame a miss-labelling of the fork as a knife. This analysis was performed with our interactive visualization tool VisQA, introduced in Chapter 6.

more, *oracle transfer* is parameter efficient and achieves on-par overall accuracy with MCAN-6 (Kim et al. 2018) while halving capacity.

QUALITATIVE ANALYSIS Finally, we qualitatively study the effects of *Oracle Transfer* by analyzing the attention modes as done in Chapter 7. As shown in Figure 9.2, after transfer, the VL-Transformer preserves the "*bimorph*" property of its attention heads, which was present in the original oracle model (Figure 7.3-a), but absent in the baseline (Figure 7.3-b). In addition, Figure 9.3 shows the attention maps of the $T_{\times}^{L \leftarrow V}$ heads in the second cross-modal layer for an instance. This head, referenced as $VL, 1, 0$ in Figure 7.4, is observed to be triggered to questions such as `verify attr` and `verify color` provided as example. We observe that the oracle model draws attention towards the object "*fork*" in the image, and also, to a lesser extent, in the transferred model, but not in the baseline model. Similar attention patterns were observed on multiple heads in the corresponding cross-modal layer — this analysis took into account possible permutations of heads between models. Interestingly, the miss-classification as a "*knife*" prevents the baseline from drawing attention to it, but not the transferred model.

9.3 GUIDING THE ORACLE TRANSFER

Although it provides encouraging experimental results, the *oracle transfer* is still limited. When transferring the reasoning patterns from oracle to noisy settings, two different shifts have to be addressed:

(1) a *presence shift*: as contrary to the oracle settings, the imperfect object detection in standard setting causes some objects to be not detected, falsely detected or detected multiple times.

(2) an *appearance shift*: while oracle objects are encoded as one-hot vectors, objects extracted using an object detector are better represented using dense vectors.

The *oracle transfer* method mainly addresses the *appearance shift*, through fine-tuning. We now propose to tackle the *presence shift*.

Method	$\|\Theta\|$	O	L	OOD	GQA
UpDn (Anderson et al. 2018)	22			42.1	51.6
BAN-4 (Kim et al. 2018)	50			47.2	54.7
MCAN-6 (Yu et al. 2019)	52			46.5	56.3
***Oracle transfer* (ours)**	26	✓		**48.5**	**56.8**
LXMERT-tiny	26		✓	47.5	56.8
LXMERT-tiny + *Oracle transfer* (ours)	26	✓	✓	48.3	57.8
LXMERT (Tan et al. 2019)	212		✓	49.8	59.6

$|\Theta|$ = number of parameters (M); OOD = GQA-OOD Acc-tail.
O = *Oracle Transfer*, L = LXMERT/BERT pretraining.

Table 9.3 – Comparison with SOTA on GQA and GQA-OOD (*cf.* Chapter 5) on testdev. Hyperparameters were optimized on GQA-validation.

We draw inspiration from results in Chapter 8, where we demonstrated that supervising the model to learn a fine-grained alignment between the question's words and visual objects helps to reason. Based on these insights, we propose to follow a similar method and guide the reasoning process during the *oracle transfer*. In particular, we propose to supervise the model to predict the whole reasoning steps required to answer the question. Indeed, as described in Figure 9.4, reasoning involves to decompose the question into multiple hops, called operation steps, each operation having a specific function and arguments (question's words or visual objects). In particular, our method is designed to mitigate the *presence shift*, by enforcing the model to identify which words and objects are necessary to answer the question. Thereby, we conjecture that supervising the VQA model to predict these operations during the oracle transfer will help to better transfer the reasoning patterns.

9.3.1 *Method: guided oracle transfer*

We conceived a regularization technique which supervises the prediction of reasoning steps required to answer the question. We therefore assume the existence of the following GT annotation of reasoning programs [2].

A given data sample consists of a sequence $\{\boldsymbol{q}_i\}$ of input question word embeddings, a set $\{\boldsymbol{v}_i\}$ of input visual objects, the ground truth answer class $\boldsymbol{y}^*$ as well as the GT reasoning program, which is structured as a tree involving operations and arguments. Operations $\{\boldsymbol{o}_i^*\}$ are elements of a predefined set {`choose color`, `filter size`, ...}. The arguments of these operations may be taken from (i) all question words, (ii) all visual objects, (iii) all operations — when an operation takes as argument the result of another operation. Hence, arguments are annotated as many-to-many relationships. In the question "*Is there a motorbike or a plane?*", for instance, the operation "`or`" depends on the

2. GT annotation of reasoning programs can be easily obtained in semi-automatically generated dataset such as GQA(Hudson et al. 2019b)

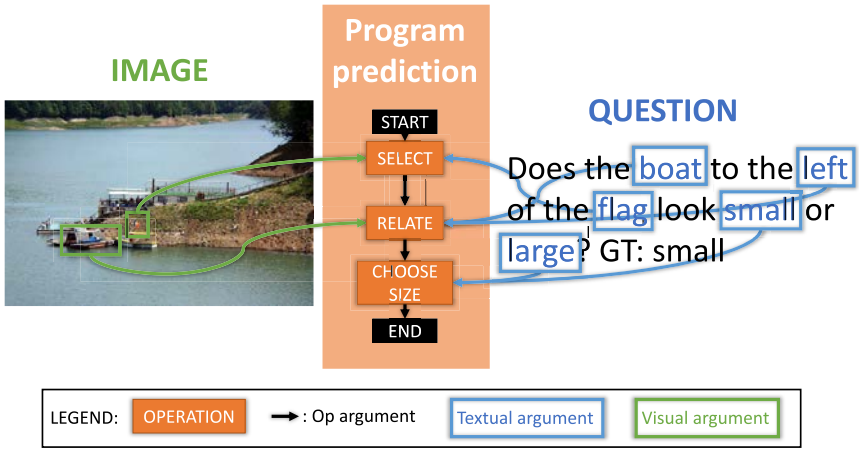


Figure 9.4 – When answering a question posed over an image, one needs to decompose the reasoning into multiple steps (*i.e. operations*), going further than the alignment between vision and language. In this illustration, the question can be answering by: ① localizing the *flag*; ② relating it to the *boat* on the *left* of it; and ③ identifying its size. Therefore, we propose to boost the *oracle transfer* by supervising the VQA model to predict the sequence of operations associated to the question-image pair.

result of the two operations checking the existence of a specific object in the image. This is denoted as $\boldsymbol{a}_{ij}^{q*} \in \{0,1\}$ where $\boldsymbol{a}_{ij}^{q*}=1$ means that operation i is associated with question word j as argument and, similarly, $\boldsymbol{a}_{ij}^{v*}=1$ indicating a visual argument and $\boldsymbol{a}_{ij}^{d*}=1$ an operation result argument.

We propose to apply the regularization on top of the VL-Transformer architecture (*cf.* Chapter 3), based on sequences of self- and cross-modality attention. For this purpose, we define a trainable module for program generation (*program decoder*), added to the output of the VL-Transformer model as shown in Figure 9.5 — an adaptation to other architectures would be straightforward.

PROGRAM DECODER In the lines of Chen et al. (2021), the program decoder has been designed in a coarse-to-fine fashion. It first generates ① a coarse sketch of the program consisting only of the operations, which are then ② refined by predicting textual and visual arguments and dependencies between operations.

COARSE: OPERATION ① This module only predicts the sequence of operations $\{\boldsymbol{o}_i\}_{i\in[0,n-1]}$ using a recurrent neural network variant (GRU) (Cho et al. 2014), whose initial hidden state is initialized with the y_{CLS} token embedding of the VQA transformer — the same embedding from which classically the final answer $\boldsymbol{y}$ is predicted, *cf.* Figure 9.5. Inference is stopped when the special `STOP` operation is predicted. At each GRU time step i, a new hidden state $\boldsymbol{h}_i$ is computed, from which the operation $\boldsymbol{o}_i$ is classified with a linear projection. It is supervised with a cross-entropy loss:

$$\mathcal{L}_{op} = \sum_i \mathcal{L}_{CE}(\boldsymbol{o}_i, \boldsymbol{o}_i^*)$$

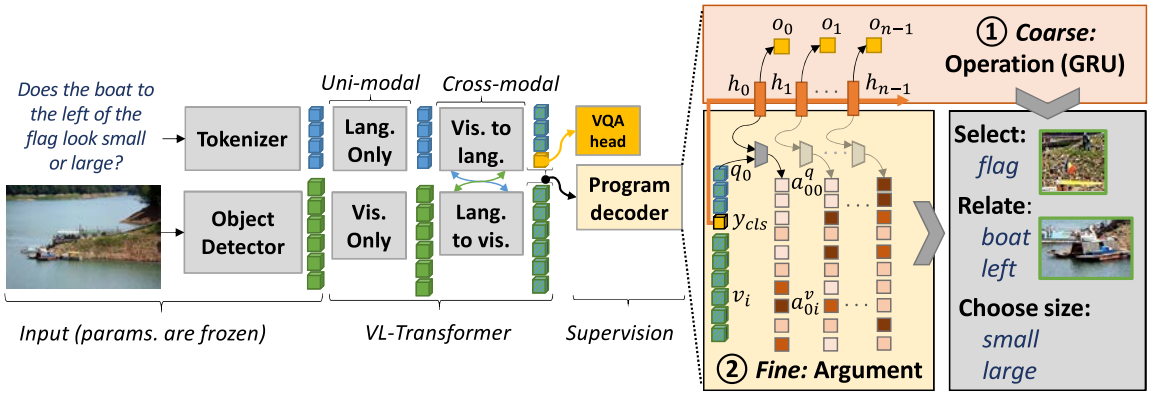


Figure 9.5 – A vision+language transformer with an attached program decoder. The decoder is fed with the VL-Transformer's penultimate embedding (just before the VQA classification head) and generates programs using a coarse-to-fine approach: ① a coarse program is generated using a GRU, consisting of a sequence of program operations embeddings $\{\boldsymbol{o}_i\}_{i\in[0,n-1]}$. ② It is then re-fined by predicting the visual $\boldsymbol{a}^v_{ij}$ and textual $\boldsymbol{a}^q_{ij}$ arguments using an affinity score between operation and input embeddings. Not shown: prediction of the operation's dependencies.

FINE: INPUT ARGUMENTS ② The coarse program is then refined by predicting the operations' arguments. We first deal with textual and visual arguments only. Affinity scores $\boldsymbol{a}^q_{ij}$ between each operation's hidden embedding $\boldsymbol{h}_i$ and each token embedding $\boldsymbol{q}_j$ are computed with a 2-layer feed-forward network from concatenated embeddings. They represent the probability of the word $\boldsymbol{q}_j$ to belong to the argument set of operation $\boldsymbol{o}_i$. Similar scores $\boldsymbol{a}^v_{ij}$ are computed for operations and visual objects. They are supervised with BCE losses:

$$\mathcal{L}_{qarg} = \sum_{ij}\mathcal{L}_{BCE}(\boldsymbol{a}^q_{ij}, \boldsymbol{a}^{q*}_{ij})$$
$$\mathcal{L}_{varg} = \sum_{ij}\mathcal{L}_{BCE}(\boldsymbol{a}^v_{ij}, \boldsymbol{a}^{v*}_{ij})$$

FINE: OP ARGUMENTS Next, the dependencies are predicted, *i.e.* arguments which correspond to results of other operations, and which structure the program into a tree. We deal with these arguments differently, and compute the set of dependency arguments for each operation o_i with another GRU, whose hidden state is initialized with the hidden state $\boldsymbol{h}_i$ of the operation. The argument index a^d_{ij} is a linear projection of the hidden state and supervised with BCE:

$$\mathcal{L}_{varg} = \sum_{ij}\mathcal{L}_{BCE}(\boldsymbol{a}^d_{ij}, \boldsymbol{a}^{d*}_{ij})$$

PROGRAM SUPERVISION The coarse-to-fine program decoder is trained with the four additional losses weighted by hyperparameters α, β, γ, δ.

$$\mathcal{L} = \underbrace{\mathcal{L}_{vqa}}_{\text{VQA}} + \underbrace{\alpha.\mathcal{L}_{op} + \beta.\mathcal{L}_{dep} + \gamma.\mathcal{L}_{qarg} + \delta.\mathcal{L}_{varg}}_{\text{Program supervision}}$$

GROUND TRUTH PROGRAMS We use ground truth information from the GQA dataset, whose questions have been automatically generated from real images. Each sample contains a program describing the operations and arguments required to derive the

answer for each question. However, the GT programs have been created for GT visual arguments (GT objects), which do not exactly match the visual input of an object detector used during training and inference (Anderson et al. 2018). We therefore construct a soft target, by computing intersection-over-union (IoU) between GT and detected objects.

GUIDED ORACLE TRANSFER Our method uses program supervision to regularize knowledge transfer from a visual oracle to noisy input, as introduced in the *oracle transfer* method. We perform the following steps:

1. Oracle pre-training on GT visual input on the GQA dataset, *including program supervision*;
2. (optionally) BERT-like pre-training on data from GQA *unbalanced*, *with program-supervision*;
3. Fine-tuning on the final VQA objective on the GQA dataset, *while keeping program supervision*.

9.3.2 *Experimental evaluation*

9.3.2.1 *Setup*

DATASET Our models are trained on the balanced GQA training set (∼1M question-answer pairs). However, LXMERT pretraining is done on the *unbalanced* training set (∼15M question-answer pairs). The latter contains more questions and programs, but the same number of images (∼100K images). Note that LXMERT (Tan et al. 2019) is originally pre-trained on a corpus gathering images and sentences from MSCOCO (Lin et al. 2014) and VisualGenome (Krishna et al. 2017). In this work, we only train on the GQA *unbalanced* set, with VisualGenome images. The maximum number of operations in one program is set to $N_{maxop} = 9$. The total number of operation's labels is $N_{op} = 212$. We evaluate on the GQA (Hudson et al. 2019b) and GQA-OOD (*cf.* Chapter 5) datasets.

ARCHITECTURE *VQA architecture*: we use the compact VL-Transformer introduced in Chapter 3. *Program decoder*: The hidden size is set to 128 (same as in the VL-Transformer). We use GeLU (Hendrycks et al. 2016) as non-linearity, along with layer norm (Ba et al. 2016). We use a one layer GRU (Cho et al. 2014) with hidden size equals to 128, to infer the operation's hidden embedding $\boldsymbol{h}_i$. It is followed by a two-layers MLP ($128 \rightarrow 64 \rightarrow N_{op}$, projecting $\boldsymbol{h}_i$ into a one-hot vector $\boldsymbol{o}_i$. Affinity scores $\boldsymbol{a}_{ij}^q$ between each operation's hidden embedding $\boldsymbol{h}_i$ and each token embedding $\boldsymbol{q}_j$ (or $\boldsymbol{v}_j$) are computed with a 2-layer feed-forward network ($256 \rightarrow 64 \rightarrow 1$) from concatenated embeddings. The op arguments are predicted from $\boldsymbol{h}_i$ using another one layer GRU with hidden size equals to 128, followed by a nonlinear projection ($128 \rightarrow N_{maxop}$). Hyperparameters are set to $\alpha = 1$, $\beta = 1$, $\gamma = 1$ and $\delta = 100$.

TRAINING DETAILS All models were trained with the Adam optimizer (Kingma et al. 2014), a learning rate of 10^{-4} with warm starting and learning rate decay. *pretraining:* performed during 20 epochs with a batch size of 320 (256 when using VinVL features).

Run	Model	#GPUs	# hours	Total number of runs
train	Oracle	1	30	≈ 5
train+test	ours 36 RCNN	1	9	≈ 100
train+test	ours 100 RCNN	2	10	≈ 5
train+test	ours VinVL	2	10	≈ 5
train+test	ours 36 RCNN + LXMERT pretrain	2	100	≈ 20
train+test	ours 36 RCNN + LXMERT finetune	1	4	≈ 50
train+test	ours VinVL + LXMERT pretrain	3	180	2
train+test	ours VinVL + LXMERT finetune	1	6	2

Table 9.4 – Training and execution time for one run. *Ours* corresponds to our *guided oracle transfer*. We also provide the approximated amount of runs done during this work (hyper parameters search, abblation, *etc*.)

	Model	Oracle transf.	Prog. sup.	GQA-OOD acc-tail	GQA-OOD acc-head	GQA test-dev	GQA binary*	GQA open*	GQA test-std	AUC† prog.
scratch	(a) Baseline			42.9	49.5	52.4	-	-	-	/
	(b) Oracle transfer	✓		48.2±0.3	54.6±1.1	57.0±0.3	74.5	42.1	57.3	/
	(c) Guided oracle transfer	✓	✓	**48.8**±0.1	**56.1**±0.3	**57.8**±0.2	**75.4**	**43.0**	**58.2**	97.1
+ lxmert	(d) Baseline			47.5	55.2	58.5	-	-	-	/
	(e) Oracle transfer	✓		47.1	54.8	58.4	77.1	42.6	58.8	/
	(f) **Guided oracle transfer**	✓	✓	**48.0**±0.6	**56.6**±0.6	**59.3**±0.3	**77.3**	**44.1**	**59.7**	96.4

Table 9.5 – *Guided oracle transfer*: Impact of program supervision on *Oracle transfer* for vision-language transformers. LXMERT (Tan et al. 2019) pre-training is done on the GQA unbalanced training set. We report scores on GQA (Hudson et al. 2019b) (*test-dev* and *test-std*) and GQA-OOD (*test*). * binary and open scores are computed on the test-std; † we evaluate visual argument prediction by computing AUC@0.66 on GQA-val.

All pretraining losses are added from the beginning, including the VQA one. *fine-tuning:* on the GQA *balanced* set during 4 epochs, with a batch size of 32 and a learning rate equal to 10^{-5}. Hyperparameters are selected either on the testdev (for GQA) or validation (for GQA-OOD) sets. When specified (with $\pm$) we provide the average accuracy and standard deviation computed on three runs with different random seeds.

COMPUTING RESOURCES & CO2 EMISSION Training and evaluation has been performed on several compute infrastructures, which include an Nvidia DGX-A100 with 8× A100 GPUs and a cluster with P100 and RTX 2080 GPUs. After design and development, the final training and evaluation runs have been performed on Geforce RTX 2080 GPUs. We provide an estimate for the amount of compute in Table 9.4 — the number of GPUs and approximate execution times for different models and experimental settings (train, validation, and test). The RTX infrastructure has a carbon efficiency of 0.035

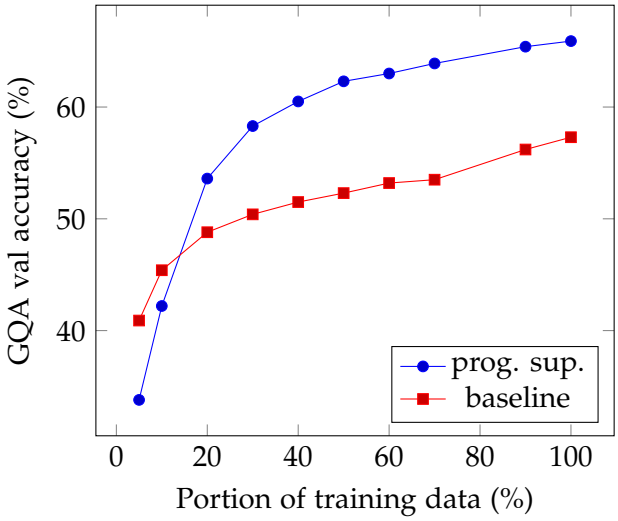


Figure 9.6 – Program supervision leads to a decreased sample complexity. We vary the amount of training data from 5% to 100%, comparing overall accuracy obtained with and without program supervision. We observe that adding program supervision allows to reach an accuracy similar to the baseline while using fewer data. In this setup, we do not use *oracle transfer* neither LXMERT pretraining.

9.3.2.2 *Results*

PROGRAM SUPERVISION IMPROVES VISUAL REASONING Table 9.5 reports the effectiveness of program prediction when combined with oracle and BERT-like pretraining on the GQA dataset, and corroborates the results found in the theoretical analysis. In addition, when using both program supervision and LXMERT (Tan et al. 2019) but without oracle transfer, we achieve an accuracy of 58.8 on the *testdev* set of GQA. This is lower than oracle transfer's accuracy, demonstrating the complementarity of the two methods. We note that the majority of the gain is achieved on the more challenging *open* questions. In addition, results on GQA-OOD (*acc-tail* and *acc-head*) suggest that the gains are obtained in, both, out- and in-distribution settings. However, as already observed in Chapter 9, LXMERT pre-training tends to decrease the *acc-tail* gains brought by oracle transfer plus program supervision. We evaluate the program prediction performance by measuring the area under the ROC curve (AUC) on the visual argument prediction with an IoU threshold of $\frac{2}{3}$=0.66. Models (c) and (e) achieve, respectively, 97.1 and 96.4 AUC scores, demonstrating the effectiveness of the program decoder.

DECREASED SAMPLE COMPLEXITY In Figure 9.6, we verify that program supervision does indeed reduce the sample complexity as demonstrated in Chapter 8. For this purpose, we measure the accuracy on GQA (validation set) while reducing the amount of data used during the training. We observe that adding program supervision allows to reach an accuracy similar to the baseline while using less data. Thus, for a given target accuracy (*e.g.* $> 55\%$), the number of required training samples is lower when using our program supervision method (30% vs. 100% of the data).

Ablations	Oracle transf.	GQA-OOD acc-tail (val.)	GQA val.
(1) VQA only		46.9	62.2
(2) Coarse only		46.5	62.5
(3) Coarse + dep.		46.8	62.8
(4) Full w/o v.arg		47.3	63.7
(5) Full		**49.9**	**66.2**
(6) Random prog.		45.7	61.4
(7) No prog	✓	50.0	66.4
(8) Uni-modal	✓	49.9	66.5
(9) Cross-modal	✓	**50.4**	**67.4**

Table 9.6 – Ablation study. (1-5): we analyze different types of program supervision, and show that visual arguments are the key. (6): we compare with the *random prog* baseline, where we randomly replace the ground truth program with a program picked from another question. (7-9): we study the impact of the program supervision position, after uni-modal layers or after cross-modal layers. The supervision is more efficient when used after cross-modal interactions. No LXMERT/BERT pre-training.

VISUAL ARGUMENTS ARE THE KEY We study the impact of different types of program supervision in Table 9.6 (1-5). We can see the importance of supervising arguments, in (4) and (5). The supervision of visual arguments (5) contributes most to the gain in performance, again corroborating that visual uncertainty is the main bottleneck for reasoning on the GQA dataset. In addition, as a sanity check, we show in (6) that supervising with random programs does not improve the baseline.

PROGRAM SUPERVISION ENHANCES CROSS-MODAL INTERACTIONS In Table 9.6 (7-9), we study how the inputs of the program prediction module influence the VQA accuracy. In particular, we test two settings: (8) *uni-modal*, where the programs are predicted from the vision and language embeddings right after the uni-modal layers (language and vision only in Figure 9.5); and (9) *cross-modal*, where the programs are predicted after the cross-modal layers. We observe that, contrary to the latter, the former does not improve the baseline ((8) vs (7) in Table 9.6). This highlights the fact that the program supervision mainly impacts the operations in the cross modal layers, where the most complex reasoning operations are performed.

PROGRAM SUPERVISION ALLOWS TAKING ADVANTAGE OF BETTER VISION We analyze the impact of using our method with a better input image representation. Increasing the number of objects from 36 to 100 per image ((g) and (h) in Table 9.7), allows to further increase the gains brought by our method. On the contrary, the score of the baseline model remains unchanged, showing that the program supervision allows taking advantage of a bigger number of object proposals. Similarly, replacing the faster-RCNN features by the more recent and more accurate VinVL ones ((i-l) in Table 9.7) results in better performances.

Model	Visual features	Oracle transf.	Prog. sup.	GQA test-dev	GQA binary*	GQA open*	GQA test-std
(g) Oracle transfer	100 RCNN	✓		57.0±0.4	-	-	-
(h) Guided oracle transfer		✓	✓	**58.2**±0.1	-	-	-
(i) Oracle transfer	VinVL	✓		59.6±0.1	-	-	-
(j) Guided oracle transfer		✓	✓	60.9±0.2	-	-	-
(k) Oracle transfer +lxmert		✓		61.4	79.6	47.5	62.5
(l) Guided oracle transfer +lxmert		✓	✓	**61.8**	**80.1**	**48.0**	**63.0**

Table 9.7 – Impact of improved visual inputs while using program supervision on Vision-Language Transformers. Scores on GQA (Hudson et al. 2019b). *binary/open are computed on test-std. VinVL (Zhang et al. 2021) RCNN (Ren et al. 2015)

Method	Visual feats.	Additional supervision	Training data (M) Img	Training data (M) Sent	GQA-OOD acc-tail	GQA-OOD acc-head	GQA bin.	GQA open	GQA all
BAN4 (Kim et al. 2018)	RCNN	-	≈ 0.1	≈1	47.2	51.9	76.0	40.4	57.1
MCAN (Yu et al. 2019)	RCNN	-	≈ 0.1	≈1	46.5	53.4	75.9	42.2	58.0
Oracle transfer	RCNN	-	≈0.18	≈1	48.3	55.5	75.2	44.1	58.7
MMN (Chen et al. 2021)	RCNN	Program	≈0.1	≈15	48.0	55.5	78.9	44.9	60.8
LXMERT (Tan et al. 2019)	RCNN	-	≈0.18	≈9	**49.8**	57.7	77.8	45.0	60.3
Guided oracle transfer	VinVL	Program	≈0.1	≈15	49.1	**59.7**	80.1	48.0	63.0
NSM (Hudson et al. 2019a)	SG	Scene graph	≈0.1	≈1	-	-	78.9	**49.3**	63.2
OSCAR+VinVL (Zhang et al. 2021)	VinVL	-	≈5.7	≈9	-	-	**82.3**	48.8	**64.7**

Table 9.8 – Comparison with the state of the art on the GQA (Hudson et al. 2019b) (*test-std*) and GQA-OOD (Kervadec et al. 2021a) (*test*) sets. For a fair comparison, we provide information about the required training data and supervision. RCNN (Anderson et al. 2018), SG (Hudson et al. 2019a), VinVL (Zhang et al. 2021)

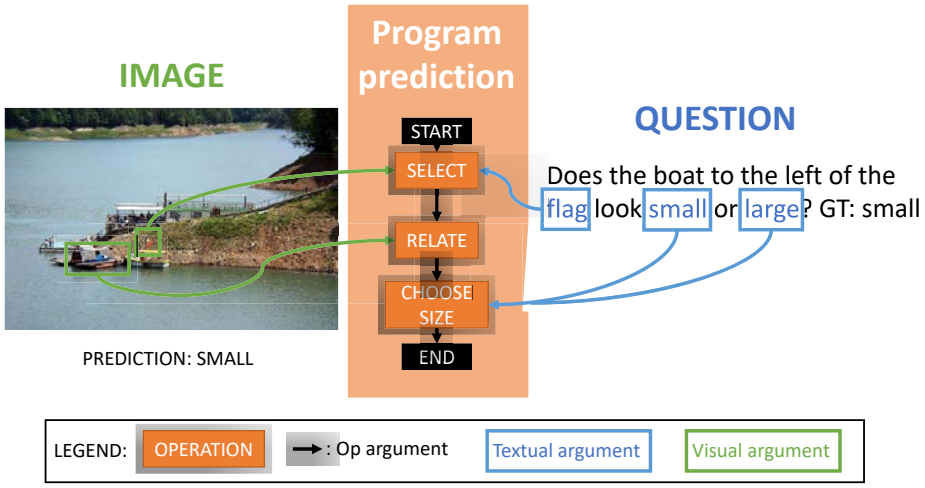


Figure 9.7 – Example of program prediction. The question is: *"Does the boat to the left of the flag looks small or large?"*. Our model (ours+lxmert with VinVL) correctly answers *"small"*.

COMPARISON WITH SOTA We report in Table 9.8 the results obtained by our approach compared to the current SOTA on the GQA and GQA-OOD datasets. In order to ensure a fair comparison, we also provide, for each method, information regarding the amount of data (images and sentences) used during training. As shown in Table 9.8, our approach compares favorably with SOTA, since it obtains the second-best accuracy (with a 0.2 points gap) on the GQA test-std set among the approaches which not use extra training data. The results also remain competitive when comparing to the OSCAR+VinVL (Zhang et al. 2021), while being trained with 50 times fewer images. On GQA-OOD, our approach obtains the second best *acc-tail* score (and the best *acc-head* one) with a much less complex architecture than current SOTA (26M vs 212M trainable parameters compared to LXMERT (Tan et al. 2019)).

VISUALIZATION OF PREDICTIONS We provide examples of program prediction in Figure 9.7 and Figure 9.8. In Figure 9.7, the question is *'does the boat to the left of the flag look small or large?'*. The program decoder successfully infers the correct program. It first predicts the coarse operations – `select, relate, choose size` –, then adds the arguments taken from the image or the question – boat, flag, small, large –. Finally, the VQA model predicts the correct answer *'small'*. In Figure 9.8, the question is *'who is wearing goggles?'*. Similarly to the first example, the program decoder generates coarse operations – `select, relate, query name` – and visual/textual arguments – woman, who, goggles, wearing–. In these two examples, the decoder correctly predicts that the programs are chains of operations (special case of a tree). At contrary, a question like *"are there nuts or vegetables?"* is a not a chain because of the presence of `exist` and `or` operations.

9.4 CONCLUSION

Drawing conclusions from analysis conducted in Part III, we have shown that reasoning patterns can be partially transferred from oracle models to SOTA VQA models based on

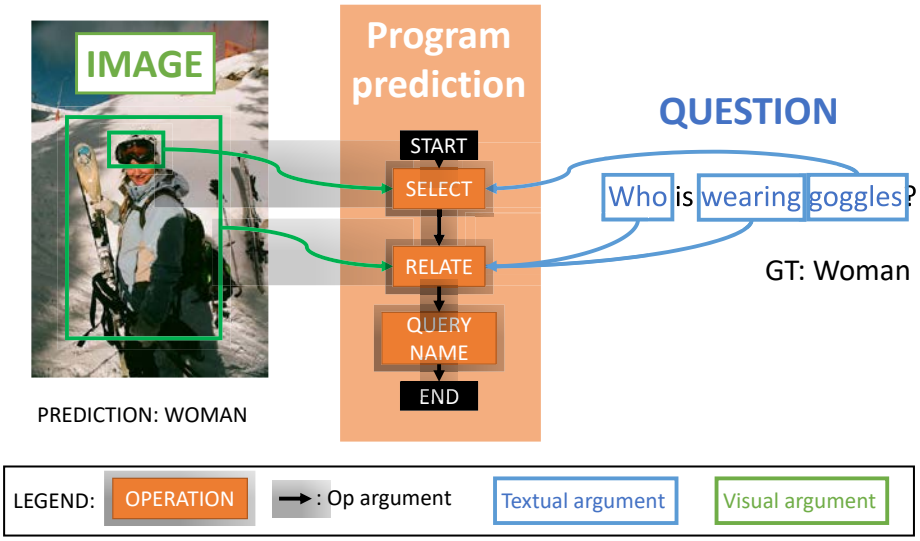


Figure 9.8 – Example of program prediction. The question is: *"Who is wearing goggles?"*. Our model (ours+lxmert with VinVL) correctly answers *"woman"*.

Transformers and BERT-like pre-training. The accuracy gained from the transfer is particularly high on questions with rare GT answers, suggesting that the knowledge transferred is related to reasoning, as opposed to bias exploitation. We have also demonstrated that it is possible to improve this knowledge transfer by providing an additional supervision of program annotations. Furthermore, our experiments are aligned with theoretical and experimental results found in Chapter 8, demonstrating that program supervision can decrease sample complexity. The proposed method relies on the availability of reasoning program annotations, which are costly to annotate, especially when dealing with human-generated questions. Recent work has already managed to gather such kind of annotations Das et al. 2016. The next step will be to extend the method to configurations where the program annotation is rare or incomplete.

*
* *

Chapter 10

GENERAL CONCLUSION

10.1 SUMMARY OF CONTRIBUTIONS

This thesis focuses on the question of bias vs reasoning in VQA. Part II and Part III provide a diagnostic on the effect of shortcut learning on VQA. Drawing a conclusion from it, Part IV proposes two complementary methods, improving the model's predictions while mitigating the effect of biases. All in all, our contributions can be summarized as follows:

EVALUATE (PART II) We conduct a comprehensive review of existing evaluation methods for VQA. We show that they struggle to correctly evaluate the reasoning ability, so we propose our own benchmark called GQA-OOD. It consists in providing a multidimensional evaluation, allowing to measure the OOD performance – which we argue to be related to the reasoning ability – by controlling the rarity of the used test examples (in-distribution vs out-of-distribution). Thus, GQA-OOD has been designed to address most of the limitations found in others benchmarks: evaluate both in- and out-of-distribution accuracies at the same time, validate in OOD setting, maintain natural biases. Thereby, we experimentally demonstrate that all the VQA models that we have tested are brittle to OOD evaluation. This suggests that they have learned to rely on shortcuts instead of reasoning. Furthermore, our results show that even methods specifically devised to mitigate the influence of biases fail in our setup. GQA-OOD is publicly available: we encourage researchers to evaluate their models on it, or to extend our methodology to other tasks.

ANALYZE (PART III) We complement the quantitative results provided by the evaluation part (Part II) with qualitative observations. For this purpose, we conduct an instance-based visualization of the attention learned by a VL-Transformer using VISQA (developed in collaboration with Théo Jaunet). This analysis highlights interesting insights about the type of reasoning which is performed by the learned model. In particular, we observe potential bottlenecks for learning to reason, such as the uncertainty in the visual part (*e.g.* useful objects are not correctly detected), the difficulty to precisely align visual regions with question words, or other language biases (*e.g.* with logical operators). Then, in a broader dataset-level study of the learned attention maps, we analyze the emergence of reasoning patterns in the same VL-Transformer. We demonstrate that the ability to relate attention to the task at hand (*i.e.* the ability to reason) is present when the training

conditions are favorable enough, *e.g.* when the uncertainty in the visual part is reduced (visual oracle), but not in the standard setting.

IMPROVE (PART IV) In the last part, we design two complementary approaches for improving reasoning in VQA. The first one focuses on training supervision. In particular, we propose to add a proxy loss for reasoning (*e.g.* a weak supervision of the fine-grained word-object alignment). In an experimental study, we show that this additional supervision helps to improve the visual reasoning performance. We complement those experimental results, by providing theoretical clues (based on PAC-learning) demonstrating that reasoning supervision reduces the sample complexity, and eases the learning of reasoning. The second method directly takes inspiration from the results obtains in the analysis part (Part III). We propose to transfer the reasoning patterns learned when the training conditions are favorable to the standard settings having uncertainty in the input. We show that this transfer is feasible and does improve the VQA performances in both in- and out-of-distribution settings. Furthermore, we combine the transfer of reasoning patterns with the reasoning supervision and experimentally demonstrate that the latter is a catalyst for the former.

10.2 PERSPECTIVES FOR FUTURE WORK

The work conducted in this thesis opens a wide range of exciting perspectives and challenges that we have listed below. It includes the conception of new evaluation process for ML, the design of methods to mitigate shortcut learning, and the exploration of reasoning beyond DL. There are also numerous other broader issues, not mentioned enough in this thesis, which are primordial for the ML field. Thus, we can cite the importance of studying and preventing the (potentially negative) societal impact caused by biases in DL-based technologies, or the urgent need to conciliate DL usage with concerns raised by climate change.

10.2.1 *Evaluation in ML*

In Part II, we propose a new method for evaluating VQA models. An interesting perspective would be to adapt our method to other tasks, going beyond VQA or vision-and-language understanding. Besides, we also think that we have to keep putting effort in improving the way we evaluate and compare ML approaches.

REAL-WORLD SCENARIOS In the real world, it can be difficult to disentangle reasoning from perception. Therefore, while synthetic datasets – such as CLEVR (Johnson et al. 2017) for VQA – are useful (and necessary) tools for diagnosing weaknesses and strengths in models, we also have to work on real-world scenarios. A large part of the work in this thesis has been conducted on the GQA database (Hudson et al. 2019b). As explained in Chapter 4, GQA is the best suited for evaluating reasoning capabilities in VQA. At the same time, it is semisynthetic, because it contains both real images and synthetic questions. While it was a necessary step, in the future, we will have to validate our

approaches on a more realistic setup. For instance, a dataset such as VizWiz (Gurari et al. 2018) is a good candidate for real-world evaluation. Such adaptation to real-world applications brings new challenges, such as a greater diversity of concept, or a higher uncertainty in annotation.

DYNAMIC BENCHMARKS We think that a good evaluation cannot be static. The quick evolution of DL, combined with the SOTA race, can lead to a kind of overfitting, where models are unconsciously selected on the test set. One possible solution could be to design *dynamic benchmarks*. Therefore, Gorman et al. (2019) analyzed published part-of-speech taggers, and propose to use *randomly generated* splits instead of the static *standard splits*. Recently, two VQA benchmarks – namely adVQA (Sheng et al. 2021) and AVVQA (Li et al. 2021) – make use of a human adversarial evaluation in order to update the standard test set with questions fooling a SOTA model. We could imagine doing such *test update* on a regular basis, to obtain a kind of *dynamic evaluation*, where benchmarks update across time.

A BETTER SCIENTIFIC METHOD Improving evaluation and diagnosing in ML goes hand in hand with a better scientific method. In this thesis, we tried (even if it is far from being perfect) to carefully evaluate the significance of our experimental results before drawing conclusions. It includes the use of adequate baselines and ablation studies, but also a statistical measure of the results' significance (here, we use basic statistics, namely the average plus std across random seed). On this subject, Picard (2021) shows how important can be the impact of the random seed selection on the final performance, suggesting that the statistical significance of experimental results do have to be carefully handled. However, there is still a large room for improvement. Thus, to avoid the negative results depicted in Chapter 4 (where some VQA methods are directly validated on the test set!), it appears to be necessary to re-think our ML practices. A good starting point is the discussion led by Forde et al. (2019), taking inspiration from physics to provide good practices, which could positively enhance the scientific method in ML.

10.2.2 *Mitigating shortcut learning*

This thesis puts a lot of efforts on diagnosing the cause and effect of shortcut learning in VQA. Designing new methods for mitigating these unwanted effects is an obvious perspective, which goes beyond the scope of VQA.

IMPROVE THE VISION PART We have seen that uncertainty in the vision part is a crucial factor leading to shortcuts in vision-and-language understanding. Therefore, a fruitful perspective of work would be to improve the vision part. We can think about designing object detectors with a better precision, in order to reduce the visual uncertainty as done in Zhang et al. (2021). Alternatively, as already started by Jiang et al. (2020), conducting analyzes on which image representation type – namely, grid-level, object-level or anything else – is the best suited for visual reasoning is also essential. Finally, vision-and-language understanding requires a strong alignment between vision and language. Then, it could be interesting to address this issue as early as possible in the pipeline, and jointly learn

vision and language features, in the same vain as in Ramesh et al. (2021) or Radford et al. (2021).

DESIGN BIAS-AGNOSTIC METHODS Many approaches for mitigating biases during learning have been proposed. However, as shown in Chapter 5 (and confirmed in Dancette et al. (2021)), most of them provide limited improvement. We think that progress has to be made on this topic. Promising approaches includes multiple domain training (Rame et al. 2021), training collection of models while favoring diversity (Teney et al. 2021), or combining ML with causal approaches *e.g.* using counterfactual examples (Teney et al. 2020a).

10.2.3 *Explore reasoning beyond DL*

It appears, from our study, that many of the obstacles preventing from learning to reason are intrinsic to DL. In particular, we think of shortcut learning (Geirhos et al. 2020) and simplicity bias (Shah et al. 2020). In that context, it would be interesting to explore reasoning beyond DL, taking inspiration from other domains, in a cross-disciplinary fashion.

EMBODIED LEARNING In the standard DL training, a neural net is optimized, through iterative gradient descent on data samples, to minimize an objective loss aligned with the task to be accomplished. In that settings, the neural net has only access to *i.i.d.* samples in a *read only* fashion. However, as demonstrated in the infamous [1] experiment conducted by Held et al. (1963), the combination of perception with interaction (through sensory feedback) is essential for the development of the mammal brain. Thereby, it seems that the ability to *interact* with its environment is an essential property for learning to reason. This motivates methods for adapting the *read only* DL training to a setup where an agent has the possibility to interact with its environment. This is the objective of embodied learning, which have been notably used for VQA in Das et al. (2018).

CAUSAL REPRESENTATION LEARNING In a similar vain, graphical causality (Pearl et al. 2000), seeks to overcome ML issues by leveraging the notion of *intervention* in the data. As already seen in Chapter 2, "*causality*" is a property of "*reasoning*". In that context, it seems relevant to combine methods from both ML and graphical causality, as proposed by Schölkopf et al. (2021). It is worth noticing that some works already introduces causality in DL, *e.g.* in VQA (Teney et al. 2020a; Agarwal et al. 2020) or in counterfactual learning of physics (Baradel et al. 2019).

COGNITIVE SCIENCES Another perspective would be to develop new ways for learning to reason by taking inspiration from cognitive sciences. For instance, Lazaridou et al. (2017) make use of game theory and language evolution in order propose to analyze

1. In a nutshell, the experiment consisted in putting two kittens in a carousel. The first one can see and move. The second one can also see, but is not free to move. Its movements are mechanically linked with the first kitten, such that it does not have any control on them. It turns out that the kitten which cannot decide where it goes does not develop normally (Held et al. 1963).

the emergence of language in multiagent (neural agents) referential games. In the same context, Chaabouni et al. (2020) study whether such emergent languages have the faculty of "*compositionality*" and "*generalization*", two properties of "*reasoning*". *Would it be possible to do the same and study the emergence of visual reasoning?* Vani et al. (2021) already try to tackle the question, and propose to use iterated learning on a synthetic VQA task. We think that these cross-disciplinary works, borrowing results from cognitive sciences, will play an important role in the development of new neural models devised to reason.

* * *

Appendix A

PROOFS: SAMPLE COMPLEXITY OF REASONING SUPERVISION

A.1 PROOF OF THEOREM 8.3.3

In the lines of Arora et al. (2019), we first define the case for a single component $\boldsymbol{y}^{(i)}$ of the vector $\boldsymbol{y}$ and define the following Corollary:

COROLLARY A.0.1 (SAMPLE COMPLEXITY FOR MULTI-MODE REASONING FUNCTIONS WITH A SINGLE SCALAR COMPONENT). *Let $\mathcal{A}$ be an overparametrized and randomly initialized two-layer MLP trained with gradient descent for a sufficient number of iterations. Suppose $g : \mathbb{R}^d \to \mathbb{R}^m$ with $g(x) = \sum_r \sum_j (\gamma_r^T \boldsymbol{x}) \alpha_{r,j} (\beta_{r,j}^T \boldsymbol{x})^{p_{r,j}}$ where $\gamma_r \in \mathbb{R}^d$, $\beta_{r,j} \in \mathbb{R}^d$, $\alpha_{r,j} \in \mathbb{R}$, and $p_{r,j} = 1$ or $p_{r,j} = 2l, l \in \mathbb{N}_+$. The sample complexity $\mathcal{C}_{\mathcal{A}}(g, \epsilon, \delta)$ is:*

$$\mathcal{C}_{\mathcal{A}}(g, \epsilon_0, \delta_0) = O\left(\frac{\sum_r \sum_j \pi p_{r,j} |\alpha| \cdot ||\gamma_r||_2 \cdot ||\beta_{r,j}||_2^{p_{r,j}} + \log(\frac{1}{\delta_0})}{\epsilon_0^2} \right),$$

PROOF OF COROLLARY A.0.1 Using Theorem 5.1 from Arora et al. (2019), we know that sums of learnable functions are learnable, and can thus focus on a single term:

$$y = g(\boldsymbol{x}) = \alpha (\gamma^T \boldsymbol{x}) (\beta^T \boldsymbol{x})^p \tag{A.1}$$

where we dropped indices r and j and the superscript (i) for convenience. We proceed in the lines of the proof of Theorem 5.1 in Arora et al. (2019). Given a set of i.i.d data samples $S = \{(\boldsymbol{x}_s, y_s)\}_{s=1}^n = (\boldsymbol{X}, \boldsymbol{y})$ from the underlying function $g(x)$, let $\boldsymbol{w}$ be the weights of the first layer of a two-layers network with ReLu activations; let $\boldsymbol{H}^\infty \in \mathbb{R}^{n,n}$ be a Gram matrix defined as follows, with elements:

$$\boldsymbol{H}_{ij}^\infty = \mathbb{E}_{\boldsymbol{w} \sim \mathcal{N}(0,1)} \left[\boldsymbol{x}_i^T \boldsymbol{x}_j \mathbb{I}\{\boldsymbol{w}^t \boldsymbol{x}_i \geq 0, \boldsymbol{w}^t \boldsymbol{x}_i \geq 0\} \right].$$

To provide bounds on the sample complexity of $g(x)$, using Theorem 5.1 of Arora et al. (2019), it suffices to show that the following bound holds:

$$\sqrt{\boldsymbol{y}^T (\boldsymbol{H}^\infty)^{-1} \boldsymbol{y}} < M_g \tag{A.2}$$

for a bound M_g independent of the number of samples n.

We first introduce some notation. For matrices $\boldsymbol{A} = [\boldsymbol{a}_1, ..., \boldsymbol{a}_{n_3}] \in \mathbb{R}^{n_1 \times n_3}$ and $\boldsymbol{B} = [\boldsymbol{b}_1, ..., \boldsymbol{b}_{n_3}] \in \mathbb{R}^{n_2 \times n_3}$, the *Khatri-Rao* product is defined as $\boldsymbol{A} \odot \boldsymbol{B} = [\boldsymbol{a}_1 \otimes \boldsymbol{b}_1, \boldsymbol{a}_2 \otimes \boldsymbol{b}_2, ..., \boldsymbol{a}_{n_3} \otimes \boldsymbol{b}_{n_3}]$. Let $\circ$ be the *Haddamard* product (element wise multiplication) of two matrices. We also denote the corresponding powers by $\boldsymbol{A}^{\otimes l}, \boldsymbol{A}^{\odot l}, \boldsymbol{A}^{\circ l}$. We denote by $\boldsymbol{A}^{\dagger} = (\boldsymbol{A}^T \boldsymbol{A})^{-1} \boldsymbol{A}^T$ the *Moore-Penrose* pseudo-inverse, and by $\boldsymbol{P}_{\boldsymbol{A}} = \boldsymbol{A}^{\frac{1}{2}} \boldsymbol{A}^{\dagger} \boldsymbol{A}^{\frac{1}{2}}$ the projection matrix for the subspace spanned by $\boldsymbol{A}$. From the proof of Theorem 5.1 in (Arora et al. 2019), we also know that:

$$\boldsymbol{H}^{\infty} \succeq \frac{\boldsymbol{K}^{\circ 2l}}{2\pi(2l-1)^2}$$

where $\boldsymbol{K} = \boldsymbol{X}^T \boldsymbol{X}$, and $\boldsymbol{X}$ is the data matrix of all row vectors $\boldsymbol{x}_i$.

Let us consider the case of $p = 1$. Reformulating Equation A.1, we get:

$$\begin{aligned} y = g(\boldsymbol{x}) &= \alpha(\gamma^T \boldsymbol{x})(\beta^T \boldsymbol{x}) && \text{(A.3)} \\ &= \alpha(\boldsymbol{x}^T \gamma)(\boldsymbol{x}^T \beta) && \text{(A.4)} \\ &= \alpha(\boldsymbol{x} \otimes \boldsymbol{x})^T (\gamma \otimes \beta) && \text{(A.5)} \end{aligned}$$

Now, taking the full set of input vectors $\boldsymbol{x}_i$ arranged into the full data matrix $\boldsymbol{X}$, we can perform similar algebraic operations to get

$$\begin{aligned} \boldsymbol{y} = g(\boldsymbol{X}) &= \alpha(\boldsymbol{X}^T \gamma) \circ (\boldsymbol{X}^T \beta) && \text{(A.6)} \\ &= \alpha(\boldsymbol{X}^{\odot 2})^T (\gamma \otimes \beta) && \text{(A.7)} \end{aligned}$$

Plugging Equation A.6 and Equation A.7 into Equation A.2, we need to show that the following expression is smaller than a constant M_g:

$$\begin{aligned} & \alpha^2((\boldsymbol{X}^T \gamma) \circ (\boldsymbol{X}^T \beta))^T (\boldsymbol{H}^{\infty})^{-1} (\boldsymbol{X}^{\odot 2})^T (\gamma \otimes \beta) && \text{(A.8)} \\ =& \alpha^2((\boldsymbol{X}^{\odot 2})^T (\gamma \otimes \beta))^T (\boldsymbol{H}^{\infty})^{-1} (\boldsymbol{X}^{\odot 2})^T (\gamma \otimes \beta) && \text{(A.9)} \\ =& \alpha^2 (\gamma \otimes \beta)^T (\boldsymbol{X}^{\odot 2}) (\boldsymbol{H}^{\infty})^{-1} (\boldsymbol{X}^{\odot 2})^T (\gamma \otimes \beta) && \text{(A.10)} \\ \leq& 2\pi\alpha^2 (\gamma \otimes \beta)^T (\boldsymbol{X}^{\odot 2}) (\boldsymbol{K}^{\circ 2})^{\dagger} (\boldsymbol{X}^{\odot 2})^T (\gamma \otimes \beta) && \text{(A.11)} \\ =& 2\pi\alpha^2 (\gamma \otimes \beta)^T \boldsymbol{P}_{\boldsymbol{X}^{\odot 2}(\boldsymbol{X}^{\odot 2})^T} (\gamma \otimes \beta) && \text{(A.12)} \\ \leq& 2\pi\alpha^2 ||(\gamma \otimes \beta)||_2^2 && \text{(A.13)} \\ =& 2\pi\alpha^2 ||\gamma||_2^2 \cdot ||\beta||_2^2 && \text{(A.14)} \end{aligned}$$

where we made use of $||a \otimes b||_2^2 = ||a||_2^2 ||b||_2^2$ for two vectors a and b and an integer n. This finishes the proof for the case $p = 1$.

Let us consider the case of $p = 2l+1$. Reformulating Equation A.1, we get:

$$\begin{aligned} \boldsymbol{y} = g(\boldsymbol{X}) &= \alpha(\boldsymbol{X}^T \gamma) \circ (\boldsymbol{X}^T \beta)^p && \text{(A.15)} \\ &= \alpha(\boldsymbol{X}^{\odot 2l})^T (\gamma \otimes \beta^{\otimes(2l+1)}) && \text{(A.16)} \end{aligned}$$

Plugging Equation A.16) into Equation A.2, we again need to show that the following expression is smaller than a constant M_g:

$$\alpha^2((\boldsymbol{X}^{\odot 2l})^T(\gamma\otimes\beta^{\otimes(2l+1)}))^T \tag{A.17}$$
$$(\boldsymbol{H}^{\infty})^{-1}(\boldsymbol{X}^{\odot 2l})^T(\gamma\otimes\beta^{\otimes(2l+1)}) \tag{A.18}$$
$$=\alpha^2(\gamma\otimes\beta^{\otimes(2l+1)})^T \tag{A.19}$$
$$(\boldsymbol{X}^{\odot 2l})(\boldsymbol{H}^{\infty})^{-1}(\boldsymbol{X}^{\odot 2l})^T(\gamma\otimes\beta^{\otimes(2l+1)}) \tag{A.20}$$
$$\leq 2\pi(2l-1)^2\alpha^2(\gamma\otimes\beta^{\otimes(2l+1)})^T \tag{A.21}$$
$$(\boldsymbol{X}^{\odot 2l})(\boldsymbol{K}^{\circ 2})^{\dagger}(\boldsymbol{X}^{\odot 2l})^T(\gamma\otimes\beta^{\otimes(2l+1)}) \tag{A.22}$$
$$=2\pi(2l-1)^2\alpha^2(\gamma\otimes\beta^{\otimes(2l+1)})^T \tag{A.23}$$
$$\boldsymbol{P}_{\boldsymbol{X}^{\odot 2l}(\boldsymbol{X}^{\odot 2l})^T}(\gamma\otimes\beta^{\otimes(2l+1)}) \tag{A.24}$$
$$\leq 2\pi(2l-1)^2\alpha^2||(\gamma\otimes\beta^{\otimes(2l+1)})||_2^2 \tag{A.25}$$
$$\leq 2\pi p^2\alpha^2||(\gamma\otimes\beta^{\otimes(2l+1)})||_2^2 \tag{A.26}$$
$$=2\pi p^2\alpha^2||\gamma||_2^2\cdot||\beta||_2^{2p} \tag{A.27}$$

where we made use of $||a\otimes b||_2^2 = ||a||_2^2||b||_2^2$ and therefore $||a^{\otimes n}||_2^2 = ||a||_2^{2n}$ for two vectors a and b and an integer n. This finishes the proof for the case $p = 2l+1$.

THE CASE OF VECTORIAL OUTPUTS In the lines of (Xu et al. 2020), we consider each component of the output vector independent and apply a union bound to Corollary A.0.1. If the individual components $\boldsymbol{y}^{(i)}$ fail to learn with probability δ_0, then the full output of dimension m fails with probability $m\delta_0$ and with an error of at most $m\epsilon_0$. A change of variables from (ϵ_0, δ_0) to (ϵ, δ) gives a complexity for the model with vectorial output of

$$\mathcal{C}_{\mathcal{A}}(g,\epsilon,\delta) = O\left(\frac{\max_i \sum_r \sum_j \pi p_{r,j}^{(i)}|\alpha|\cdot||\gamma||_2\cdot||\beta_{r,j}||_2^{p_{r,j}^{(i)}} + \log(m/\delta)}{(\epsilon/m)^2}\right),$$

This ends the proof of Theorem 4.2.

A.2 PROOF OF THE INEQUALITY IN EQUATION 8.18

Let us denote by $p(x)$ the density of normal distribution. And to make the notation more succinct and to avoid confusion between different usages of superscripts, in this proof we will change γ_r^i to γ_i, *i.e.* the ith component of the vector γ, not to be confused with γ_r, a vector corresponding to the embedding of the rth reasoning mode. Then:

$$\mathbb{E}_{\gamma_i\sim N(0,1)}||\gamma||_2\cdot||\beta||_2^p \tag{A.28}$$
$$=||\beta||_2^p\mathbb{E}_{\gamma_i\sim N(0,1)}\left(\sum_i \gamma_i^2\right)^{\frac{1}{2}} \tag{A.29}$$

We now perform a change of variables and introduce a new random variable:

$$z = \sum_i \gamma_i^2 \tag{A.30}$$

Since each individual γ_i is distributed normal, z is distributed according to a χ^2 distribution with m degrees of freedom, and we get:

$$\mathbb{E}_{\gamma_i \sim N(0,1)} ||\gamma||_2 \cdot ||\beta||_2^p \tag{A.31}$$

$$= ||\beta||_2^p \, \mathbb{E}_{z \sim \chi^2}[z^{\frac{1}{2}}] \tag{A.32}$$

The expectation now corresponds to $\frac{1}{2}^{th}$ centered moment of the χ^2 distribution with m degrees of freedom, whose kth moments are given as:

$$\mathbb{E}_{z \sim \chi^2}[z^k] = 2^k \frac{\Gamma(\frac{m}{2} + k)}{\Gamma(\frac{m}{2})} \tag{A.33}$$

This ends the proof of the equality.

BIBLIOGRAPHY

Abbasnejad, Ehsan, Damien Teney, Amin Parvaneh, Javen Shi, and Anton van den Hengel (2020). "Counterfactual vision and language learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10044–10054 (cit. on p. 27).

Agarwal, Vedika, Rakshith Shetty, and Mario Fritz (2020). "Towards causal vqa: Revealing and reducing spurious correlations by invariant and covariant semantic editing". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9690–9698 (cit. on pp. 16, 45, 46, 142).

Agrawal, Aishwarya (2019). "Visual question answering and beyond". PhD thesis. Georgia Institute of Technology (cit. on p. 73).

Agrawal, Aishwarya, Dhruv Batra, and Devi Parikh (2016). "Analyzing the Behavior of Visual Question Answering Models". In: *EMNLP*, pp. 1955–1960 (cit. on pp. 35, 45).

Agrawal, Aishwarya, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi (2018). "Don't Just Assume; Look and Answer: Overcoming Priors for Visual Question Answering". In: *CVPR* (cit. on pp. 26, 47–49, 53, 58, 59, 66, 73).

Anderson, Peter, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang (2018). "Bottom-up and top-down attention for image captioning and visual question answering". In: *CVPR*, pp. 6077–6086 (cit. on pp. 4, 5, 20–22, 28, 30, 45, 47, 58–60, 64–66, 89, 110, 127, 130, 134).

Andreas, Jacob, Marcus Rohrbach, Trevor Darrell, and Dan Klein (2016). "Neural module networks". In: *CVPR* (cit. on p. 26).

Antol, Stanislaw, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh (2015). "Vqa: Visual question answering". In: *ICCV* (cit. on pp. 17, 18, 38, 39, 60, 66).

Arora, S., S.S. Du, W. Hu, Z. Li, and R. Wang (2019). "Fine-grained Analysis of optimization and generalization for overparametrized two-layer neural networks". In: *ICML* (cit. on pp. 114, 116, 118, 147, 148).

Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E Hinton (2016). "Layer normalization". In: *arXiv preprint arXiv:1607.06450* (cit. on pp. 107, 130).

Bahdanau, Dzmitry, Harm de Vries, Timothy J O'Donnell, Shikhar Murty, Philippe Beaudoin, Yoshua Bengio, and Aaron Courville (2019). "Closure: Assessing systematic generalization of clevr models". In: *arXiv preprint arXiv:1912.05783* (cit. on p. 47).

Baradel, Fabien, Natalia Neverova, Julien Mille, Greg Mori, and Christian Wolf (2019). "CoPhy: Counterfactual Learning of Physical Dynamics". In: *International Conference on Learning Representations* (cit. on pp. 13, 142).

Barrett, David, Felix Hill, Adam Santoro, Ari Morcos, and Timothy Lillicrap (2018). "Measuring abstract reasoning in neural networks". In: *International conference on machine learning*. PMLR, pp. 511–520 (cit. on pp. 3, 18).

Battaglia, Peter W, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan

Faulkner, et al. (2018). “Relational inductive biases, deep learning, and graph networks”. In: *arXiv preprint arXiv:1806.01261* (cit. on p. 22).
Beery, Sara, Grant Van Horn, and Pietro Perona (2018). “Recognition in terra incognita”. In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 456–473 (cit. on p. 14).
Belkin, Mikhail (2021). “Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation”. In: *arXiv preprint arXiv:2105.14368* (cit. on p. 114).
Ben-Younes, Hedi, Rémi Cadene, Matthieu Cord, and Nicolas Thome (2017). “Mutan: Multimodal tucker fusion for visual question answering”. In: *ICCV* (cit. on pp. 3, 22).
Ben-Younes, Hedi, Remi Cadene, Nicolas Thome, and Matthieu Cord (2019). “Block: Bilinear superdiagonal fusion for visual question answering and visual relationship detection”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33, pp. 8102–8109 (cit. on p. 22).
Bhattacharya, Nilavra and Danna Gurari (2019). “VizWiz Dataset Browser: A Tool for Visualizing Machine Learning Datasets”. In: *arXiv preprint arXiv:1912.09336* (cit. on p. 41).
Bommasani, Rishi, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. (2021). “On the Opportunities and Risks of Foundation Models”. In: *arXiv preprint arXiv:2108.07258* (cit. on p. 12).
Bottou, Léon (2014). “From machine learning to machine reasoning”. In: *Machine learning* 94.2, pp. 133–149 (cit. on pp. 11–14, 35, 87, 121).
Bugliarello, Emanuele, Ryan Cotterell, Naoaki Okazaki, and Desmond Elliott (2020). “Multimodal Pretraining Unmasked: Unifying the Vision and Language BERTs”. In: *arXiv preprint arXiv:2011.15124* (cit. on p. 24).
Buolamwini, Joy and Timnit Gebru (2018). “Gender shades: Intersectional accuracy disparities in commercial gender classification”. In: *Conference on fairness, accountability and transparency*. PMLR, pp. 77–91 (cit. on pp. 5, 15).
Cadene, Remi, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. (2019). “RUBi: Reducing Unimodal Biases for Visual Question Answering”. In: *Advances in Neural Information Processing Systems*, pp. 839–850 (cit. on pp. 26, 50, 53, 58, 63, 64, 66).
Cao, Jize, Zhe Gan, Yu Cheng, Licheng Yu, Yen-Chun Chen, and Jingjing Liu (2020). “Behind the Scene: Revealing the Secrets of Pre-trained Vision-and-Language Models”. In: *arXiv preprint arXiv:2005.07310* (cit. on p. 78).
Carter, Shan, Zan Armstrong, Ludwig Schubert, Ian Johnson, and Chris Olah (2019). “Activation Atlas”. In: *Distill*. https://distill.pub/2019/activation-atlas (cit. on p. 77).
Cashman, Dylan, Genevieve Patterson, Abigail Mosca, Nathan Watts, Shannon Robinson, and Remco Chang (2018). “Rnnbow: Visualizing learning via backpropagation gradients in rnns”. In: *IEEE Computer Graphics and Applications* 38.6, pp. 39–50 (cit. on p. 77).
Chaabouni, Rahma, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni (2020). “Compositionality and Generalization In Emergent Languages”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4427–4442 (cit. on p. 143).

Chandrasekaran, A., V. Prabhu, D. Yadav, P. Chattopadhyay, and D. Parikh (2018). "Do explanations make VQA models more predictable to a human?" In: *EMNLP* (cit. on p. 82).

Chen, Wenhu, Zhe Gan, Linjie Li, Yu Cheng, William Wang, and Jingjing Liu (2021). "Meta module network for compositional visual reasoning". In: *WACV* (cit. on pp. 26, 60, 66, 128, 134).

Chen, Yen-Chun, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu (2020). "UNITER: UNiversal image-TExt representation learning". In: *ECCV* (cit. on p. 24).

Cho, Kyunghyun, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio (2014). "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *EMNLP*, pp. 1724–1734 (cit. on pp. 20, 128, 130).

Chollet, François (2019). "On the measure of intelligence". In: *arXiv preprint arXiv:1911.01547* (cit. on pp. 3, 18).

Clark, Christopher, Mark Yatskar, and Luke Zettlemoyer (2019). "Don't Take the Easy Way Out: Ensemble Based Methods for Avoiding Known Dataset Biases". In: *EMNLP*, pp. 4060–4073 (cit. on pp. 27, 50, 53, 58, 63, 64, 66).

Dancette, Corentin, Remi Cadene, Damien Teney, and Matthieu Cord (2021). "Beyond question-based biases: Assessing multimodal shortcut learning in visual question answering". In: *ICCV* (cit. on pp. 48, 142).

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: *EMNLP* (cit. on pp. 26, 136).

Das, Abhishek, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra (2018). "Embodied Question Answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 18, 142).

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pp. 248–255 (cit. on p. 20).

DeRose, Joseph F, Jiayao Wang, and Matthew Berger (2020). "Attention Flows: Analyzing and Comparing Attention Mechanisms in Language Models". In: *IEEE Transactions on Visualization and Computer Graphics* (cit. on pp. 73, 78).

Descola, Philippe (2013). *Beyond nature and culture*. University of Chicago Press (cit. on p. 14).

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (cit. on pp. 3, 20, 24, 30, 79, 105).

Forde, Jessica Zosa and Michela Paganini (2019). "The scientific method in the science of machine learning". In: *arXiv preprint arXiv:1904.10922* (cit. on pp. 51, 141).

Fukui, Akira, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach (2016). "Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding". In: *EMNLP* (cit. on p. 22).

Gao, Peng, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven CH Hoi, Xiaogang Wang, and Hongsheng Li (2019). "Dynamic fusion with intra-and inter-modality attention flow for visual question answering". In: *CVPR* (cit. on p. 23).

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: *Nature Machine Intelligence* 2.11, pp. 665–673 (cit. on pp. ix, 4, 12, 15, 121, 142).

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. ix, 3, 16).

Gokhale, Tejas, Pratyay Banerjee, Chitta Baral, and Yezhou Yang (2020a). "MUTANT: A Training Paradigm for Out-of-Distribution Generalization in Visual Question Answering". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 878–892 (cit. on p. 27).

Gokhale, Tejas, Pratyay Banerjee, Chitta Baral, and Yezhou Yang (2020b). "Vqa-lol: Visual question answering under the lens of logic". In: *European conference on computer vision*. Springer, pp. 379–396 (cit. on p. 47).

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep learning*. MIT press (cit. on pp. ix, 3, 6).

Gorman, Kyle and Steven Bedrick (2019). "We need to talk about standard splits". In: *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 2786–2791 (cit. on pp. 51, 141).

Goyal, Yash, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh (2017). "Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering". In: *CVPR*, pp. 6904–6913 (cit. on pp. 18, 37–40, 53, 55, 58, 66, 110, 123).

Goyal, Yash, Akrit Mohapatra, Devi Parikh, and Dhruv Batra (2016). "Towards transparent ai systems: Interpreting visual question answering models". In: *arXiv preprint arXiv:1608.08974* (cit. on p. 78).

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*, pp. 3608–3617 (cit. on pp. 40, 141).

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*, pp. 770–778 (cit. on p. 20).

Held, Richard and Alan Hein (1963). "Movement-produced stimulation in the development of visually guided behavior." In: *Journal of comparative and physiological psychology* 56.5, p. 872 (cit. on p. 142).

Hendricks, Lisa Anne, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell (2016). "Generating visual explanations". In: *European conference on computer vision*. Springer, pp. 3–19 (cit. on p. 78).

Hendricks, Lisa Anne, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach (2018). "Women also snowboard: Overcoming bias in captioning models". In: *ECAI*. Springer, pp. 793–811 (cit. on pp. 5, 15, 26, 60).

Hendrycks, Dan and Kevin Gimpel (2016). "Gaussian error linear units (gelus)". In: *arXiv preprint arXiv:1606.08415* (cit. on p. 130).
Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780 (cit. on pp. 20, 58).
Hohman, Fred, Minsuk Kahng, Robert Pienta, and Duen Horng Chau (2018). "Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers". In: *IEEE Transactions on Visualization and Computer Graphics* (cit. on p. 73).
Hohman, Fred, Haekyu Park, Caleb Robinson, and Duen Horng Chau (2020). "Summit: Scaling Deep Learning Interpretability by Visualizing Activation and Attribution Summarizations". In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)* (cit. on p. 77).
Hohman, Fred Matthew, Minsuk Kahng, Robert Pienta, and Duen Horng Chau (2019). "Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers". In: *IEEE Transactions on Visualization and Computer Graphics* (cit. on p. 78).
Hu, Ronghang, Anna Rohrbach, Trevor Darrell, and Kate Saenko (2019). "Language-conditioned graph networks for relational reasoning". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10294–10303 (cit. on pp. 23, 110).
Hudson, Drew and Christopher D Manning (2019a). "Learning by abstraction: The neural state machine". In: *Advances in Neural Information Processing Systems*, pp. 5901–5914 (cit. on pp. 26, 110, 111, 134).
Hudson, Drew A and Christopher D Manning (2018). "Compositional Attention Networks for Machine Reasoning". In: *International Conference on Learning Representations* (cit. on pp. 110, 112).
Hudson, Drew A and Christopher D Manning (2019b). "Gqa: A new dataset for real-world visual reasoning and compositional question answering". In: *CVPR*, pp. 6700–6709 (cit. on pp. 4, 5, 16, 18, 37, 42–46, 53, 55, 58, 59, 62–64, 80, 90, 93, 110, 111, 123, 125, 127, 130, 131, 134, 140).
Jaunet, Theo, Corentin Kervadec, Romain Vuillemot, Grigory Antipov, Moez Baccouche, and Christian Wolf (2021). "VisQA: X-raying Vision and Language Reasoning in Transformers". In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)* (cit. on pp. 7, 74, 75, 78).
Jiang, Huaizu, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen (2020). "In defense of grid features for visual question answering". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10267–10276 (cit. on pp. 21, 141).
Jiang, Yu, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh (2018). "Pythia vo. 1: the winning entry to the vqa challenge 2018". In: *arXiv preprint arXiv:1807.09956* (cit. on p. 47).
Johnson, Justin, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick (2017). "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning". In: *CVPR*, pp. 2901–2910 (cit. on pp. 18, 37, 40, 41, 47, 140).
Kafle, Kushal and Christopher Kanan (2017). "An Analysis of Visual Question Answering Algorithms". In: *ICCV* (cit. on p. 44).

Karpathy, Andrej and Li Fei-Fei (2015a). "Deep visual-semantic alignments for generating image descriptions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3128–3137 (cit. on pp. 17, 105).

Karpathy, Andrej, Justin Johnson, and Li Fei-Fei (2015b). "Visualizing and understanding recurrent networks". In: *arXiv preprint arXiv:1506.02078* (cit. on p. 77).

Kazemzadeh, Sahar, Vicente Ordonez, Mark Matten, and Tamara Berg (2014). "Referitgame: Referring to objects in photographs of natural scenes". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 787–798 (cit. on p. 17).

Kervadec, C., G. Antipov, M. Baccouche, and C. Wolf (2021a). "Roses Are Red, Violets Are Blue... but Should VQA Expect Them To?" In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 91, 134).

Kervadec, Corentin, Grigory Antipov, Moez Baccouche, and Christian Wolf (2019). "Weak Supervision helps Emergence of Word-Object Alignment and improves Vision-Language Tasks". In: *European Conference on Artificial Intelligence (ECAI)* (cit. on pp. 7, 104).

Kervadec, Corentin, Grigory Antipov, Moez Baccouche, and Christian Wolf (2021b). "Roses Are Red, Violets Are Blue... but Should Vqa Expect Them To?" In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 7, 36).

Kervadec, Corentin, Theo Jaunet, Grigory Antipov, Moez Baccouche, Romain Vuillemot, and Christian Wolf (2021c). "How Transferable are Reasoning Patterns in VQA?" In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 7, 74, 104).

Kervadec, Corentin, Christian Wolf, Grigory Antipov, Moez Baccouche, and Madiha Nadri (2021d). "Supervising the Transfer of Reasoning Patterns in VQA". In: *Advances in Neural Information Processing Systems (NeurIPS)* (cit. on pp. 7, 104).

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. 22, 60, 66, 126, 127, 134).

Kim, Jin-Hwa, Kyoung-Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang (2016). "Hadamard product for low-rank bilinear pooling". In: *arXiv preprint arXiv:1610.04325* (cit. on p. 22).

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: (cit. on pp. 58, 111, 124, 130).

Kipf, Thomas N and Max Welling (2017). "Semi-supervised classification with graph convolutional networks". In: (cit. on p. 23).

Krishna, Ranjay, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. (2017). "Visual genome: Connecting language and vision using crowdsourced dense image annotations". In: *IJCV* 123.1, pp. 32–73 (cit. on pp. 59, 79, 88, 110, 123, 130).

Kwon, Bum Chul, Min-Je Choi, Joanne Taery Kim, Edward Choi, Young Bin Kim, Soonwook Kwon, Jimeng Sun, and Jaegul Choo (2018). "Retainvis: Visual analytics with interpretable and interactive recurrent neural networks on electronic medical records". In: *IEEE transactions on visualization and computer graphics* 25.1, pp. 299–309 (cit. on p. 77).

Lazaridou, Angeliki, Alexander Peysakhovich, and Marco Baroni (2017). "Multi-Agent Cooperation and the Emergence of (Natural) Language". In: (cit. on p. 142).
Lee, Kuang-Huei, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He (2018). "Stacked cross attention for image-text matching". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 201–216 (cit. on p. 17).
Legg, Shane, Marcus Hutter, et al. (2007). "A collection of definitions of intelligence". In: (cit. on p. 12).
Li, Linjie, Jie Lei, Zhe Gan, and Jingjing Liu (2021). "Adversarial VQA: A New Benchmark for Evaluating the Robustness of VQA Models". In: *arXiv preprint arXiv:2106.00245* (cit. on pp. 48, 141).
Li, Liunian Harold, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang (2020a). "What Does BERT with Vision Look At?" In: *ACL (short)* (cit. on p. 78).
Li, Xiujun, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. (2020b). "Oscar: Object-semantics aligned pre-training for vision-language tasks". In: *European Conference on Computer Vision*. Springer, pp. 121–137 (cit. on p. 24).
Li, Yikang, Nan Duan, Bolei Zhou, Xiao Chu, Wanli Ouyang, Xiaogang Wang, and Ming Zhou (2018). "Visual question generation as dual task of visual question answering". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6116–6124 (cit. on p. 17).
Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick (2014). "Microsoft coco: Common objects in context". In: *ECCV* (cit. on pp. 17, 58, 79, 105, 110, 123, 130).
Lipton, Zachary C (2016). "The mythos of model interpretability". In: *arXiv preprint arXiv:1606.03490* (cit. on p. 73).
Liu, Mengchen, Jiaxin Shi, Zhen Li, Chongxuan Li, Jun Zhu, and Shixia Liu (2016). "Towards better analysis of deep convolutional neural networks". In: *IEEE transactions on visualization and computer graphics* 23.1, pp. 91–100 (cit. on p. 77).
Lu, Jiasen, Dhruv Batra, Devi Parikh, and Stefan Lee (2019). "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks". In: *NeurIPS* (cit. on pp. 24, 30, 111).
Lu, Jiasen, Jianwei Yang, Dhruv Batra, and Devi Parikh (2016). "Hierarchical question-image co-attention for visual question answering". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 289–297 (cit. on p. 82).
Malinowski, Mateusz and Mario Fritz (2014). "A multi-world approach to question answering about real-world scenes based on uncertain input". In: *NeurIPS* (cit. on pp. 44, 45).
Manjunatha, Varun, Nirat Saini, and Larry S Davis (2019). "Explicit bias discovery in visual question answering models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9562–9571 (cit. on p. 78).
Mansimov, Elman, Emilio Parisotto, Lei Jimmy Ba, and Ruslan Salakhutdinov (2016). "Generating Images from Captions with Attention". In: *ICLR* (cit. on p. 17).
Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013). "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*, pp. 3111–3119 (cit. on p. 19).

Mitchell, Tom M (1980). *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research ... (cit. on p. 14).
Norcliffe-Brown, Will, Stathis Vafeias, and Sarah Parisot (2018). "Learning Conditioned Graph Structures for Interpretable Visual Question Answering". In: *NeurIPS* (cit. on p. 23).
Olah, Chris and Shan Carter (2016). "Attention and Augmented Recurrent Neural Networks". In: *Distill*. URL: http://distill.pub/2016/augmented-rnns (cit. on p. 78).
Park, Dong Huk, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach (2018). "Multimodal explanations: Justifying decisions and pointing to the evidence". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8779–8788 (cit. on p. 78).
Pearl, Judea et al. (2000). "Models, reasoning and inference". In: *Cambridge, UK: CambridgeUniversityPress* 19 (cit. on pp. 13, 142).
Pennington, Jeffrey, Richard Socher, and Christopher D Manning (2014). "Glove: Global vectors for word representation". In: *EMNLP*, pp. 1532–1543 (cit. on p. 19).
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*. Vol. 32. 1 (cit. on p. 112).
Picard, David (2021). "Torch. manual_seed (3407) is all you need: On the influence of random seeds in deep learning architectures for computer vision". In: *arXiv preprint arXiv:2109.08203* (cit. on p. 141).
Plummer, Bryan A, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik (2015). "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2641–2649 (cit. on p. 17).
Popper, Karl (1934). *The Logic of Scientific Discovery*. Routledge (cit. on p. 12).
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: *arXiv preprint arXiv:2103.00020* (cit. on p. 142).
Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang (2016). "SQuAD: 100, 000+ Questions for Machine Comprehension of Text". In: *EMNLP* (cit. on pp. 17, 18).
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*, pp. 1541–1551 (cit. on pp. 26, 50).
Rame, Alexandre, Corentin Dancette, and Matthieu Cord (2021). "Fishr: Invariant Gradient Variances for Out-of-distribution Generalization". In: *arXiv preprint arXiv:2109.02934* (cit. on p. 142).
Ramesh, Aditya, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever (2021). "Zero-shot text-to-image generation". In: *arXiv preprint arXiv:2102.12092* (cit. on pp. 17, 142).
Ramsauer, Hubert, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff,

et al. (2020). “Hopfield networks is all you need”. In: *arXiv preprint arXiv:2008.02217* (cit. on pp. 79, 91, 92, 94–96).

Ray, Arijit, Karan Sikka, Ajay Divakaran, Stefan Lee, and Giedrius Burachas (2019). “Sunny and Dark Outside?! Improving Answer Consistency in VQA through Entailed Question Generation”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5863–5868 (cit. on p. 46).

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*, pp. 91–99 (cit. on pp. 3, 20, 21, 28, 80, 88, 90, 107, 123, 134).

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*. ACM, pp. 1135–1144 (cit. on p. 73).

S. Shalev-Shwart, Shai and S. Ben-David (2014). *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press (cit. on p. 114).

Santoro, Adam, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap (2017). “A simple neural network module for relational reasoning”. In: *Advances in Neural Information Processing Systems* 30 (cit. on p. 23).

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. 142).

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: *ICCV* (cit. on pp. 27, 50).

Selvaraju, Ramprasaath R, Purva Tendulkar, Devi Parikh, Eric Horvitz, Marco Tulio Ribeiro, Besmira Nushi, and Ece Kamar (2020). “SQuINTing at VQA Models: Introspecting VQA Models With Sub-Questions”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10003–10011 (cit. on p. 46).

Shah, Harshay, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli (2020). “The Pitfalls of Simplicity Bias in Neural Networks”. In: *NeurIPS*. URL: https://proceedings.neurips.cc/paper/2020/hash/6cfe0e6127fa25df2a0ef2ae1067d915-Abstract.html (cit. on pp. 15, 142).

Shah, Meet, Xinlei Chen, Marcus Rohrbach, and Devi Parikh (2019). “Cycle-consistency for robust visual question answering”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6649–6658 (cit. on pp. 16, 45).

Sheng, Sasha, Amanpreet Singh, Vedanuj Goswami, Jose Alberto Lopez Magana, Wojciech Galuba, Devi Parikh, and Douwe Kiela (2021). “Human-Adversarial Visual Question Answering”. In: *arXiv preprint arXiv:2106.02280* (cit. on pp. 36, 48, 141).

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* (cit. on p. 50).

Shridhar, Mohit, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox (2020). "Alfred: A benchmark for interpreting grounded instructions for everyday tasks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10740–10749 (cit. on p. 18).

Strobelt, Hendrik, Sebastian Gehrmann, Michael Behrisch, Adam Perer, Hanspeter Pfister, and Alexander M Rush (2018). "Seq2seq-vis: A visual debugging tool for sequence-to-sequence models". In: *IEEE transactions on visualization and computer graphics* 25.1, pp. 353–363 (cit. on p. 78).

Strobelt, Hendrik, Sebastian Gehrmann, Hanspeter Pfister, and Alexander M Rush (2017). "Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks". In: *IEEE transactions on visualization and computer graphics* 24.1, pp. 667–676 (cit. on p. 77).

Suhr, Alane, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi (2019). "A Corpus for Reasoning about Natural Language Grounded in Photographs". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6418–6428 (cit. on pp. 17, 109, 111, 112).

Sun, Chen, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid (2019). "Videobert: A joint model for video and language representation learning". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7464–7473 (cit. on p. 25).

Tan, Hao and Mohit Bansal (2019). "LXMERT: Learning Cross-Modality Encoder Representations from Transformers". In: *EMNLP*, pp. 5103–5114 (cit. on pp. 24, 25, 27, 28, 30, 31, 47, 58–60, 63, 66, 79, 91, 103–105, 107, 110–112, 118, 123–125, 127, 130–132, 134, 135).

Teney, Damien, Ehsan Abbasnedjad, and Anton van den Hengel (2020a). "Learning what makes a difference from counterfactual examples and gradient supervision". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*. Springer, pp. 580–599 (cit. on pp. 13, 27, 142).

Teney, Damien, Ehsan Abbasnejad, and Anton van den Hengel (2020b). "Unshuffling data for improved generalization". In: *arXiv preprint arXiv:2002.11894* (cit. on pp. 27, 50).

Teney, Damien, Ehsan Abbasnejad, Kushal Kafle, Robik Shrestha, Christopher Kanan, and Anton van den Hengel (2020c). "On the Value of Out-of-Distribution Testing: An Example of Goodhart's Law". In: *NeurIPS* (cit. on pp. 27, 47, 49, 50).

Teney, Damien, Ehsan Abbasnejad, Simon Lucey, and Anton van den Hengel (2021). "Evading the Simplicity Bias: Training a Diverse Set of Models Discovers Solutions with Superior OOD Generalization". In: *arXiv preprint arXiv:2105.05612* (cit. on p. 142).

Teney, Damien, Lingqiao Liu, and Anton van Den Hengel (2017). "Graph-structured representations for visual question answering". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9 (cit. on p. 23).

Valiant, L.G. (1984). "A theory of the learnable". In: *Communications of the ACM*. Vol. 27(11) (cit. on pp. 114–116).

Vani, Ankit, Max Schwarzer, Yuchen Lu, Eeshan Dhekane, and Aaron Courville (2021). "Iterated learning for emergent systematicity in {VQA}". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=Pd_oMxH8IlF (cit. on p. 143).

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*, pp. 5998–6008 (cit. on pp. x, 3, 21, 23, 28, 78, 105).

Vig, Jesse (2019a). "A Multiscale Visualization of Attention in the Transformer Model". In: *arXiv preprint arXiv:1906.05714*. URL: https://arxiv.org/abs/1906.05714 (cit. on p. 73).

Vig, Jesse (2019b). "A Multiscale Visualization of Attention in the Transformer Model". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (cit. on p. 78).

Voita, Elena, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov (2019). "Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5797–5808 (cit. on p. 96).

Wikipedia (2021). *Commonsense reasoning — Wikipedia, The Free Encyclopedia*. [Online; accessed 10-September-2021]. URL: https://en.wikipedia.org/wiki/Commonsense_reasoning (cit. on p. 13).

Wiktionary (2021). *Reasoning — Wiktionary, The Free Dictionary*. [Online; accessed 10-September-2021]. URL: https://en.wiktionary.org/wiki/reasoning (cit. on p. 11).

Wu, Jialin and Raymond Mooney (2019). "Self-Critical Reasoning for Robust Visual Question Answering". In: *Advances in Neural Information Processing Systems*, pp. 8601–8611 (cit. on pp. 27, 50).

Wu, Qi, Damien Teney, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel (2017). "Visual question answering: A survey of methods and datasets". In: *Computer Vision and Image Understanding* 163, pp. 21–40 (cit. on p. 44).

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. (2016). "Google's neural machine translation system: Bridging the gap between human and machine translation". In: *arXiv preprint arXiv:1609.08144* (cit. on p. 28).

Xie, Ning, Farley Lai, Derek Doran, and Asim Kadav (2019). "Visual entailment: A novel task for fine-grained image understanding". In: *arXiv preprint arXiv:1901.06706* (cit. on p. 17).

Xu, K., J. Li, M. Zhang, S.S. Du, K.-I. K., and S. Jegelka (2020). "What can Neural Networks Reason About". In: *ICLR* (cit. on pp. 106, 114, 116–118, 149).

Xu, Kelvin, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio (2015). "Show, attend and tell: Neural image caption generation with visual attention". In: *International conference on machine learning*. PMLR, pp. 2048–2057 (cit. on pp. 20, 21).

Yang, Zichao, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola (2016). "Stacked attention networks for image question answering". In: *CVPR*, pp. 21–29 (cit. on p. 22).

Yi, Kexin, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum (2018). "Neural-symbolic vqa: Disentangling reasoning from vision and language understanding". In: *NeurIPS* (cit. on pp. 26, 38, 42).

Yu, Zhou, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian (2019). "Deep modular co-attention networks for visual question answering". In: *CVPR*, pp. 6281–6290 (cit. on pp. 3, 23, 24, 29, 58, 60, 66, 127, 134).

Zeiler, Matthew D and Rob Fergus (2014). "Visualizing and understanding convolutional networks". In: *European conference on computer vision*. Springer, pp. 818–833 (cit. on p. 77).

Zhang, C., S. Bengio, M. Hardt, B. Recht, and O. Vinyals (2017). "Understanding deep learning requires rethinking generalization ". In: *ICLR* (cit. on p. 114).

Zhang, Pengchuan, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao (2021). "Vinvl: Revisiting visual representations in vision-language models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5579–5588 (cit. on pp. 20, 38, 134, 135, 141).

Zhou, Zhi-Hua (2018). "A brief introduction to weakly supervised learning". In: *National science review* 5.1, pp. 44–53 (cit. on p. 109).

Zhu, Yuke, Oliver Groth, Michael Bernstein, and Li Fei-Fei (2016). "Visual7w: Grounded question answering in images". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4995–5004 (cit. on p. 44).

FOLIO ADMINISTRATIF

THESE DE L'UNIVERSITE DE LYON OPEREE AU SEIN DE L'INSA LYON

NOM : KERVADEC DATE de SOUTENANCE : 09/12/2021

Prénoms : Corentin Adrien Joseph

TITRE : Biais et raisonnement dans les systèmes de questions réponses visuelles

NATURE : Doctorat Numéro d'ordre : 2021LYSEI101

Ecole doctorale : Ecole doctorale d'Informatique et Mathématique de Lyon (INFOMATHS, ED 512)

Spécialité : Informatique

RESUME :

De quelle couleur est le terrain de tennis ? Quelle est la taille du chien ? Y a-t-il une voiture à droite du vélo sous le cocotier ? Répondre à ces questions fondamentales est le sujet de la tâche appelée question-réponses visuelle (VQA, en anglais), dans laquelle un agent doit répondre à des questions posées sur des images.

Plus précisément, le VQA requiert de mettre au point un agent capable de maitriser une grande variété de compétences : reconnaître des objets, reconnaitre des attributs (couleur, taille, matériaux, *etc*.), identifier des relations (e.g. spatiales), déduire des enchainements logiques, *etc*. C'est pourquoi, le VQA est parfois désigné comme un test de Turing visuel, dont le but est d'évaluer la capacité d'un agent à raisonner sur des images. Cette tâche a récemment connu d'important progrès grâce à l'utilisation des réseaux de neurones et de l'apprentissage profond.

Après une revue détaillée de l'État de l'Art sur le VQA, ainsi qu'une définition de notre utilisation du terme *raisonnement*, nous nous intéressons à la question suivante : *les modèles de VQA actuels raisonnent-ils vraiment ?* La mise en œuvre d'une nouvelle méthode d'évaluation (GQA-OOD) nous permettra de répondre négativement à cette question. En particulier, nous mettrons en évidence la tendance des modèles à apprendre des raccourcis, autrement appelés *biais*, présent dans les données d'entrainement, mais heurtant les capacités de généralisation. Nous proposerons alors, dans une troisième partie une analyse approfondie des mécanismes d'attention appris par les réseaux de neurones artificiels. Nous étudierons quels sont les enchainements aboutissant à un raisonnement, ou, au contraire, à une prédiction biaisée par un raccourci frauduleux. La dernière et quatrième partie tire conclusion de nos évaluations et analyses, afin de développer de nouvelles méthodes améliorant les performances des modèles de VQA.

En résumé, cette thèse a pour objet l'étude du raisonnement visuel dans des réseaux de neurones artificiels entrainés par apprentissage profond, dans le cadre du VQA. Mais surtout, ce qui nous intéressera en premier lieu, c'est l'évaluation et l'analyse de l'influence qu'ont les biais, présents dans les données d'apprentissage, sur les prédictions de nos modèles.

MOTS-CLÉS : Machine Learning; Deep Learning; Vision and Language; Visual Reasoning; // Apprentissage Automatique ; Apprentissage Profond; Vision et Langage; Raisonnement Visuel ;

Laboratoire (s) de recherche : LIRIS (INSA Lyon), Orange Innovation

Directeur de thèse: Christian Wolf (directeur), Grigory Antipov (co-encadrant), Moez Baccouche (co-encadrant)

Président de jury : David Picard

Composition du jury : Rapporteur : David Picard, Nicolas Thome // Examinateur·rice·s : Cordelia Schmid, Damien Teney, Akata Zeynep // Directeur de thèse : Christian Wolf // Co-encadrants : Grigory Antipov, Moez Baccouche