SCIENCES
SORBONNE
UNIVERSITÉ

**THÈSE DE DOCTORAT DE
SORBONNE UNIVERSITÉ**

Spécialité
**Informatique**
École Doctorale Informatique, Télécommunications et Électronique (Paris)

Présentée par

**Rémi Cadène**

Pour obtenir le grade de
**DOCTEUR de SORBONNE UNIVERSITÉ**

Sujet de la thèse :

# Deep Multimodal Learning for Vision and Language Processing

## Apprentissage Multimodal Profond pour le Traitement de la Vision et du Langage

Devant le jury composé de :

| | | |
|---|---|---|
| Mme. Gabriela Csurka | Naver LABS Europe | Rapportrice |
| M. Ivan Laptev | INRIA Paris | Rapporteur |
| M. Patrick Gallinari | Sorbonne Université – LIP6 | Examinateur |
| M. Thomas Serre | Brown University | Examinateur |
| M. Eduardo Valle | Campinas University – RECOD | Examinateur |
| M. Nicolas Thome | CNAM – CEDRIC | Co-directeur de thèse |
| M. Matthieu Cord | Sorbonne Université – LIP6 | Directeur de thèse |

# ABSTRACT

Digital technologies have become instrumental in transforming our society. Recent statistical methods have been successfully deployed to automate the processing of the growing amount of images, videos, and texts we produce daily. In particular, deep neural networks have been adopted by the computer vision and natural language processing communities for their ability to perform accurate image recognition and text understanding once trained on big sets of data. Advances in both communities built the groundwork for new research problems at the intersection of vision and language. Integrating language into visual recognition could have an important impact on human life through the creation of real-world applications such as next-generation search engines or AI assistants.

In the first part of this thesis, we focus on systems for cross-modal text-image retrieval. We propose a learning strategy to efficiently align both modalities while structuring the retrieval space with semantic information. In the second part, we focus on systems able to answer questions about an image. We propose a multimodal architecture that iteratively fuses the visual and textual modalities using a factorized bilinear model while modeling pairwise relationships between each region of the image. In the last part, we address the issues related to biases in the modeling. We propose a learning strategy to reduce the language biases which are commonly present in visual question answering systems.

# RÉSUMÉ

Les technologies du numérique ont joué un rôle déterminant dans la transformation de notre société. Des méthodes statistiques récentes ont été déployées avec succès afin d'automatiser le traitement de la quantité croissante d'images, de vidéos et de textes que nous produisons quotidiennement. En particulier, les réseaux de neurones profonds ont été adopté par les communautés de la vision par ordinateur et du traitement du langage naturel pour leur capacité à interpréter le contenu des images et des textes une fois entraînés sur de grands ensembles de données. Les progrès réalisés dans les deux communautés ont permis de jeter les bases de nouveaux problèmes de recherche à l'intersection entre vision et langage. L'intégration du langage dans la reconnaissance visuelle pourrait avoir un impact important sur la vie humaine grâce à la création d'applications telles que des moteurs de recherche de nouvelle génération ou des smart assistants.

Dans la première partie de cette thèse, nous nous concentrons sur des moteurs de recherche multimodaux images-textes. Nous proposons une stratégie d'apprentissage pour aligner efficacement les deux modalités tout en structurant l'espace de recherche avec de l'information sémantique. Dans la deuxième partie, nous nous concentrons sur des systèmes capables de répondre à toute question sur une image. Nous proposons une architecture multimodale qui fusionne itérativement les modalités visuelles et textuelles en utilisant un modèle bilinéaire factorisé, tout en modélisant les relations par paires entre chaque région de l'image. Dans la dernière partie, nous abordons les problèmes de biais dans la modélisation. Nous proposons une stratégie d'apprentissage réduisant les biais linguistiques généralement présents dans les systèmes de réponse aux questions visuelles.

# ACKNOWLEDGMENTS

I am very grateful to the many people who contributed to making my PhD work an entertaining, exciting and pleasant journey.

First, I would especially like to thank my two advisors, Matthieu Cord and Nicolas Thome. They have provided me unwavering support from the very first day, including for the PhD grant application I got from the SMART Labex. Their precious advice guided me to the solutions I was looking for. I always left Matthieu's office with my mind full of new ideas. I always hung up my phone with new strength and new perspective after getting advice from Nicolas. All along this work, they were able to carefully channel my enthusiasm while allowing me to feel free to explore many different paths.

I thank Gabriela Csurka and Ivan Laptev for agreeing to review my PhD thesis. I also thank Patrick Gallinari, Thomas Serre and Eduardo Valle for being part of my PhD thesis committee.

During this journey, I have been very fortunate to meet and learn from many other inspiring scientists: Ludovic Denoyer, who fuelled my curiosity about artificial neural networks back in my undergraduate days; David Picard and Laure Soulier, who shared with me numerous discussions leading to our fructuous collaboration; Edouard Oyallon who gave me a lot of good advice; Andrej Karpathy, who was so kind for his warm welcome to the Tesla adventure; Devi Parikh and Dhruv Batra, who have so nicely accepted me in their research team during my stay at FAIR Menlo Park; Marcus Rohrbach and Xinlei Chen who shared with me many interesting debates at FAIR; Luigi Celona for our early collaboration on pretrained models.

Obviously, I am indebted to the MLIA research team as well as so many members of LIP6. Many have been my teachers, and I am particularly grateful to Patrick Gallinari and Christophe Marsala. Others have supported me from the first days as Christophe Bouder with the fun and the compute, and as Nadine Taniou with her deep kindness.

I also thank my fellow PhD students that have been precious co-authors and friends during this journey: Hedi Ben-Younes who opened the way to VQA on which I could progress with him; Micael Carvalho with whom I share the same taste for adventure; Corentin Dancette who have been of tremendous help during my last year.

The many hours shared with all post-doc and PhD students were precious, including crazy foosball games and night-long raclette parties. Many of you shared my offices during this journey and became my friends, such as Emmanuel De Bezenac, Michael Blot, Michael Chen, Yifu Chen, Marion Chevalier, Gabriella Contardo, Arnaud Dapogny, Edouard Delasalles, Arthur Douillard, Thibaut Durand, Yonathan Efroni, Martin Engilberge, Clara Gainon de Forsan, Thomas Gérald, Jonathan Lacotte, Mehdi Lamrayah, Ahmed Mazari, Taylor Mordan, Thomas Robert, Antoine Saporta, Rémy Sun, Tom Veniat, Xin Wang, Éloi Zablocki, ... but there are too many of you to be all named!

I should not conclude without a special thought to my family and to my friends for their love and absolute support. Among many people who are too numerous for being listed here, I would like to especially thank Simon for your helpful hosting while I was writing this thesis, Raphael and Milan for always being there when I need it, Florence for your infinite patience and affectionate caring, Aude and Sophie for your grenadine syrup and kiri, Guillaume for introducing me to climbing among other things, Philippe and Virginie for your help and profound kindness.

# CONTENTS

# LIST OF FIGURES

Chapter 1: general introduction 1

Chapter 4: reducing unimodal biases for visual question answering 79

# LIST OF TABLES

# ACRONYMS

| | |
|---|---|
| AI | Artificial Intelligence |
| AMT | Amazon Mechanical Turk |
| Bi-RNN | Bidirectional Recurrent Neural Network |
| Bi-LSTM | Bidirectional Long Short-Term Memory |
| CCA | Canonical Correlation Analysis |
| ConvNet | Convolutional Neural Network |
| CV | Computer Vision |
| DL | Deep Learning |
| DT-RNN | Dependecy-Tree Recursive Neural Network |
| FCN | Fully Convolutional Network |
| FiLM | Featurewise Linear Modulation |
| GRU | Gated Recurrent Unit |
| IoU | Intersection over Union |
| LSTM | Long-Short Term Memory |
| MAC | Memory, Attention, and Composition |
| MLP | Multi-Layer Perceptron |
| ML | Machine Learning |
| NAG | Nesterov's Accelerated Gradient |
| NLP | Natural Language Processing |
| NMS | Non-Maximum Suppression |
| R-CNN | Region-based Convolutional Neural Network |
| ReLU | Rectified Linear Unit |
| RNN | Recurrent Neural Network |
| RoI | Region of Interest |
| SGD | Stochastic Gradient Descent |
| SVM | Support Vector Machine |
| VQA | Visual Question Answering |
| VISIIR | VIsual Seek for Interactive Image Retrieval |

Chapter 1

# GENERAL INTRODUCTION

## Contents

## 1.1 Context

> An attempt will be made to find how to make machines use language, form abstractions and concepts, solve the kinds of problems now reserved for humans, and improve themeselves... For the present purpose, the artificial intelligence problem is taken to be that of making a machine behave in ways that would be called intelligent if a human were so behaving.
>
> *John McCarthy, Marvin Minsky, Nathaniel Rochester, and Claude Shannon*
> "Proposal for the Dartmouth Workshop on Artificial Intelligence," 1955

The Dartmouth Workshop held during summer 1956 is considered to be the founding event of Artificial Intelligence (AI) as a research field. Since then, the field has experienced several hype cycles associated with major breakthroughs followed by disappointments and criticisms. Around a decade ago, AI has entered a new era with the emergence of Deep Learning (DL). This family of statistical methods is based upon deep neural networks and belongs to the broader family of Machine Learning (ML) methods. Thanks to their ability to learn complex behaviors from a massive amount of data, these artificial neural networks are at the heart of a new wave of automation. The rapid adoption of Deep Learning

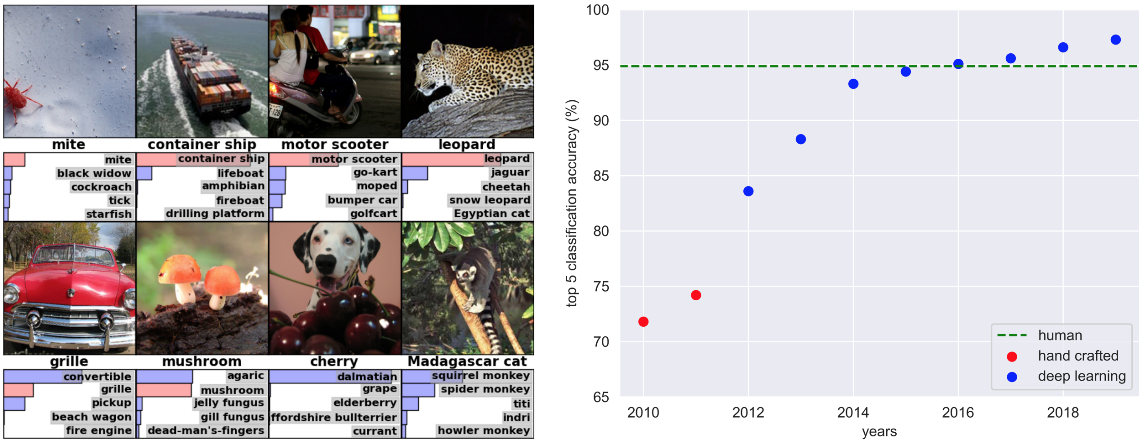
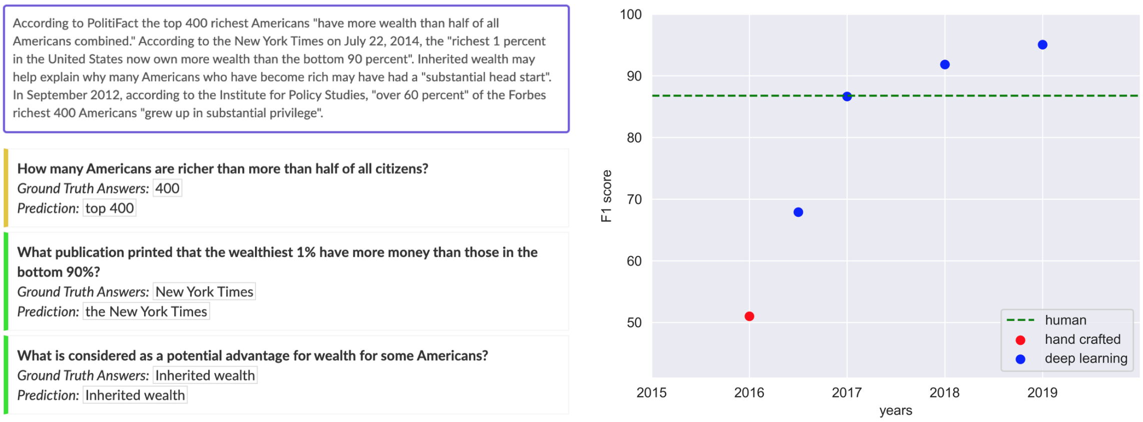


Figure 1.1 – Progress in vision illustrated by the improvements provided by Deep Learning approaches at the ILSVRC (Russakovsky et al. 2015a) large-scale image classification challenge over the years. On the left, examples of deep neural network predictions on this task. The correct class in red must appear in the 5 predicted classes. On the right, Deep Learning approaches significantly surpass handcrafted approaches and reach the estimated performance of a single trained human on this task.

is largely due to the exponential growth of available data, the creation of cheap specialized hardware and the important breakthroughs in research.

In particular, a foundational event in 2012 led the Computer Vision (CV) community to adopt the Deep Learning approaches. For the first time, the winning solution (Krizhevsky et al. 2012) of ILSVRC, a large scale image classification challenge, (Russakovsky et al. 2015a) was a deep neural network. As illustrated in Figure 1.1, the goal of this challenge is to develop approaches able to associate a label to an image using a training set of 1.2 million labeled images. Importantly, the labels come from a set of a thousand diverse nouns designating animals, plants, activities, materials, instruments, scenes or foods. Contrarily to the handcrafted methods which rely on a critical features engineering step in their classification pipeline (Fournier et al. 2001; Csurka et al. 2004), this network is able to learn richer features from a random initialization of its 60 million parameters. In the following years, Deep Learning approaches improved and have been extended to more complex visual tasks such as object detection or segmentation, exceeding all expectations (Girshick et al. 2014; S. Ren et al. 2015; He et al. 2016; He et al. 2017). However, these networks only describe visual content with simple words. They appear limited when compared to humans which are able to use language to precisely describe their visual surroundings and to interact with others.

Similarly to the Computer Vision community, the Natural Language Processing (NLP) community also adopted Deep Learning approaches with impressive results.



According to PolitiFact the top 400 richest Americans "have more wealth than half of all Americans combined." According to the New York Times on July 22, 2014, the "richest 1 percent in the United States now own more wealth than the bottom 90 percent". Inherited wealth may help explain why many Americans who have become rich may have had a "substantial head start". In September 2012, according to the Institute for Policy Studies, "over 60 percent" of the Forbes richest 400 Americans "grew up in substantial privilege".

**How many Americans are richer than more than half of all citizens?**
*Ground Truth Answers:* 400
*Prediction:* top 400

**What publication printed that the wealthiest 1% have more money than those in the bottom 90%?**
*Ground Truth Answers:* New York Times
*Prediction:* the New York Times

**What is considered as a potential advantage for wealth for some Americans?**
*Ground Truth Answers:* Inherited wealth
*Prediction:* Inherited wealth

Figure 1.2 – Progress in language illustrated by the improvements provided by Deep Learning approaches at the SQuAD1.1 (Rajpurkar et al. 2016) question answering challenge over the years. On the left, three questions about a text, their ground truth answer, and the predicted answer of a deep neural network. On the right, Deep Learning approaches significantly surpass handcrafted approaches and reach the estimated performance of a single trained human on this task.

In 2013, the word2vec method (Tomas Mikolov et al. 2013b) produced rich word and sentence representations in the form of vectors by training their model on news articles of one billion words. These vectors are automatically organized in the latent space and relationships are learned implicitly between them. For instance, the representation of the *country-capital* relationship is obtained by subtracting the representations of the words *Spain* and *Madrid*. Then, it becomes possible to infer that Paris is the capital of France by summing the representation of the word *France* with the representation of the relation *country-capital* to obtain the representation of the word *Paris*. Quickly after, Deep Learning approaches were able to model complex sentences, translate from one language to the other and, as illustrated in Figure 1.2, answer questions about a text more accurately than ever before (Sutskever et al. 2014; Lample et al. 2016; Kiros et al. 2015; Vaswani et al. 2017; Devlin et al. 2018).

The advances of deep learning in computer vision and natural language processing drive the emergence of new research issues and tasks in AI. In particular, the community currently explores new ways to bridge the gap between vision and language in order to produce more human level behaviors. A critical step towards better visual understanding is to go beyond the simple word as a label by associating richer language structures such as a simple sentence, a description, a paragraph or even a cooking recipe with a set of ingredients. The image captioning task (Hodosh et al. 2013; Tsung-Yi Lin et al. 2014; Xu et al. 2015; Lu et al. 2017) is emblematic of this ambition. It consists in producing a textual description of an image given a dataset of image-description pairs (X. Chen et al. 2015). The

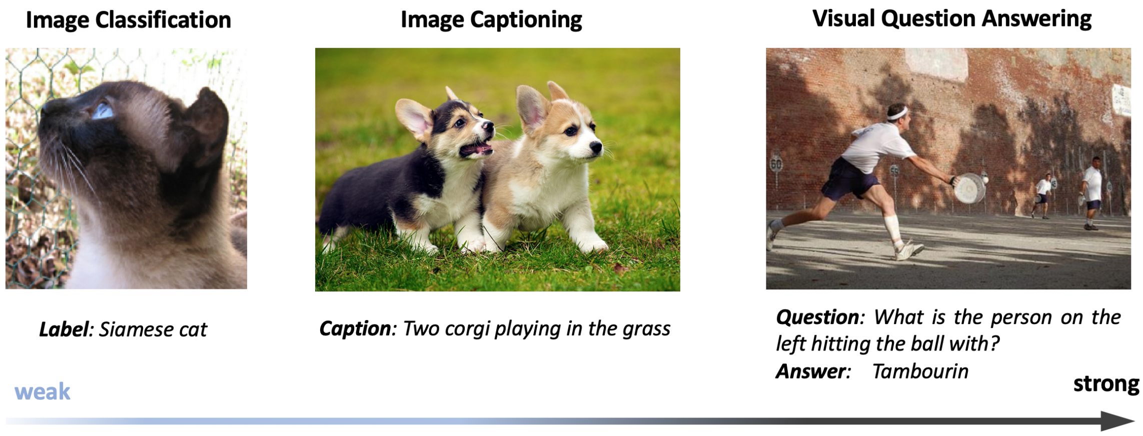


Image Classification
Image Captioning
Visual Question Answering
Label: Siamese cat
Caption: Two corgi playing in the grass
Question: What is the person on the left hitting the ball with?
Answer: Tambourin
weak
strong

Figure 1.3 – Ability to integrate language into visual recognition. On the left, the task of assigning labels to an image (Russakovsky et al. 2015a). In the middle, the task of describing an image with sentences (X. Chen et al. 2015). On the right, the task of answering any questions about an image (Agrawal et al. 2015).

ability to describe an image is at the heart of visual understanding from a human standpoint. A common way to tackle this task is through cross-modal retrieval methods (Rasiwasia et al. 2010; K. Wang et al. 2016) which enable flexible retrieval of visual and textual data across modalities. The core of these methods consists in projecting both modalities on a joint representation space where their similarity can be assessed. For instance, this allows retrieving the most similar description for a given image.

Visual Question Answering (VQA) (Malinowski et al. 2014a; Agrawal et al. 2015; Goyal et al. 2017) is another recent task that goes beyond the simple word as a label. It consists in answering a question about the visual content of an image given a dataset of image-question-answer triplets. As illustrated in Figure 1.3, VQA allows for a stronger integration of language into visual recognition than image captioning. In fact, solving VQA requires the ability to understand a very large set of concepts in order to answer the numerous questions that can potentially be asked about an image. Additionally, VQA requires a reasoning ability in order to decompose questions into sub-tasks and to address them one after the other. For instance, the question *How many people are standing on the left of the women in red* is highly compositional and requires to ground linguistic concepts on the visual scene. A smart system would answer the question sequentially by locating the women in red, by looking at the people on her left, and by counting how many are standing. Finally, VQA requires commonsense knowledge about the world. For instance, the question *In which continent has this picture been taken?* given an image of an elephant requires a smart system to know that looking at the size of the elephant ears is required to determine if it is an African or Asian elephant. For all these reasons, the VQA task is considered as a visual Turing test (Malinowski et al. 2014b).

Bridging the gap between vision and language could have a tremendous impact on human life through the creation of real-world applications such as a healthier recipe recommendation engine aiming at reducing bad eating habits (Elsweiler et al. 2017), an assistant helping visually impaired users to understand their physical and online surroundings (Gurari et al. 2018), an engine that search through large quantities of visual data via natural language interfaces (Johnson et al. 2016), or even robots using more efficient and intuitive communication interfaces (Das et al. 2017).

In Section 1.2, we first provide a background on deep learning methods for computer vision and natural language processing. This section covers the core material on which this work builds upon. In Section 1.3, we present the current challenges for developing smart AI systems at the frontier between vision and language. Finally, we provide our contributions to this aim in Section 1.4.

## 1.2 Deep learning background

In this section, we provide the basic deep learning concepts that we use throughout the next chapters of this PhD thesis. For a more in-depth introduction, we recommend the books from Bishop 2006, Nielsen 2015 and Goodfellow et al. 2016.

### 1.2.1 Learning framework

Deep learning methods are based on the statistical learning theory (Vapnik et al. 1972; Vapnik 1999). We consider the *supervised learning* setting where the goal is to estimate *the best* mapping function $f : X \rightarrow Y$ between an input space $X$ and an output space $Y$, given a training set of input-output pairs $(x, y) \in X \times Y$. In the context of a dog or cat image datasets, $X$ is the space of images and $Y$ is the space of labels in the couple $\{0, 1\}$ where 1 indicates the presence of a dog and 0 indicates the presence of a cat. A dog or cat classifier outputs a confidence score in $[0, 1]$. The latter is converted into a 0 or 1 label using a threshold which is usually set to 0.5. We assume that this training set is composed of $n$ random independent identically distributed (i.i.d) observations $(x_1, y_1), ..., (x_n, y_n) \in (X \times Y)^n$ sampled from a distribution $D$. We want to estimate $f$ such that it provides accurate output predictions on unseen input data from a testing set sampled following the same distribution $D$. We consider a loss function $l : Y \times Y \rightarrow \mathbb{R}^+$ that measures the disagreement between a predicted label $\hat{y}_i = f(x_i)$ and a truth label $y_i$.

**Learning objective** The problem of learning consists in finding the optimal function $f^*$ from a class of functions F that minimizes the risk $R(f)$ such as:

$$\begin{aligned} R(f) &:= \mathbb{E}_{(x,y)\sim D} l(f(x), y) \\ f^* &= \operatorname{argmin}_{f\in \mathrm{F}} \Big\{ R(f) \Big\} \end{aligned} \tag{1.1}$$

In practice, we do not have access to $D$ and approximate this learning problem using the available training set to minimize the empirical risk $R_n(f)$ which is defined such as:

$$\begin{aligned} R_n(f) &:= \frac{1}{n}\sum_{i=1}^{n} l(f(x_i), y_i) \\ f^* &\approx \operatorname{argmin}_{f\in \mathrm{F}} \Big\{ R_n(f) \Big\} \end{aligned} \tag{1.2}$$

However, this approximation leads to an important issue. It becomes possible to find a function $f^*$ that minimizes $R_n(f)$ while failing at predicting the correct label on the testing set. For instance, trivial memorization of the training set fails to *generalize* on unseen data. Even if the class of functions F is chosen to avoid trivial solutions, many functions might all achieve the same minimization of $R_n(f)$, but lead to highly different generalization ability.

**Regularization** Minimizing $R_n(f)$ is not sufficient to find $f^*$. One common solution consists in adding a regularization term $r(f)$ such as:

$$f^* = \operatorname{argmin}_{f\in \mathrm{F}} \Big\{ R_n(f) + r(f) \Big\} \tag{1.3}$$

$r(f)$ is a measure of the complexity of the function. Adding this term to the objective allows controlling the complexity of $f^*$ during training. By expressing $r(f)$ as a prior on $f$, Bayesian learning provides a statistical justification of the regularization. Several other practical methods and theories have been devoted to this issue (Bishop 2006).

### 1.2.2 Tasks and loss functions

We review common instantiations of loss functions, class of functions and regularization terms. We consider datasets of $n$ input-output pairs $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^d \times \mathbb{R}^t$ on which we define different machine learning problems. The regression problem consists in predicting continuous variables, while the classification problem consists in predicting discrete variables. The latter can be decomposed into three categories. In the binary setting, $t = 1$ and the only component of $\mathbf{y}$, that we define as the scalar $y$, can be either zero or one to indicate the presence or absence of the class. In the multiclass setting, only one class from $t$ classes is associated with the input. Thus, $\mathbf{y}$ can be expressed as a one-hot vector of $t$ dimensions. In the multilabel setting, several classes can be associated with the input. Thus, $\mathbf{y}$ can be expressed as a t-dimensional vector of ones and zeros.

**Loss for regression** In the context of a regression problem, we consider a $\mathbf{y} \in \mathbb{R}^t$ and a function $f : \mathbb{R}^d \rightarrow \mathbb{R}^t$. A common loss function to address the regression problem is the squared $L_2$ distance such that:

$$l(f(\mathbf{x}), \mathbf{y}) := ||f(\mathbf{x}) - \mathbf{y}||_2^2 \tag{1.4}$$

where $||\mathbf{x}||_2 := \sqrt{\sum_i x[i]^2}$. It is called *mean squared error* when averaged over all the input-output pairs.

**Loss for classification** In the context of a binary classification problem, we consider a scalar $y \in \{0, 1\}$ and a function $f : \mathbb{R}^d \rightarrow [0, 1]$. A common loss function to address this problem is the *binary cross-entropy* such that:

$$l(f(\mathbf{x}), y) := -\Big[ylog(f(\mathbf{x})) + (1 - y)log(1 - f(\mathbf{x}))\Big] \tag{1.5}$$

A sum of $t$ *binary cross-entropy* losses can be used to address the multi-label classification problem of $t$ classes.

In the context of a multi-class classification problem of $t$ classes, we consider a t-dimension one hot vector $y$ and a function $f : \mathbb{R}^d \rightarrow \mathbb{R}^t$. A common loss function to address this problem is the *cross-entropy* such that:

$$l(f(\mathbf{x}), \mathbf{y}) := -\sum_{k=i}^{t} y[k]log(\frac{exp(x[k])}{\sum_j exp(x[j])}) \tag{1.6}$$

**Linear models** We can restrict our function class to be linear parametric functions such that $f(\mathbf{x}) = \boldsymbol{W}^T\mathbf{x} + \mathbf{b}$ with $\boldsymbol{W} \in \mathbb{R}^{d \times t}$ and $\mathbf{b} \in \mathbb{R}^t$. A common regularization term for these models is the $L_2$ norm of $\boldsymbol{W}$, such that we optimize the following objective:

$$f^* = \text{argmin}_{f \in \mathrm{F}}\Big\{\frac{1}{n}\sum_{i=1}^{n} l(\boldsymbol{W}^T\mathbf{x}_i + \mathbf{b} - \mathbf{y}_i), \mathbf{y}_i) + \lambda||\boldsymbol{W}||_2^2\Big\} \tag{1.7}$$

where $\lambda \in \mathbb{R}$ is a hyperparameter controlling the amount of regularization. Interestingly, different instantiations of linear models have been used to model a single biological neuron such as the McCulloch&Pitts model (McCulloch et al. 1943), the Perceptron (Rosenblatt 1958) or the Adaline (Widrow et al. 1960).

**Kernel methods** This combination of linear models and L2 regularization has been widely studied as linear Support Vector Machine (SVM) (Cortes et al. 1995).

In the context of a binary classification problem ($t = 1$), it is learned using a hinge loss such as:

$$f^* = \text{argmin}_{f \in \mathrm{F}} \Big\{ \frac{1}{n} \sum_{i=1}^{n} max(0, 1 - y_i \boldsymbol{W}^T \mathbf{x}_i + \mathbf{b}) + \lambda ||\boldsymbol{W}||_2^2 \Big\} \tag{1.8}$$

where $y_i \in \{-1, 1\}$ depending on the class. This quadratic optimization problem is known as the primal problem.

The SVM can be extended to perform a non-linear mapping $\phi$ of the input vector into a higher dimensional Hilbert space such as:

$$f(\mathbf{x}) = \boldsymbol{W}^T \phi(\mathbf{x}) + \mathbf{b} \tag{1.9}$$

The mapping can also be defined as a kernel function in the dual formulation of the SVM. This class of functions is used to increase the complexity of the functions that can be learned.

**Neural networks** Another class of functions that leads to the learning of more complex functions is the neural networks. These functions are the combination of several stacked linear models, which are separated with non-linear activation functions $\sigma$. They are used to learn a non-linear mapping from input to output spaces. For instance, a neural network made of one hidden layer of dimension $h$ can be defined as:

$$f(\mathbf{x}) = \boldsymbol{W}_2^T \sigma(\boldsymbol{W}_1^T \phi(\mathbf{x}) + \mathbf{b_1}) + \mathbf{b_2} \tag{1.10}$$

where $\boldsymbol{W}_1 \in \mathbb{R}^{d \times h}$, $\mathbf{b_2} \in \mathbb{R}^h$, $\boldsymbol{W}_2 \in \mathbb{R}^{h \times t}$ and $\mathbf{b_2} \in \mathbb{R}^t$. The most used activation functions are the sigmoid, the hyperbolic tangent and the Rectified Linear Unit (ReLU) (Krizhevsky et al. 2012). This family of functions is also called *multi-layer perceptron (MLP)*, *feed-forward network*, or *fully connected layers*.

### 1.2.3 Training

Once the choice of a class of functions $F$, a loss function $l$ and a regularization term has been made, the problem of learning can be reduced to an optimization problem of the form $\theta^* = argmin_\theta g(\theta)$, where $f$ is a parametric function, $\theta$ is a parameter vector and $g$ is an objective function defined as:

$$g(\theta) = \sum_{i=1}^{n} l(f_\theta(x_i), y_i) + r(f_\theta) \tag{1.11}$$

Numerous methods exist to minimize this objective. Many can be found in the book by Nocedal et al. 2006. Here, we focus on the most used method to minimize the objective when $f$ is a neural network.

**Gradient based optimization** First order derivative based convex optimization methods are commonly used to optimize $\theta$. These methods require to be able to calculate the gradient $\nabla_\theta g$ in order to use it as a search direction in the parameters space. The gradient descent algorithm is then applied to minimize $g(\theta)$. It consists in alternating between a gradient evaluation step and a small parameters update step until a stopping criterion is met. A common criterion is an arbitrary large enough number of update steps $s$ to let the algorithm converge. As defined in algorithm 1.1, its stochastic version is used in practice. It consists in estimating the gradient on a subset of the dataset to reduce memory consumption and computation cost. Even in the context of neural networks, which model highly non-convex functions, these convex optimization methods are used to reach local minima of surprising generalization abilities.

The backpropagation algorithm (Rumelhart et al. 1986; LeCun et al. 1998) based on the chain rule is commonly used with neural networks to calculate $\nabla_\theta g$, Also, different architecture-dependent approaches can be found in the literature to initialize their parameters $\theta$ (Glorot et al. 2010; Ilya Sutskever et al. 2013; He et al. 2015b).

**Algorithm 1.1** Stochastic Gradient Descent

```
input: a training dataset of $n$ input-output pairs $(x, y) \in X \times Y$
input: an initialized vector of parameters $\theta$
input: an objective function $g(\theta)$ to minimize
input: a batch size $b$
input: a learning rate $\eta$
repeat
    1. randomly sample a mini-batch of $b$ input-output pairs
    2. estimate the gradient $\nabla_\theta g$ on the mini-batch using backpropagation
    3. compute the update direction: $\delta := -\eta \nabla_\theta g$
    4. update the parameters: $\theta := \theta + \delta$
until the stopping criterion is met;
```

**Advanced update rules** It is often possible to reach faster convergence by relying on advanced update rules. The *momentum* approach (Polyak 1964; Ilya Sutskever et al. 2013) takes the previous directions into account to encourage progress along consistent directions of the gradient. It relies on a memory vector $\mathbf{v}$ of the same dimension as $\theta$:

$$\begin{aligned} \mathbf{v} &:= \mu \mathbf{v} + \eta \nabla_\theta g \\ \delta &:= -\mathbf{v} \end{aligned} \tag{1.12}$$

where the scalar $\mu \in [0, 1]$ is the momentum coefficient. The Nesterov's Accelerated Gradient (NAG) approach (Nesterov 1964; Ilya Sutskever et al. 2013) speed up *momentum* by evaluating the gradient at the next update step such as:

$$\mathbf{v} := \mu\mathbf{v} + \eta\nabla_\theta g(\theta + \mu\mathbf{v}) \tag{1.13}$$

Other update rules work well on neural networks such as Adagrad (Duchi et al. 2011), RMSProp (Tieleman et al. 2012) or Adam (Kingma et al. 2014). Additionally, scheduling rules that lower the learning rate $\eta$ along the optimization process are often used in practice.

**Hyperparameters tuning** These methods propose efficient solutions to optimize the parameters $\theta$. However, gradients of parameters such as the learning rate $\eta$, batch size $b$, number of steps $s$, amount of regularization $\lambda$ or hidden size $h$, can not be easily calculated and thus optimized by gradient descent. To optimize these *hyperparameters*, we often rely on cross-validation methods. A common method consists in evaluating the parametric function $f$ on a held-out split of the training set, which is called the *validation set*, in order to select the best set of hyperparameters.

### 1.2.4 Neural networks for computer vision

Several classes of functions have been efficiently used to model computer vision data such as images. A common class is the Convolutional Neural Network (ConvNet). These functions take advantage of the spatial information redundancy throughout the image to share parameters between neurons. ConvNet is a bio-inspired model first introduced in 1980 (Fukushima 1980) and trained with backpropagation in 1989 (LeCun et al. 1989). We review its more recent versions that we use throughout this PhD thesis. Some of these approaches led to significant improvements on a wide array of CV tasks (Oquab et al. 2014) such as image classification (Krizhevsky et al. 2012), semantic segmentation (Long et al. 2015) or object detection (Girshick et al. 2014).

**Deep Convolutional Neural Networks** As illustrated in Figure 1.4, ConvNet are made of convolutional layers in place of the linear layers. They commonly encompass methods such as spatial pooling to progressively reduce the spatial dimension until the final output prediction. Their deep sequence of layers is used to learn hierarchical abstractions of visual concepts in a *end-to-end* manner. That is, all of their parameters are optimized at the same time by minimizing a common objective. Contrarily to handcrafted methods, which rely on well-defined features extraction steps, *end-to-end* training allows learning richer features directly from the raw pixels. Features extracted from layers closer to the input image represent low-level concepts such as shapes, colors or textures, whereas features closer

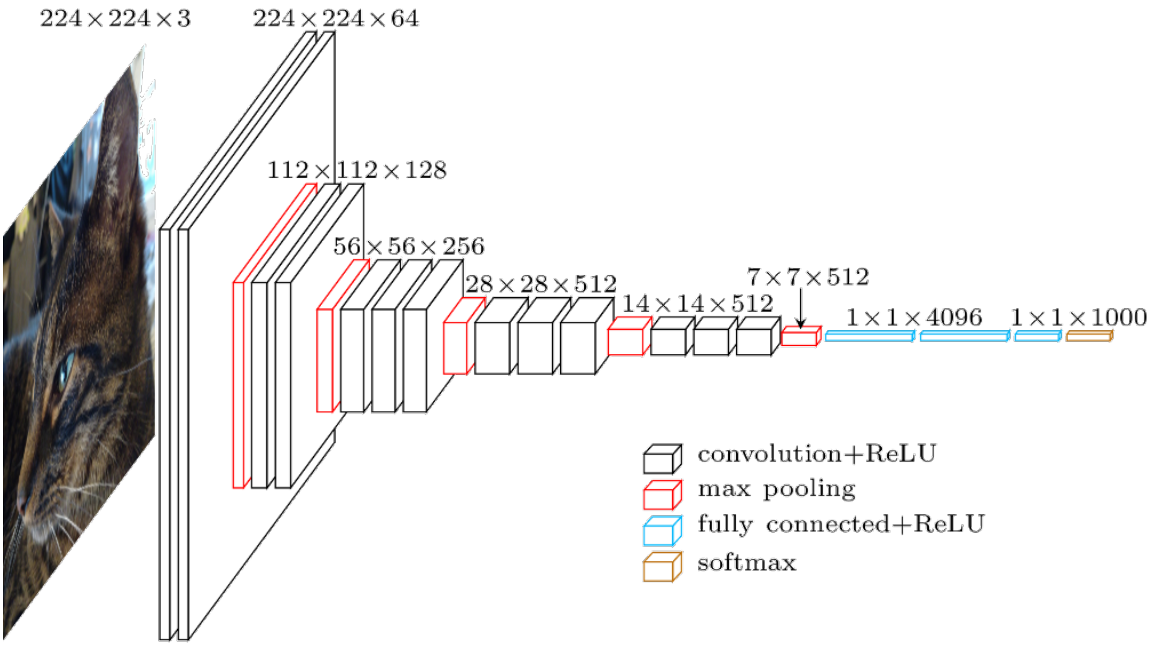


Figure 1.4 – Illustration of VGG-16 (Simonyan et al. 2015), a common Deep ConvNet. Illustration taken from Durand 2017.

to the output predictions represent high-level concepts such as classes, objects or object-parts. A deeper sequence of layers allows learning finer hierarchical abstractions. However, training Deep ConvNet has long been a challenge due to the gradient vanishing and exploding problems (Hochreiter 1991; Hochreiter et al. 2001). The ReLU activation function (Krizhevsky et al. 2012) and the residual connections between layers (He et al. 2016) have been proposed to address these issues.

**Pretrained Convolutional Neural Networks** ConvNet pretrained on large-scale datasets can be used to extract visual features on other datasets of variable size. Then, different models can be trained on top of these visual representations. Importantly, they can be further adapted to the new task by *fine-tuning* (Donahue et al. 2014) the pretrained ConvNet. This strategy consists in backpropagating the loss computed from the new task objective function to the pretrained parameters in order to optimize them at the same time. This *fine-tuning* method often leads to significant gains, in particular when the domains of both datasets are semantically far. Commonly used ConvNet pretrained on the ImageNet dataset (Russakovsky et al. 2015b) are AlexNet (Krizhevsky et al. 2012), VGG16 (Simonyan et al. 2015) and ResNet-152 (He et al. 2016). See Cadene 2017, for a more exhaustive list of pretrained ConvNet.

**Fully Convolutional Neural Network (FCN)** ConvNet are designed to take a fixed size image as input and output a vector representations. However, they can

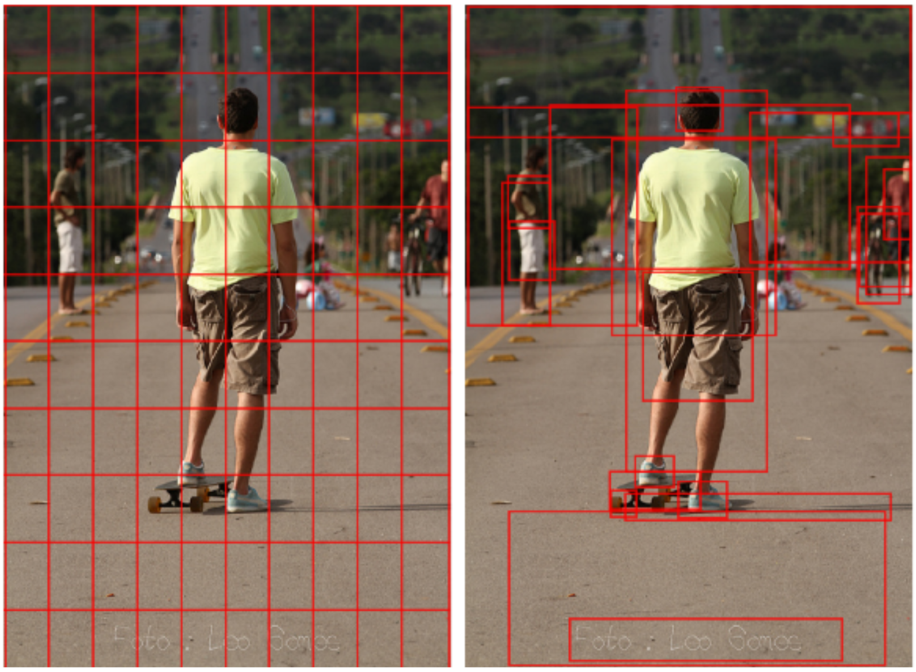


Figure 1.5 – Illustration of two region-based ConvNet methods. On the left, the fixed-grid over the image illustrates the dense features extraction of Fully Convolutional Network (FCN). On the right, each object bounding boxes are detected and their corresponding features extracted by a Region-based Convolutional Neural Network (R-CNN). Illustration from (Anderson et al. 2018).

be transformed into FCN to take wider images of variable sizes. As illustrated in Figure 1.5, ConvNet can be used as a convolutional operation over a wider image to output grid-like representations $\boldsymbol{h}_{fcn} \in \mathbb{R}^{(h*w) \times d}$, with $h$ the height of the grid, $w$ the width of the grid and $d$ the dimensions of the features. For instance, for a grid size of $14 \times 14$, the number of considered regions can reach 196. Pretrained ConvNet for image classification are commonly transformed to a FCN by replacing their linear layers by $1 \times 1$ convolutional layers. Then, they can be used in different contexts such as weakly supervised learning for object detection (Oquab et al. 2015; Durand et al. 2016; Durand et al. 2017) or image segmentation (Long et al. 2015).

**Region-based Convolutional Neural Networks (R-CNN)** As illustrated in Figure 1.5, ConvNet can also be embedded into a R-CNN architecture (Girshick et al. 2014; Girshick 2015) to output object-based representations $\boldsymbol{h}_{rcnn} \in \mathbb{R}^{n \times d}$ with $n$ the number of detected objects and $d$ the dimensions of the features. They are trained on specially tailored datasets to detect bounding boxes around objects, predict their classes and sometimes attributes. A common approach for object detection is Faster-RCNN (S. Ren et al. 2015). First, a FCN is used to extract grid-like representations from the image. A Region of Interest (RoI) pooling is applied on top to calculate features associated with default bounding boxes in the image. These default boxes are called *anchors*. Then, a Non-Maximum Suppression (NMS) with Intersection over Union (IoU) threshold is applied to keep the top boxes and their corresponding features. Secondly, a two heads ConvNet is used to classify each box proposal and refine their coordinates. Finally, a class-dependent NMS with IoU threshold is applied on the detected objects. Extensions of R-CNN have been developped for simultaneous segmentation (He et al. 2017), or for faster computation (W. Liu et al. 2016; Redmon et al. 2017).

### 1.2.5 Neural networks for language processing

Similarly to visual data, several classes of functions have been used to model textual data of variable length. They may rely on different atomic elements such as characters, sub-word units, words, bi-grams of words, or even tri-grams of words. Here, we focus on common classes of functions and training methods used in the context of this PhD thesis.

**Word embeddings** A common method for processing language consists in associating a vector representations $\mathbf{x} \in \mathbb{R}^d$ to each word of a given dictionary, where typical values of $d$ lie between 50 and 1000. These word embeddings may be randomly initialized and optimized for an end task, or pretrained using methods such as Word2Vec (Tomas Mikolov et al. 2013b) or Glove (Pennington et al. 2014). These methods optimize the word embeddings to reconstruct the linguistic contexts in which each word appears. With respect to the corpora on which they are trained on, different word properties emerge during the optimization process. For instance, words sharing the same linguistic contexts possess similar representations. It becomes also possible to infer semantic relationships by using basic mathematical operations on the word vector representations. For instance, the addition of "Germany" and "capital" is close to "Berlin" in the embedding space provided by Word2Vec. Word embeddings can be used to represent sentences of a variable number of words by calculating the average. In this case, word orders are not taken into account.

**Recurrent Neural Networks (RNN)** A family of neural architectures designed for modeling sequences is the Recurrent Neural Network (RNN) (Elman 1990). They can be used to extract sentence representations (Tomáš Mikolov et al. 2010). Given a sequence of word representations $[\mathbf{x}_1, ..., \mathbf{x}_T]$, the RNN updates its internal hidden state $\mathbf{h}$ such as:

$$\mathbf{h}_t = f(\mathbf{h}_{t-1}, \mathbf{x}_t) \tag{1.14}$$

where $\mathbf{h}_T$ corresponds to the vector representations of the sentence. A common linear instantiation of $f$, also known as *vanilla*, is defined such as:

$$\mathbf{h}_t = tanh(\boldsymbol{W}_h^T \mathbf{h}_{t-1} + \boldsymbol{W}_x^T \mathbf{x}_t + \mathbf{b}) \tag{1.15}$$

However, this *vanilla* RNN may exhibit vanishing and exploding gradients problem over large sequences (Hochreiter et al. 2001) such as sentences. The Long-Short Term Memory (LSTM) (Hochreiter et al. 1997) has been proposed to address these issues. It is composed of three gating operations, which output a *remember* state vector $\mathbf{r}_t$, a *save* state vector $\mathbf{s}_t$ and a *forget* state vector $\mathbf{f}_t$, such as:

$$\begin{aligned} \mathbf{r}_t &= \sigma(\boldsymbol{W}_{hr}^T \mathbf{h}_{t-1} + \boldsymbol{W}_{xr}^T \mathbf{x}_t + \mathbf{b}_r) \\ \mathbf{s}_t &= \sigma(\boldsymbol{W}_{hs}^T \mathbf{h}_{t-1} + \boldsymbol{W}_{xs}^T \mathbf{x}_t + \mathbf{b}_s) \\ \mathbf{f}_t &= \sigma(\boldsymbol{W}_{hf}^T \mathbf{h}_{t-1} + \boldsymbol{W}_{xf}^T \mathbf{x}_t + \mathbf{b}_f) \end{aligned} \tag{1.16}$$

where $\mathbf{h}_t \in \mathbb{R}^{d_h}$, $\mathbf{x}_t \in \mathbb{R}^{d_x}$, and $\sigma$ is the sigmoid operation. It is also composed of three different memory states. The *internal memory* $\mathbf{m}_t$ is defined such as:

$$\mathbf{m}_t = tanh(\boldsymbol{W}_{hm}^T \mathbf{h}_{t-1} + \boldsymbol{W}_{xm}^T \mathbf{x}_t + \mathbf{b}_m) \tag{1.17}$$

It is used to update the *long-term memory* $\mathbf{l}_t$ using the *remember* and *save* states such as:

$$\mathbf{l}_t = \mathbf{r}_t \odot \mathbf{l}_{t-1} + \mathbf{s}_t \odot \mathbf{m}_t \tag{1.18}$$

Finally, the *forget* state is used to update the *working memory* $\mathbf{h}_t$ such as:

$$\mathbf{h}_t = \mathbf{f}_t \odot tanh(\mathbf{l}_t) \tag{1.19}$$

Another commonly used instantiation of RNN, which addresses the problems of vanishing and exploding gradient is the Gated Recurrent Unit (GRU) (Chung et al. 2014). It simplifies the LSTM architecture while reaching similar performances. It is defined such as:

$$\begin{aligned} \mathbf{f}_t &= \sigma(\boldsymbol{W}_{hf}^T \mathbf{h}_{t-1} + \boldsymbol{W}_{xf}^T \mathbf{x}_t + \mathbf{b}_f) \\ \mathbf{s}_t &= \sigma(\boldsymbol{W}_{hs}^T \mathbf{h}_{t-1} + \boldsymbol{W}_{xs}^T \mathbf{x}_t + \mathbf{b}_s) \\ \mathbf{m}_t &= tanh(\mathbf{f}_t \odot \boldsymbol{W}_{hm}^T \mathbf{h}_{t-1} + \boldsymbol{W}_{xm}^T \mathbf{x}_t + \mathbf{b}_m) \\ \mathbf{h}_t &= \mathbf{s}_t \odot \mathbf{h}_{t-1} + (1 - \mathbf{s}_t)\mathbf{m}_t) \end{aligned} \tag{1.20}$$

where $\mathbf{h}_t \in \mathbb{R}^{d_h}$ and $\mathbf{x}_t \in \mathbb{R}^{d_x}$.

Similarly to word embeddings, pretraining methods can be applied to RNN in order to extract rich sentence representations. In particular, the *skip-thought* method (Kiros et al. 2015) optimizes the RNN parameters to reconstruct the linguistic contexts in which each sentence appears. As illustrated in Figure 1.6, this method consists in computing the sentence representation through a RNN encoder, and to feed it to a RNN decoder to recover the previous sentence and the next sentence occurring in the text.

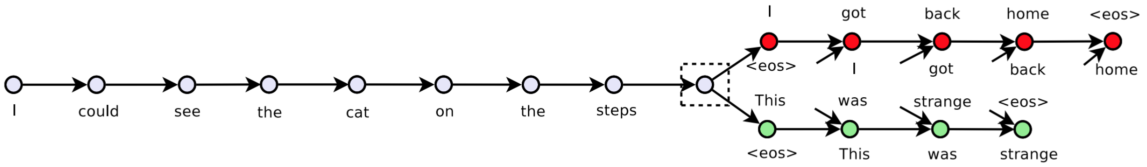


Figure 1.6 – Skip-thought method (Kiros et al. 2015) to pretrain RNN for sentence modeling. The grey, red and green dots correspond respectively to the word embeddings of the current, next and previous sentence. The arrow illustrates the input-output flow from the RNN.

## 1.3 Challenges

The recent progress in deep learning gave rise to a lot of promises. However, many challenges need to be overcome before being able to bridge the gap between vision and language.

**Data representations** An image is made of pixels and can not be easily compared with a sentence made of words. Our first challenge consists in producing rich, robust and comparable representations of the data for each modality. A rich representation would allow identifying all the useful semantic concepts to solve a given multimodal task. A robust representation should vary the least when the same semantic concept possess a high variability in the raw data space. For the image modality, this variability may be due to different kind of noises, difference of viewpoints, illumination changes, various appearance of objects, object occlusions, etc. For the textual modality, this variablity may be due to spelling mistakes, synonyms, style variations, etc. Deep learning methods propose to learn a mapping from the input space to a vector space where each modality can be compared. However, learning these representations in an end-to-end manner requires to collect large enough task-specific datasets. Pretrained models are commonly used to reduce this constraint. They are able to produce representations that already possess some desired properties. A lot of questions still remain regarding how to pretrain them and how to use them.

**Multimodal representations and fusion** A second challenge consists in constructing multimodal representations that contain meaningful combinations and structural relationships between the visual and language concepts in order to solve a given task. Some tasks such as image captioning require to align the visual and language concepts. Some other tasks such as visual question answering require to fuse the visual and language concepts. In both cases, the difficulty lies in the exploitation of the complementary and redundancy of the information in the context of modality-dependent representations. For instance, an image and a sentence can be represented as vectors of different dimensionality. In the case where both have the same dimensionality, each dimension may represent different concepts. They are not aligned. Also, both representations may contain different information. For instance, in visual question answering, the image representation may contain information about the global visual scene, whereas the question may only focus on a small part of the scene. Some visual information may need to be discarded before constructing the multimodal representations that will be used to answer the question.

**Learning issues** A third more practical challenge involves the development of methods to train and to use multimodal models. First, large scale datasets need to be collected following protocols that ensure the quality of the data and their annotations. In fact, the raw data can be noisy, unclean, incomplete, unbalanced, and the annotations can be weak or even wrong. Secondly, powerful hardware and software infrastructures must be used to store and load the data, and to train and evaluate the models. Finally, different hypotheses must be validated to ensure that these models did not learn behaviors that exploit unwanted statistical regularities or biases.

## 1.4 Contributions

This dissertation is about the development of intelligent systems capable of processing visual and textual data. We focus our efforts on recent large-scale challenging tasks that aim at creating links between concepts from the visual and textual modalities. We propose several deep multimodal learning approaches to tackle core problems related to these multimodal tasks.

**Multimodal alignment** The crossmodal retrieval task explicitly aims at creating links between modalities. Its goal is to retrieve items of interest belonging to one modality using a query belonging to the other modality. For instance, this task may consist in retrieving sentences that better describe a given image, or reciprocally, images that better illustrate a given sentence. Approaches from metric learning are commonly used to tackle this task. They consist in learning a crossmodal similarity function to rank data of the opposite modality. It involves aligning matching image-text data in a shared representation space. Nevertheless, it is not clear how to best align these multimodal representations and how to structure the retrieval space using additional semantic information.

We address these questions in Chapter 2 by proposing a new alignment approach, which can leverage additional semantic information about the image-text pairs. Additionally, we introduce an adaptive strategy to select image-text pairs that will contribute the most in the crossmodal alignment. A more practical contribution is the creation of an experimental framework for large scale training that we used throughout this thesis.

The work presented in Chapter 2 has led to the publication of a conference paper and a workshop paper at equal contribution with Micael Carvalho:

- Micael Carvalho*, Remi Cadene*, David Picard, Laure Soulier, Nicolas Thome, and Matthieu Cord (2018b). "Cross-modal retrieval in the cooking context: Learning semantic text-image embeddings". In: *ACM Conference on Research and Development in Information Retrieval (SIGIR)*. URL: https://arxiv.org/abs/1804.11146

- Micael Carvalho*, Remi Cadene*, David Picard, Laure Soulier, and Matthieu Cord (2018a). "Images & Recipes: Retrieval in the cooking context". In: *Proceedings of the IEEE International Conference on Data Engineering (ICDE). Data Engineering meets Intelligent Food and Cooking Recipe Workshop (DECOR)*. URL: https://arxiv.org/abs/1805.00900

**Multimodal fusion and reasoning** We go one step further towards creating links between modalities by tackling the VQA task. It consists in answering a question about the visual content of an image. It requires textual grounding and visual reasoning abilities. To solve this task, we need to know how to fuse both input modalities. In Chapter 3, we propose a theoretically grounded fusion framework based on bilinear models. Our framework allows expressing several fusion modules from the literature. We leverage it to propose novel learnable, efficient and powerful fusion modules. They model fine and rich interactions between the image and the question while maintaining a tractable number of free parameters.

In the line of previous works, we embed our fusion modules in a state-of-the-art VQA architecture. The latter acts as an inductive bias constraining the VQA model to focus its attention on visual regions that are useful to answer the question. To advance towards multimodal reasoning, we propose a novel reasoning architecture that goes beyond the classical attentional framework. Our approach consists in fusing object-based visual representations with the question while adding contextual information about the pairwise relationships between each region. It mimics a multistep reasoning process over a graphical representation of the visual scene.

The work presented in Chapter 3 has led to the publication of two conference papers and two workshop papers at equal contribution with Hedi Ben-Younes, and one conference paper as a second author contribution:

- Hedi Ben-Younes*, Rémi Cadène*, Nicolas Thome, and Matthieu Cord (2017b). "MUTAN: Multimodal Tucker Fusion for Visual Question Answering". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. URL: https://arxiv.org/abs/1705.06676

- Hedi Ben-Younes, Rémi Cadène, 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 (AAAI)*. URL: https://arxiv.org/abs/1902.00038

- Rémi Cadène*, Hedi Ben-Younes*, Nicolas Thome, and Matthieu Cord (2019). "MUREL: Multimodal Relational Reasoning for Visual Question Answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. URL: https://arxiv.org/abs/1902.09487

- Hedi Ben-Younes*, Remi Cadene*, Nicolas Thome, and Matthieu Cord (2017a). "VQA Challenge Workshop: MUTAN 2.0". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). VQA Challenge and Visual Dialog Workshop*

- Hedi Ben-Younes*, Remi Cadene*, Nicolas Thome, and Matthieu Cord (2018). "VQA Challenge Workshop: Bilinear Superdiagonal Fusion". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). VQA Challenge and Visual Dialog Workshop*

**Unimodal biases** An important aspect of multimodal learning is the fact that statistical models can leverage a specific kind of biases. Due to the heterogeneity of the input data, they tend to give too much importance to some predictive features from one modality. In Chapter 4, we address this problem in the context of the VQA task where models overrely on the textual modality. We propose a novel strategy to reduce unimodal biases learned during training. Our strategy is based on a text-only model that captures the language biases. This unimodal model identifies data that contribute to learning these biases during training. When a given question can be answered without looking at the image, we dynamically adjust the loss in order to compensate for biases.

The work presented in Chapter 4 has led to the publication of one conference paper at equal contribution with Corentin Dancette:

- Remi Cadene*, Corentin Dancette*, Hedi Ben-Younes, Matthieu Cord, and Devi Parikh (2019). "RUBi: Reducing Unimodal Biases for Visual Question Answering". In: *Advances in Neural Information Processing Systems (NeurIPS)*. URL: https://arxiv.org/abs/1906.10169

Chapter 2

# MULTIMODAL ALIGNMENT FOR IMAGE-TEXT RETRIEVAL

## Contents

### *Chapter abstract*

*We tackle the task of crossmodal retrieval for images and texts. Our target application consists in a large-scale search engine for cooking recipes. Its main purposes are to retrieve the most similar recipes given a picture of a dish, or the most illustrative pictures given a textual recipe. We propose AdaMine to align the two modalities in the same representation space. Our main contributions are two folds. First, we propose an adaptive learning scheme providing consistent gradient information during training. Secondly, we introduce a triplet-based auxiliary loss to structure the retrieval space with additional semantic information. We validate our approach on Recipe1M, a dataset of nearly 1 million pictures of dishes and their recipes. We show the effectiveness of AdaMine regarding previous state-of-the-art models and present qualitative results.*

*The work in this chapter, at equal contribution with Micael Carvalho, has led to the publication of a conference paper and a workshop paper:*

- Micael Carvalho*, Remi Cadene*, David Picard, Laure Soulier, Nicolas Thome, and Matthieu Cord (2018b). "Cross-modal retrieval in the cooking context: Learning semantic text-image embeddings". In: *ACM Conference on Research and Development in Information Retrieval (SIGIR)*. URL: https://arxiv.org/abs/1804.11146

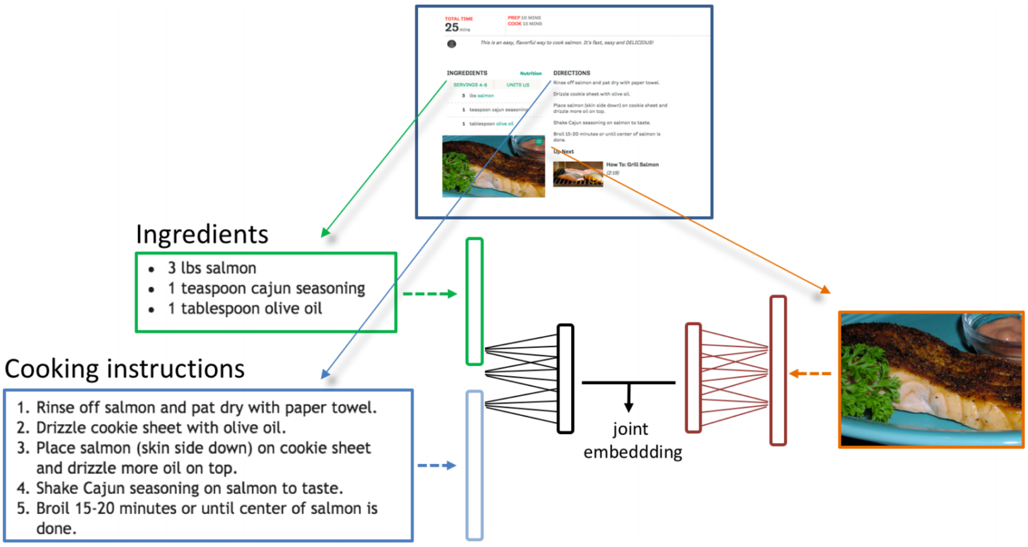
- Micael Carvalho*, Remi Cadene*, David Picard, Laure Soulier, and Matthieu Cord (2018a). "Images & Recipes: Retrieval in the cooking context". In: *Proceedings of the IEEE International Conference on Data Engineering (ICDE). Data Engineering meets Intelligent Food and Cooking Recipe Workshop (DECOR)*. URL: https://arxiv.org/abs/1805.00900

## 2.1 Introduction

As explained in Chapter 1, a core problem when dealing with text and image data is to give the ability to a system to connect concepts from both modalities. Such a model needs to break down these two complex and heterogeneous high dimensional objects to grasp their semantic meaning and to comprehend how both modalities can be related. This ability is a prerequisite for a lot of multimodal tasks such as searching images on the internet using natural language, linking words to objects in photo collections or providing captions to images (Karpathy et al. 2015; Lazaridou et al. 2015).

Due to the growing interest of home-made food and social media platforms (Sanjo et al. 2017), massive amounts of cooking-related image and text data have recently been created. Consequently, novel applications arose such as ingredient classification (J. Chen et al. 2016) or recipe classification (X. Wang et al. 2015; Bossard et al. 2014). To go one step further, we focus on the recipe recognition task through a cross-modal retrieval framework illustrated in Figure 2.1. The goal is to retrieve the cooking recipe associated with a given image, or conversely, to retrieve the image that better illustrates a given cooking recipe. We tackle this task on Recipe1M (Salvador et al. 2017) which is one of the largest datasets including both English cooking recipes with their ingredients and instructions, images and recipe categories.

Deep learning approaches are able to tackle the cross-modal retrieval task by creating a same latent space for both modalities (Bossard et al. 2014; Kawano et al. 2014; Kiros et al. 2014; Karpathy et al. 2015; Salvador et al. 2017; J. Chen et al. 2017). This shared space is organized so that data with similar meanings are represented similarly. Text and image data can be projected in this space to assess their similarity. Then, a text query or image query can be used to retrieve the most similar recipes of the other modality. A common approach to organize this space consists in aligning text and image data through a surrogate optimization of a ranking problem. Loss functions, such as the pairwise loss or the triplet loss,



Ingredients

- 3 lbs salmon
- 1 teaspoon cajun seasoning
- 1 tablespoon olive oil

Cooking instructions

1. Rinse off salmon and pat dry with paper towel.
2. Drizzle cookie sheet with olive oil.
3. Place salmon (skin side down) on cookie sheet and drizzle more oil on top.
4. Shake Cajun seasoning on salmon to taste.
5. Broil 15-20 minutes or until center of salmon is done.

Figure 2.1 – Cross-modal learning framework for recipe recognition. By aligning recipe-image pairs collected from online resources, it allows retrieving the cooking recipe associated with a given picture of a dish. Contrarily, it allows retreiving images illustrating a recipe.

can be used to minimize the distance between matching text-image pairs by some margin and maximize the distance between non-matching pairs by some margin.

A first difficulty of applying these alignment methods on Recipe1M is to sample the matching and non-matching text-image pairs to optimize. To ensure fast convergence on this large-scale dataset, it is crucial to select those that violate the distance constraint so that the loss stays high enough during the training process. However, only selecting those that violate the constraint the most might lead to poor generalization, as mislabelled and outliers might dominate (Schroff et al. 2015; Faghri et al. 2018). Doing the appropriate sampling is still an open question. A second difficulty is to semantically organize the multimodal space to ensure optimal generalization. Notably, Salvador et al. 2017 highlight that a simple alignment can lead to poor retrieval performances on the Recip1M dataset. Their approach involves optimizing an auxiliary task of recipe type classification. To do so, they add an extra layer on top of the multimodal output of the neural network. However, the classifier adds many parameters that are discarded at the end of the training.

In Section 2.2, we review different learning strategies and architectures from the state-of-the-art to tackle the problem of crossmodal retrieval. In Section 2.3, we introduce AdaMine, a learning strategy embedded into a crossmodal framework to align the text and image modalities in a same retrieval space. Contrarily to previous works, our alignment strategy is based on four different crossmodal triplet losses. Two of them take advantage of the additional semantic information in the form of the recipe categories to better structure the retrieval space. Additionally, we propose a novel mining strategy which adaptively adjusts the contribution of each loss to reduce gradient vanishing. In Section 2.4, we present several experimental results to validate our learning approach. We also show examples of applications that exploit the potential of our crossmodal framework in the computational cooking context. A live demo of our large-scale search engine for cooking recipes is available on the VIsual Seek for Interactive Image Retrieval (VISIIR) project web page:

- visiir.lip6.fr

## 2.2 Related work

Deep learning approaches provide practical and powerful ways to tackle crossmodal retrieval tasks for text and image data. Pretrained deep neural networks can be used to extract rich and transferable representations from images (Krizhevsky et al. 2012; Sermanet et al. 2014; Simonyan et al. 2015; He et al. 2016) as well as from words (Tomas Mikolov et al. 2013b; Pennington et al. 2014) and sentences (Chung et al. 2014; Kiros et al. 2015; Vaswani et al. 2017; Devlin et al. 2018). Then, different strategies can be used to learn a crossmodal similarity function which is critical to perform retrieval across modalities. A common strategy consists in aligning representations of both modalities in a shared latent space where simple distance functions such as the euclidean distance can be used to retrieve the most similar data. In the following, we review these learning strategies.

### 2.2.1 Learning crossmodal alignments

**Unsupervised strategies** The first category of works learns to align the two modal spaces without requiring any annotations. For instance, the well-known Canonical Correlation Analysis (CCA) (Hotelling 1936) relies on the correlation between the representations of two sets. In our case, a first set corresponds to the textual recipes and a second to the images. Any modification in the ordering of the vector representations inside the sets does not affect the results. CCA projects both sets into a lower-dimensional space in which their linear correlation is maximized. However, CCA will fail to align both sets if the correlations between them are non-linear. In this case, its non-linear variations can be used. For instance, Kernel-CCA (Lai et al. 2000; Bach et al. 2002) substitutes the inner product by kernel functions. Deep-CCA (Andrew et al. 2013) learns unimodal non-linear projections. To more specifically align text and image data, L. Wang et al. 2016; Yan et al. 2015 exploit Deep-CCA and propose ways to reduce the overfitting and optimization complexity. CCA-based methods can also be used as regularization constraints helping to organize the crossmodal retrieval space Eisenschtat et al. 2017.

**Supervised strategies** The second category of works learns to align the two modal spaces by solving a ranking problem. They are more efficient but necessitate relevance annotations. Formally, let consider a query item $x_q$ such as a textual recipe, its set of relevant items $P_q = \{x_p\}$ such as a few images that illustrate this recipe, and its set of irrelevant items $P_n = \{x_n\}$ such as all the images that do not depict this recipe. Then, solving the ranking problem implies to find the distance function $d$ such that:

$$\forall x_q, \forall x_p, \forall x_n, d(x_q, x_p) < d(x_q, x_n) \tag{2.1}$$

Different surrogate loss functions can be used to measure the cost of violating each of these inequalities. Xing et al. 2003; Hadsell et al. 2006; Salvador et al. 2017 consider a *pair-wise loss* function to minimize a simple distance function on the shared multimodal space between pairs of matching objects and maximize the distance between pairs of non-matching objects. When the euclidean distance is used, this loss is also called *squared loss*. When the cosine similarity is used, this loss is also called *cosine similarity loss*. A margin hyperparameter $\alpha_{neg}$ controls the force of the constraint on the distance. It is often needed to avoid overfitting by forcing two matching objects to have the exact same representation on the latent space. More formally, the *pair-wise loss* can be defined as:

$$\begin{aligned}\mathcal{L}_{pwc}(\theta, x_q, x) = \; & y\big[d_\theta(x_q, x)\big]_+ \\ & + (1-y)\big[\alpha_{neg} - d_\theta(x_q, x)\big]_+\end{aligned} \tag{2.2}$$

where $x_q$ is the query item, $y = 1$ when $x$ is a relevant item, $y = 0$ when $x$ is an irrelevant item, and $\big[x\big]_+ := max(0, x)$. J. Hu et al. 2014 introduces an additional positive margin to further reduce overfitting, such as:

$$\begin{aligned}\mathcal{L}_{pwc++}(\theta, x_q, x) = \; & y\big[d_\theta(x_q, x) - \alpha_{pos}\big]_+ \\ & + (1-y)\big[\alpha_{neg} - d_\theta(x_q, x)\big]_+\end{aligned} \tag{2.3}$$

Weinberger et al. 2009; Chechik et al. 2010; Schroff et al. 2015; Karpathy et al. 2015; Faghri et al. 2018; Kiros et al. 2014 consider a triplet-based loss function, also called *ranking loss*, which is a more natural surrogate of the ranking inequalities constraints. In this case, a loss is computed only if the distance between the query object and the relevant object is larger than the distance between the query and the irrelevant item. More formally, it can be defined as:

$$\mathcal{L}_{triplet}(\theta, x_q, x_p, x_n) = \big[d_\theta(x_q, x_p) + \alpha - d_\theta(x_q, x_n)\big]_+ \tag{2.4}$$

where $\alpha$ is a margin. This loss function can be thought as a pairwise loss that dynamically adjusts its margin according to the neighborhood of the query.

Other losses can be found in the literature and come with different benefits such as the magnet loss (Rippel et al. 2016), the quadruplet loss (Law et al. 2013) or the SoDeep loss (Engilberge et al. 2019).

**Image-text retrieval datasets** Three standard real scene datasets are commonly used to evaluate crossmodal image-text retrieval systems: Flickr8k (Hodosh et al. 2013), Flickr30k (Young et al. 2014), MSCOCO (Tsung-Yi Lin et al. 2014; X. Chen et al. 2015). These datasets contain around 8,000, 32,000 and 123,000 images respectively and each image is annotated with roughly 5 sentences using Amazon Mechanical Turk (AMT). After removing rare words, the average sentence length of Flickr30k is 10.5 words and the average sentence length of MSCOCO is 8.7 words.

Another type of retrieval dataset comes with parallel corpora without any annotation costs. Cooking recipe datasets from social media platforms possess textual information about the recipe which is illustrated by several images. In terms of cooking-related image-text datasets, Recipe1M dataset (Salvador et al. 2017) is currently the largest one in English. It contains approximately 1 million cooking-related image-text pairs, which is twice as many recipes as Kusmierczyk et al. 2016 and eight times as many images as J. Chen et al. 2016. There are also two unique aspects of the Recipe1M dataset. First, textual data about recipes are more structured than the standard text-image datasets (Hodosh et al. 2013; Young et al. 2014; X. Chen et al. 2015). Each recipe is composed of a set of ingredients and a list of instructions. Secondly, each image and recipe comes with an additional annotation type of the form of a cooking category. These specificities make the Recipe1M dataset suited for the development of different language models that can better take advantage of this recipe structure, as well as learning strategies that take advantage of the additional semantic information. Very recently, an extended version of this dataset has been released (Marin et al. 2019). It contains 13 million food images, but was not available at the time of submission.

### 2.2.2 Image-text alignment architectures

**Hand-crafted approaches** Early approaches that perform alignment of image-text data rely on hand-crafted features and unsupervised alignment strategies. For instance, Socher et al. 2010 use hand-crafted color, texture, position and shape features to represent images, co-occurrence counts of words to represent texts, and kernel-CCA (Lai et al. 2000) to align both modalities.

The following approaches rely on unsupervised strategies to obtain pretrained representations of each modality, and align both modalities using supervised strategies on parallel corpora. For instance, Socher et al. 2013 use the unsupervised method of Coates et al. 2011 to extract image vectors based on sparse coding and SIFT descriptors (Lowe 2004), the unsupervised method of E. H. Huang et al. 2012 to extract 50-dimensional word vectors, and the pairwise loss to learn a linear projection between modalities.

**Deep approaches** The approach proposed by Frome et al. 2013 is among the first to perform text-image alignment with deep neural network representations. It uses a pretrained AlexNet (Krizhevsky et al. 2012) to extract 4096 dimensional vectors from images, a pretrained skip-gram model (Tomas Mikolov et al. 2013a) to extract 1000 dimensional vectors from words, and a pairwise loss with a linear projection between modalities.

Later on, approaches that directly align images with sentences instead of words were proposed. Socher et al. 2014 align the image representations extracted from AlexNet and the sentence representations from a Dependecy-Tree Recursive Neural Network (DT-RNN) using two crossmodal triplet losses. Alternatively, (Kiros et al. 2014) use a Long-Short Term Memory (LSTM) and a different regularization scheme. (Eisenschtat et al. 2017) use the richer VGG16 features with a pairwise loss. (Faghri et al. 2018) use the once again richer ResNet152 features and a Gated Recurrent Unit (GRU) with a triplet loss.

**Object-based approaches** Other approaches on real scene datasets rely on object-based features descriptors. They more precisely model interactions between parts of the image and parts of their associated sentence. Karpathy et al. 2014 use a pretrained Region-based Convolutional Neural Network (R-CNN) (Girshick et al. 2014) to extract a bag of object-based of visual features from the image, and use sentence dependency tree relations (De Marneffe et al. 2006) to extract a bag of pairwise relationships between words of the sentence. The parameters of a two layer fully connected on top of the sentence representations and a linear projection are optimized with a part-based triplet loss and two crossmodal triplet losses. Karpathy et al. 2015 improve the textual representation using a pretrained word embedding matrix from word2rec (Tomas Mikolov et al. 2013b) followed by a Bidirectional Recurrent Neural Network (Bi-RNN), and improve the alignment strategy by forcing each part of the sentence to be associated with only one part of the image.

**Fusion approaches** Instead of aligning the modalities in a late fusion, early fusion approaches can be used to learn the distance function (Clinchant et al. 2011). L. Wang et al. 2018 project vector representations of both modalities in a same dimensional space and train a multi-layer network on top. Y. Huang et al. 2017 use a LSTM to iteratively fuse both modalities until the final similarity processing. Instead of a LSTM, Nam et al. 2017 use several dual attention network to fuse the modalities. However, a downside of these approaches is their higher computational cost during inference which makes them unsuited for real applications.

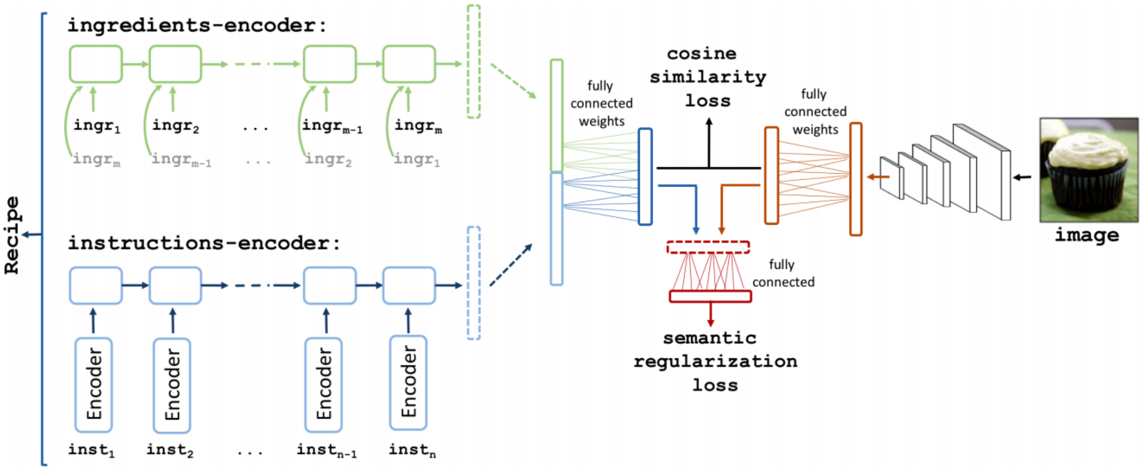


Figure 2.2 – Data specific approach for image-recipe alignment (Salvador et al. 2017). On the left, vector representations are extracted from a set of ingredients and a list of cooking instructions. On the left, vector representations are extracted from the associated image. In the middle, both representations are linearly projected and aligned in the same retrieval space. An auxiliary classification task is used to regularize the training by taking advantage of the additional semantic information. Figure from (Salvador et al. 2017).

**Recipe1M approach** Some approaches introduce architectural priors regarding a certain kind of data. Salvador et al. 2017 introduce a neural architecture specifically tailored for cooking recipes. It is trained on the Recipe1M dataset which is defined in the preceding Section 2.2.1. This dataset is composed of around 1 million image-recipe pairs and comes with additional semantic information of the form of cooking categories such as *pizza* or *hamburger*. As illustrated in Figure 2.2, they extract vector representations from a set of ingredients using a Bidirectional Long Short-Term Memory (Bi-LSTM) on top of a word embedding matrix. The latter is initialized with the word2vec representations (Tomas Mikolov et al. 2013b). They extract vector representations from a list of cooking instructions using a hierarchical LSTM. The first LSTM takes the word representations from word2vec and extract a representation of each instruction separately. It is pretrained using the skip-thought method (Kiros et al. 2015) on this dataset which is detailed in Section 1.2.5. In other words, it was trained to predict the next and previous instructions. A second LSTM sequentially processes the extracted representations of each instruction to output a contextualized vector representations. They extract a global vector representation of the recipe by concatenating the two vector representations of its ingredients and cooking instructions. They extract vector representations from an image of a dish using a pretrained ResNet152. The latter is fine-tuned after a certain amount of epochs. They linearly project both

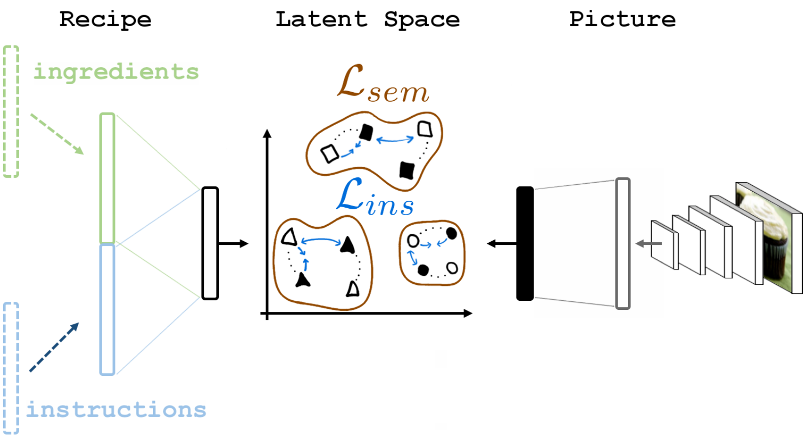


Figure 2.3 – AdaMine approach to learn a multimodal space for cross-modal retrieval. Recipes, made of ingredients and instructions, and pictures are embedded by two different neural networks on a shared retrieval space. Both are aligned by optimizing a combination of different triplet losses. The instance-based loss $\mathcal{L}_{ins}$ aligns recipes with their corresponding pictures. The semantic-based loss $\mathcal{L}_{sem}$ adds structure by aligning recipes and pictures of the same class.

modalities in the same retrieval space and use a pairwise loss to align them. They also use a classification head to take advantage of the additional information. This head is trained to output the cooking category for each vector representation from the retrieval space. This auxiliary task acts as a regularization which helps structure the retrieval space.

In the following section, we propose a crossmodal retrieval approach for a real application. The latter consists in a large-scale search engine for cooking recipes based on the Recipe1M dataset defined in Section 2.2.1. Our approach efficiently takes the additional semantic information into account in the modeling. Contrarily to the approach proposed by Salvador et al. 2017, we use multiple triplet losses instead of pairwise loss to align the two spaces. Instead of relying on a classification head on top of the retrieval space, we propose to integrate the semantic information using crossmodal triplet losses.

## 2.3 AdaMine approach

### 2.3.1 Model Overview

The objective of our model **AdaMine** (ADAptive MINing Embeding) is to learn the representations of recipe items (texts and images) through a joint retrieval and classification learning framework based on a double-triplet learning scheme. More particularly, our model relies on the following hypotheses:

- H1: Aligning items according to a retrieval task allows capturing the fine-grained semantics of items, since the obtained embeddings must rank individual items with respect to each other.
- H2: Aligning items according to class meta-data allows capturing the high-level semantic information underlying items since it ensures the identification of item clusters that correspond to class-based meta-data.
- H3: Learning simultaneously retrieval and class-based features allows enforcing a multi-scale structure within the latent space, which covers all aspects of item semantics. In addition, we conjecture that adding a classification layer sequentially to manifold-alignment as in Salvador et al. 2017 might be under-effective.

Based on these hypotheses, we propose to learn the latent space structure (and item embeddings) by integrating both retrieval objective and semantic information in a single cross-modal metric learning problem (see the Latent Space in Figure 2.3). We take inspiration from the learning-to-rank retrieval framework by building a learning schema based on query/relevant item/irrelevant item triplets noted $(x_q, x_p, x_n)$. Following hypothesis H3, we propose a double-triplet learning scheme that relies on both instance-based and semantic-based triplets, noted respectively $(x_q, x_p, x_n)$ and $(x'_q, x'_p, x'_n)$, in order to satisfy the multi-level structure (fine-grained and high-level) underlying semantics. More particularly, we learn item embeddings by minimizing the following objective function:

$$\mathcal{L}_{total}(\theta) = \mathcal{L}_{ins}(\theta) + \lambda \mathcal{L}_{sem}(\theta) \tag{2.5}$$

where $\theta$ is the network parameter set. $\mathcal{L}_{ins}$ is the loss associated with the retrieval task over instance-based triplets $(x_q, x_p, x_n)$, and $\mathcal{L}_{sem}$ is the loss coming with the semantic information over semantic-based triplets $(x'_q, x'_p, x'_n)$. Unlike Salvador et al. 2017 that expresses this second term $\mathcal{L}_{sem}$ acting as a regularization over the $\mathcal{L}_{ins}$ optimization, in our framework, it is expressed as a joint classification task.

This double-triplet learning framework is a difficult learning problem since the trade-off between $\mathcal{L}_{ins}$ and $\mathcal{L}_{sem}$ is not only influenced by $\lambda$ but also by the sampling of instance-based and semantic-based triplets and depends on their natural distribution. Furthermore, the sampling of violating triplets can be difficult as the training progresses which usually leads to vanishing gradient problems that are common in triplet-based losses, and are amplified by our double-triplet framework. To alleviate these problems, we propose an adaptive sampling strategy that normalizes each loss allowing to fully control the trade-off with $\lambda$ alone while also ensuring non-vanishing gradients throughout the learning process.

In the following, we present the network architecture, as well as each component of our learning framework, and then discuss the learning scheme of our model.

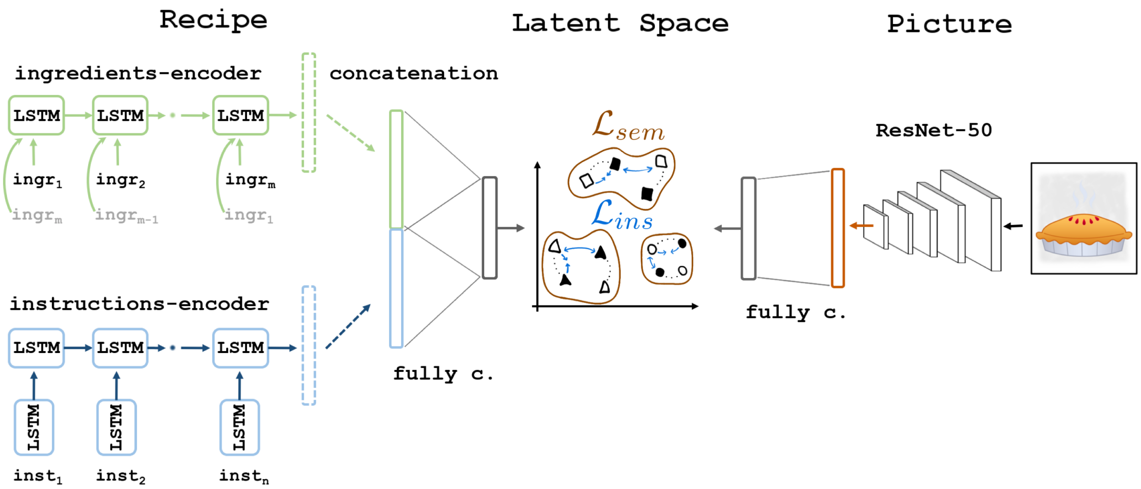


Figure 2.4 – AdaMine architecture to learn a multimodal space for cross-modal retrieval. On the left, the ingredients are embedded using a pretrained bidirectionnal LSTM. The instructions are embedded using a hierarchical LSTM. The first level is pretrained and produces a vector representation for each instruction. The concatenation between the vector output of the second level LSTM and the ingredients embedding vector is then linearly projected on the shared space by a fully connected layer. On the right, the vector representation of the image is extracted with a pretrained ResNet-50 and projected on the same latent space using another fully connected layer.

## 2.3.2 Multimodal learning framework

### 2.3.2.1 Multimodal architecture

Our architecture is based on the proposal of Salvador et al. 2017, which consists of two branches based on deep neural networks that map each modality (image or text recipe) into a common representation space, where they can be compared. Our global architecture is depicted in Figure 2.4.

The image branch is composed of a ResNet-50 model (He et al. 2015a). It contains 50 convolutional layers, totaling more than 25 million parameters. This architecture is further detailed in He et al. 2015a, and was chosen in order to obtain comparable results to Salvador et al. 2017 by sharing a similar setup. The ResNet-50 is pretrained on the large-scale dataset of the ImageNet Large Scale Visual Recognition Challenge (Russakovsky et al. 2015b), containing 1.2 million images, and is fine-tuned with the whole architecture. This neural network is followed by a fully connected layer, which maps the outputs of the ResNet-50 into the latent space, and is trained from scratch.

In the recipe branch, ingredients and instructions are first embedded separately, and their obtained representations are then concatenated as input of a fully connected layer that maps the recipe features into the latent space. For ingredients, we use a bidirectional LSTM (Hochreiter et al. 1997) on pretrained embeddings obtained with the word2vec algorithm (Tomas Mikolov et al. 2013b). With the objective to consider the different granularity levels of the instruction text, we use a hierarchical LSTM in which the word-level is pretrained using the skip-thought method (Kiros et al. 2015) and is not fine-tuned while the sentence-level is learned from scratch.

#### 2.3.2.2 Retrieval loss

The objective of the retrieval loss $\mathcal{L}_{ins}$ is to learn item embeddings by constraining the latent space according to the following assumptions (Hypothesis H1): 1) ranking items according to a similarity metric in order to gather matching items together and 2) discriminating irrelevant ones. We propose to use a loss function $\mathcal{L}_{ins}$ based on a particular triplet $(x_q, x_p, x_n)$ consisting of a query $x_q$, its matching counterpart in the other modality $x_p$ and a dissimilar item $x_n$. The retrieval loss function $\mathcal{L}_{ins}$ is the aggregation of the individual loss $\mathcal{L}_{ins}$ over all triplets. The aim of $\mathcal{L}_{ins}$ is to provide a fine-grained structure to the latent space where the nearest item from the other modality with respect to the query is optimized to be its matching pair. More formally, the individual retrieval loss $\mathcal{L}_{ins}(\theta, x_q, x_p, x_n)$ is formalized as follows:

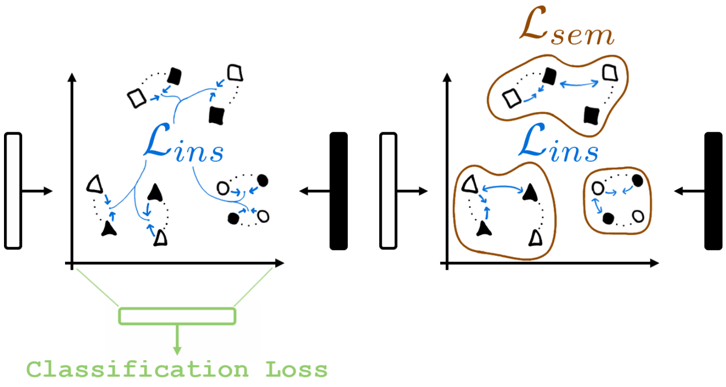
$$\mathcal{L}_{ins}(\theta, x_q, x_p, x_n) = \left[ d(x_q, x_p) + \alpha - d(x_q, x_n) \right]_+ \tag{2.6}$$

where $d(x, y)$ expresses the cosine distance between vectors $x$ and $y$ in the latent space $\mathcal{F}$.

#### 2.3.2.3 Semantic loss

$\mathcal{L}_{sem}$ is acting as a regularization term capable of taking advantage of semantic information in the multi-modal alignment, without adding extra parameters to the architecture nor graph dependencies. To leverage class information (Hypotheses H2), we propose to construct triplets that optimize a surrogate of the k-nearest neighbor classification task. Ideally, for a given query $x_q$, and its corresponding class $c(x_q)$, we want its associated the closest sample $x_{\star,q}$ in the feature space to respect $c(x_q) = c(x_{\star,q})$. This enforces a semantic structure on the latent space by making sure that related dishes are closer to each other than to non-related ones. To achieve this, we propose the individual triplet loss $\mathcal{L}_{sem}$:

$$\mathcal{L}_{sem}(\theta, x'_q, x'_p, x'_n) = \left[ d(x'_q, x'_p) + \alpha - d(x'_q, x'_n) \right]_+ \tag{2.7}$$



(a) (Salvador et al. 2017)    (b) AdaMine (ours)

Figure 2.5 – Comparison between (a) the classification augmented latent space learning of Salvador et al. 2017 and (b) our joint retrieval and semantic latent space learning, which combines instance-based ($\mathcal{L}_{ins}$) and semantic-based ($\mathcal{L}_{sem}$) losses.

where $x'_p$ belongs to the set of items with the same semantic class $c(x'_q)$ as the query, and $x'_n$ belongs to the set of items with different semantic classes than the one of the query.

Contrary to the classification machinery adopted by Salvador et al. 2017, $\mathcal{L}_{sem}$ optimizes semantic relations directly in the latent space without changing the architecture of the neural network, as shown in Figure 2.5. This promotes a smoothing effect on the space by encouraging instances of the same class to stay closer to each other.

### 2.3.3 Adaptive learning schema

As commonly used in Deep Learning (DL), we use the Stochastic Gradient Descent (SGD) algorithm which approximates the true gradient over mini-batches. The updated term is generally computed by aggregation of the gradient using the *average* over all triplets in the mini-batch. However, this *average* strategy tends to produce a vanishing update with triplet losses. This is especially true towards the end of the learning phase, as the few active constraints are averaged with many zeros coming from the many inactive constraints. We believe this problem is amplified as the size of the training set grows. To tackle this issue, our proposed *adaptive* strategy considers an update term $\delta_{adm}$ that takes into account informative triplets only (*i.e.*, non-zero loss). More formally, given a mini-batch $\mathbb{B}$, $\mathbb{P}_q^r$ the set of matching items with respect to a query $x_q$ and $\mathbb{P}_q^s$ the set of items with the same class as $x_q$, the update term $\delta_{adm}$ is defined by:

$$\delta_{adm} = \sum_{x_q \in \mathbb{B}} \left( \sum_{x_p \in \mathbb{B} \cap \mathbb{P}_q^r} \sum_{x_n \in \mathbb{B} \setminus \mathbb{P}_q^r} \frac{\nabla \mathcal{L}_{ins}(\theta, x_q, x_p, x_n)}{\beta_r'} \right. \\ \left. + \sum_{x_p \in \mathbb{B} \cap \mathbb{P}_q^s} \sum_{x_n \in \mathbb{B} \setminus \mathbb{P}_q^s} \lambda \frac{\nabla \mathcal{L}_{sem}(\theta, x_q, x_p, x_n)}{\beta_s'} \right) \tag{2.8}$$

with $\beta_r'$ and $\beta_s'$ being the number of triplets contributing to the cost:

$$\begin{aligned} \beta_r' &= \sum_{x_q \in \mathbb{B}} \sum_{x_p \in \mathbb{B} \cap \mathbb{P}_q^r} \sum_{x_n \in \mathbb{B} \setminus \mathbb{P}_q^r} \mathbb{1}_{\mathcal{L}_{ins} \neq 0} \\ \beta_s' &= \sum_{x_q \in \mathbb{B}} \sum_{x_p \in \mathbb{B} \cap \mathbb{P}_q^s} \sum_{x_n \in \mathbb{B} \setminus \mathbb{P}_q^s} \mathbb{1}_{\mathcal{L}_{sem} \neq 0} \end{aligned} \tag{2.9}$$

At the very beginning of the optimization, all triplets contribute to the cost and, as constraints stop being violated, they are dropped. At the end of the training phase, most of the triplets will have no contribution, leaving the hardest negatives to be optimized without vanishing gradient issues. Remark that this corresponds to a *curriculum learning* starting with the average strategy and ending with the hard negative strategy like in Schroff et al. 2015, but without the burden of finding the time-step at which to switch between strategies as this is automatically controlled by the weights $\beta_r$ and $\beta_s$.

Remark also that an added benefit of $\delta_{adm}$ is due to the independent normalization of each loss by its number of active triplets. Thus, $\delta_{adm}$ keeps the trade-off between $\mathcal{L}_{ins}$ and $\mathcal{L}_{sem}$ unaffected by differences between the number of active triplets in each loss. Thus, it reduces the number of hyperparameter to optimize in the loss function. $\lambda$ becomes the only effective hyperparameter.

## 2.4 Experiments

### 2.4.1 Experimental setup

**Dataset** We use the Recipe1M dataset (Salvador et al. 2017). As introduced in Section 2.2.1, it is the only large-scale dataset including both English cooking recipes (ingredients and instructions), images, and categories. The raw Recipe1M dataset consists of about 1 million image and recipe pairs. It is currently the largest one in English, including twice as many recipes as Kusmierczyk et al. 2016 and eight times as many images as J. Chen et al. 2016. Furthermore, the availability of semantic information makes it particularly suited to validate our model: around half of the pairs are associated with a class, among 1048 classes parsed from the recipe titles. Using the same preprocessed pairs of recipe-image provided by Salvador et al. 2017, we end up with 238,399 matching pairs of images and recipes for the training set, while the validation and test sets have 51,119 and 51,303 matching pairs, respectively.

**Evaluation methodology** We carry out a cross-modal retrieval task following the process described in Salvador et al. 2017. Specifically, we first sample 10 unique subsets of 1,000 (1k setup) or 5 unique subsets of 10,000 (10k setup) matching text recipe-image pairs in the test set. Then, we consider each item in a modality as a query (for instance, an image), and we rank items in the other modality (resp. text recipes) according to the cosine distance between the query embedding and the candidate embeddings. The objective is to retrieve the associated item in the other modality at the first rank. The retrieved lists are evaluated using standard metrics in cross-modal retrieval tasks. For each subset (1k and 10k), we estimate the median retrieval rank (MedR), as well as the recall percentage at top K (R@K), over all queries in a modality. The R@K corresponds to the percentage of queries for which the matching item is ranked among the top K closest results.

**Baselines** To test the effectiveness of our model **AdaMine**, we evaluate our multi-modal embeddings with respect to those obtained by state-of-the-art (SOTA) baselines:

- **CCA**, which denotes the Canonical Correlation Analysis method (Hotelling 1936). This baseline allows testing the effectiveness of global alignment methods.
- **PWC**, the pairwise loss with the classification layer from Salvador et al. 2017. We report their state-of-the-art results for the 1k and 10k setups when available. This baseline exploits the classification task as a regularization of embedding learning.
- **PWC***, our implementation of the architecture and loss described by Salvador et al. 2017. The goal of this baseline is to assess the results of its improved version **PWC++**, described below.
- **PWC++**, the improved version of our implementation **PWC***. More particularly, we add a positive margin to the pairwise loss adopted in Salvador et al. 2017, as proposed by J. Hu et al. 2014. This additional positive margin allows matching pairs to have different representations, thus reducing the risk of overfitting. In practice, the positive margin is set to 0.3 and the negative margin to 0.9.

We evaluate the effectiveness of our model **AdaMine**, which includes both the triplet loss and the adaptive learning, in different setups, and having the following objectives:

- Evaluating the impact of the retrieval loss: we run the **AdaMine_ins** scenario which refers to our model with the instance loss $\mathcal{L}_{ins}$ only and the adaptive learning strategy (the semantic loss $\mathcal{L}_{sem}$ is discarded);

• Evaluating the impact of the semantic loss: we run the **AdaMine_sem** scenario which refers to our model with the semantic loss $\mathcal{L}_{sem}$ only and the adaptive learning strategy (the instance loss $\mathcal{L}_{ins}$ is discarded);

• Evaluating the impact of the strategy used to tackle semantic information: we run the **AdaMine_ins+cls** scenario which refers to our **AdaMine** model by replacing the semantic loss by the classification head proposed by Salvador et al. 2017;

• Measuring the impact of our adaptive learning strategy: we run the **AdaMine_avg**. The architecture and the losses are identical to our proposal, but instead of using the adaptive learning strategy, this one performs the stochastic gradient descent averaging the gradient over all triplets, as is common practice in the literature;

• Evaluating the impact of the text structure: we run our whole model (retrieval and semantic losses + adaptive SGD) by considering either ingredients only (noted **AdaMine_ingr**) or instructions only (noted **AdaMine_instr**).

## 2.4.2 Approach validation

### 2.4.2.1 Validation of the semantic triplet loss

We analyze our main hypotheses related to the importance of semantic information for learning multi-modal embeddings (see Hypothesis H2 in 2.3.1). Specifically, in this part we test whether semantic information can help to better structure the latent space, taking into account class information and imposing structural coherence. Compared with Salvador et al. 2017 which adds a classification layer, we believe that directly injecting this semantic information with a global loss $\mathcal{L}(\theta)$ (Equation 2.5) comes as a more natural approach to integrating class-based meta-data (see Hypothesis H3 in 2.3.1).

To test this intuition, we start by quantifying, in Table 2.1, the impacts of the semantic information in the learning process. To do so, we evaluate the effectiveness of different scenarios of our model **AdaMine** with respect to the multi-modal retrieval task (image-to-text and text-to-image) in terms of *MedR* and Recall at ranks 1, 5, and 10. Compared with a retrieval loss alone (**AdaMine_ins**), we point out that adding semantic information with a classification cost **AdaMine_ins+cls** or a semantic loss **AdaMine** improves the results. When evaluating with 10,000 pairs (10k setting), while **AdaMine_ins** obtains MedRs 15.4 and 15.8, the semantic models (**AdaMine_ins+cls** and **AdaMine**) lower these values to 14.8 and 15.2, and 13.2 and 12.2, respectively (lower is better) for both retrieval tasks (image-to-text and text-to-image).

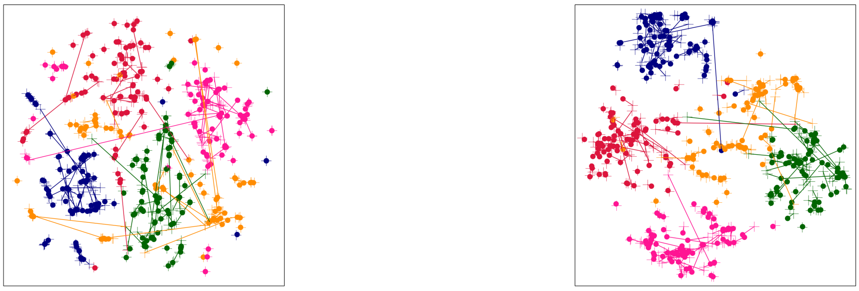
The importance of semantic information becomes clearer when we directly compare the impact of adding the semantic loss to the base model (**AdaMine** vs **AdaMine_ins**), since the former obtains the best results for every metric. To better understand this phenomenon, we depict in Figure 2.6 item embeddings obtained

Table 2.1 – Impact of the semantic information. MedR means Median Rank (lower is better). R@K means Recall at K (between 0% and 100%, higher is better). The average value over 5 bags of 10,000 pairs each is reported.

| Scenarios | Strategies | Image to Textual recipe | | | |
|---|---|---|---|---|---|
| | | MedR | R@1 | R@5 | R@10 |
| AdaMine_ins | Retrieval loss | 15.4 | 13.3 | 32.1 | 42.6 |
| AdaMine_ins+cls | Retrieval loss + Classification loss | 14.8 | 13.6 | 32.7 | 43.2 |
| AdaMine | Retrieval loss + Semantic loss | **13.2** | **14.9** | **35.3** | **45.2** |
| Scenarios | Strategies | Textual recipe to Image | | | |
| | | MedR | R@1 | R@5 | R@10 |
| AdaMine_ins | Retrieval loss | 15.8 | 12.3 | 31.1 | 41.7 |
| AdaMine_ins+cls | Retrieval loss + Classification loss | 15.2 | 12.9 | 31.8 | 42.5 |
| AdaMine | Retrieval loss + Semantic loss | **12.2** | **14.8** | **34.6** | **46.1** |

by the **AdaMine_ins** and **AdaMine** models using a t-SNE visualization. This figure is generated by selecting 400 matching recipe-image pairs (800 data points), which are randomly selected from, and equally distributed among 5 of the most occurring classes of the Recipe1M dataset. Each item is colored according to its category (e.g., blue points for the cupcake class), and items of the same instance are connected with a trace. Therefore, Figure 2.6 allows drawing two conclusions: 1) our model—on the right side of the figure—is able to structure the latent space while keeping items of the same class close to each other (see color clusters); 2) our model reduces the sum of distances between pairs of instances (in the figure, connected with traces), thus reducing the MedR and increasing the recall.

We also illustrate this comparison through qualitative examples. In Figure 2.7, **AdaMine** (top row) and **AdaMine_ins** (bottom row) are compared on four queries, for which both models are able to rank the correct match in the top-5 among 10,000 candidates. For the first and second queries (cucumber salad and roasted chicken, respectively), both models are able to retrieve the matching image in the first position. However, the rest of the top images retrieved by our model are semantically related to the query, by sharing critical ingredients (cucumber, chicken) of the recipe. In the third and fourth queries (pizza and chocolate chip, respectively), our model is able to rank both the matching image and semantically connected samples in a more coherent way, due to a better alignment of the retrieval space produced by the semantic modeling. These results reinforce our intuition that it is necessary to integrate semantic information in addition to item pairwise anchors while learning multi-modal embeddings.



(a) Without AdaMine

(b) With AdaMine

Figure 2.6 – t-SNE visualization. Image (resp. Recipe) points are denoted with the + (resp. •) symbol. Matching pairs are connected with a trace. Blue points are associated with the cupcake class, orange to hamburger, pink to green beans, green to pork chops, and red to pizza.

Second, we evaluate our intuition that classification is under-effective for integrating the semantics within the latent space (see Hypothesis H3 in 2.3.1). Table 2.1 shows that our semantic loss **AdaMine**, proposed in Section 2.3.2.3, outperforms our model scenario **AdaMine_ins+cls** which relies on a classification head as proposed in Salvador et al. 2017. For instance, we obtain an improvement of $+9.57\%$ in terms of $R@1$ with respect to the classification loss setting **AdaMine_ins+cls**. This result suggests that our semantic loss is more appropriate to organize the latent space in order to retrieve text-image matching pairs. It becomes important, then, to understand the impacts of the weighting factor $\lambda$ between the two losses $\mathcal{L}_{ins}$ and $\mathcal{L}_{sem}$ (Equation 2.5). In Figure 2.8, we observe a fair level of robustness for lower values of $\lambda$, but any value over 0.5 has a hindering effect on the retrieval task, since the semantic grouping starts to be of considerable importance. These experiments confirm the importance of additional semantic clues: despite having one million fewer parameters than Salvador et al. 2017's proposal, our approach still achieves better scores, when compared to the addition of the classification head.

**State-of-the-art comparison** In the following, we evaluate the effectiveness of our model, compared to different baseline models. Results are presented in Table 2.2 for the image-to-recipe and in Table 2.3 recipe-to-image retrieval tasks. We report results on the 1K setup and test the robustness of our model on the 10k setup by reporting only the best state-of-the-art (SOTA) [1] baseline for comparison. From a general point of view, we observe that our model **AdaMine** overpasses the different baselines and model scenarios. Small values of standard deviation outlines the low variability of experimented models, and accordingly the robustness of obtained results. For instance, our model reaches a value equal to 1 for the Median Rank metric (MedR) for the 1k setting and both retrieval tasks

[1] at the time of submission

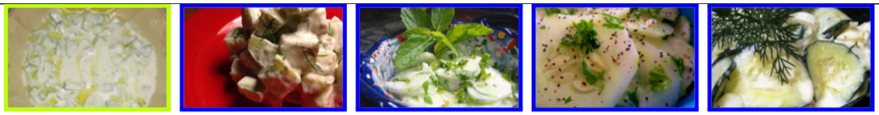
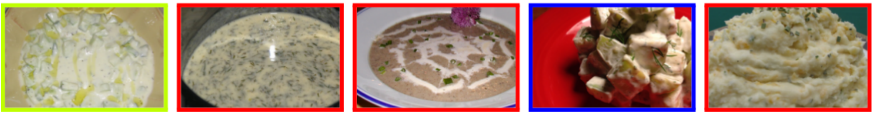
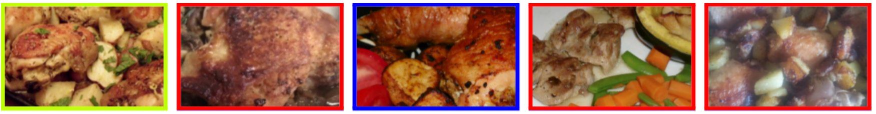
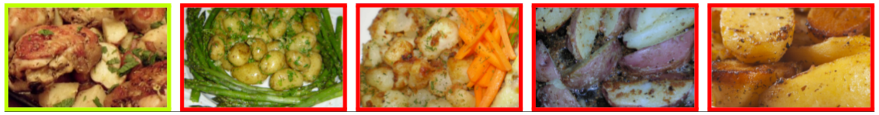
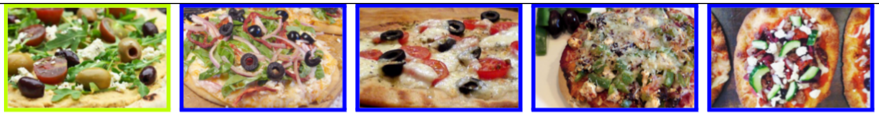
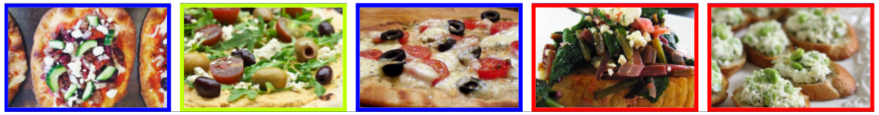
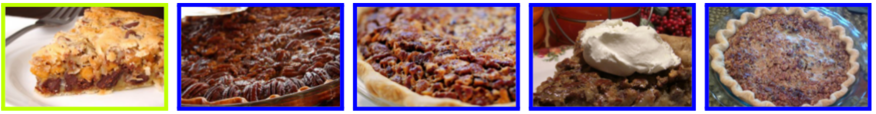
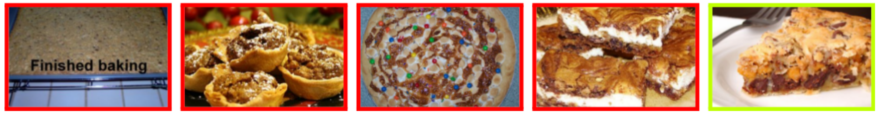
| Ingredient query | Cooking instruction query | Top 5 retrieved images | |
|---|---|---|---|
| *Yogurt, cucumber, salt, garlic clove, fresh mint.* | *Stir yogurt until smooth. Add cucumber, salt, and garlic. Garnish with mint. Normally eaten with pita bread. Enjoy!* |  | AM |
| | |  | AM_ins |
| *Olive oil, balsamic vinegar, thyme, lemons, chicken drumsticks with bones and skin, garlic, potatoes, parsley.* | *Whisk together oil, mustard, vinegar, and herbs. Season to taste with a bit of salt and pepper and a large pinch or two of brown sugar. Place chicken in a non-metal dish and pour marinade on top to coat. [...]* |  | AM |
| | |  | AM_ins |
| *Pizza dough, hummus, arugula, cherry or grape tomatoes, pitted greek olives, feta cheese.* | *Cut the dough into two 8-ounce sized pieces. Roll the ends under to create round balls. Then using a well-floured rolling pin, roll the dough out into 12-inch circles. [...]* |  | AM |
| | |  | AM_ins |
| *Unsalted butter, eggs, condensed milk, sugar, vanilla extract, chopped pecans, chocolate chips, butterscotch chips, [...]* | *Preheat the oven to 375 degrees F. In a large bowl, whisk together the melted butter and eggs until combined. Whisk in the sweetened condensed milk, sugar, vanilla, pecans, chocolate chips, butterscotch chips, [...]* |  | AM |
| | |  | AM_ins |

Figure 2.7 – Visualization of a recipe-to-image search. For each recipe, we have the top row, indicating the top 5 images retrieved by our AdaMine model for a given recipe query, and the bottom row, indicating the top 5 images by the triplet loss for the same recipe. In green, the matching image. In blue, images belonging to the same class than the recipe. In red, images belonging to a different class. *AM* indicates AdaMine, and *AM_ins* AdaMine_ins.

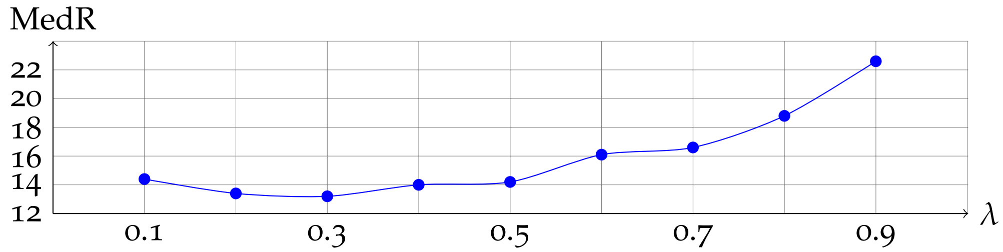


Figure 2.8 – MedR scores for different values of $\lambda$, responsible for weighting the semantic regularization cost $\mathcal{L}_{sem}$ of **AdaMine**, calculated over 5 bags of 10.000 validation samples.

while the well-known SOTA models CCA and PWC++ obtain respectively 15.7 and 3.3. Contrary to PWC, all of our model scenarios, denoted **AdaMine_**$_*$, adopt the triplet loss. Ablation tests on our proposals show their effectiveness. This trend is noticed over all retrieval tasks and all metrics. The comparison of the results obtained over 1k and 10k settings outlines the same statement with larger improvements (with similar standard deviation) for our model **AdaMine** with respect to SOTA models and **AdaMine**-based scenarios. More particularly, we first begin our discussion with the comparison with respect to SOTA models and outline the following statements:

• Global alignment models (baseline **CCA**) are less effective than advanced models (**PWC**, **PWC++**, and **AdaMine**). Indeed, the **CCA** model obtains a MedR value of 15.7 for the image-to-text retrieval task (1k setting) while the metric range of advanced models is between 1 and 5.2. This suggests the effectiveness of taking into account dissimilar pairs during the learning process.

• We observe that our triplet based model **AdaMine** consistently outperforms pairwise methods (**PWC** and **PWC++**). For instance, our model obtains a significant decrease of $-61.84\%$ in terms of MedR with respect to **PWC++** for the 10k setting and the image-to-text retrieval task. This suggests that relative cosine distances are better at structuring the latent space than absolute cosine distances.

• Our model **AdaMine** surpasses the current state-of-the-art results by a large margin. For the 1k setup, it reduces the medR score by a factor of 5—from 5.2 and 5.1 to 1.0 and 1.0—, and by a factor bigger than 3 for the 10k setup. One strength of our model is that it has fewer parameters than **PWC++** and **PWC**, since the feature space is directly optimized with a semantic loss, without the addition a parameter-heavy head to the model.

Second, the comparison according to different versions of our model outlines three main statements:

• The analysis of **AdaMine_ins**, **AdaMine_ins+cls**, and **AdaMine** corroborates the results observed in Section 5.1 dealing with the impact of the semantic loss on the performance of the model. In the 1k setting, the instance-based approach (**AdaMine_ins**) achieves a MedRs value equal of 1.5 and 1.6 for both tasks (lower is

Table 2.2 – State-of-the-art comparison for the image-to-recipe retrieval setting. MedR means Median Rank (lower is better). R@K means Recall at K (between 0% and 100%, higher is better). The mean and std values over 10 (resp. 5) bags of 1k (resp. 10k) pairs each are reported for the top (resp. bottom) table. Items marked with a star (*) are our reimplementation of the cited methods.

| | | | Image to Textual recipe | | | |
|---|---|---|---|---|---|---|
| | | | MedR | R@1 | R@5 | R@10 |
| 1k items | SOTA | Random | 499 | 0.0 | 0.0 | 0.0 |
| | | CCA (Salvador et al. 2017) | 15.7 | 14.0 | 32.0 | 43.0 |
| | | PWC (Salvador et al. 2017) | 5.2 | 24.0 | 51.0 | 65.0 |
| | | PWC* (Salvador et al. 2017) | 5.0 ± 0.4 | 22.8 ± 1.4 | 47.7 ± 1.4 | 60.1 ± 1.4 |
| | | PWC++ | 3.3 ± 0.4 | 25.8 ± 1.6 | 54.5 ± 1.3 | 67.1 ± 1.4 |
| | Model scenarios | AdaMine_sem | 21.1 ± 2.0 | 8.7 ± 0.7 | 25.5 ± 0.9 | 36.5 ± 0.9 |
| | | AdaMine_ins | 1.5 ± 0.5 | 37.5 ± 1.1 | 67.0 ± 1.3 | 76.8 ± 1.5 |
| | | AdaMine_ins+cls | 1.1 ± 0.3 | 38.3 ± 1.6 | 67.5 ± 1.2 | 78.0 ± 0.9 |
| | | AdaMine_avg | 2.3 ± 0.5 | 30.6 ± 1.1 | 60.3 ± 1.2 | 71.4 ± 1.3 |
| | | AdaMine_ingr | 4.9 ± 0.5 | 22.6 ± 1.4 | 48.5 ± 1.6 | 59.8 ± 1.3 |
| | | AdaMine_instr | 3.9 ± 0.5 | 24.4 ± 1.6 | 52.6 ± 2.0 | 65.4 ± 1.6 |
| | | AdaMine | **1.0** ± **0.1** | **39.8** ± **1.8** | **69.0** ± **1.8** | **77.4** ± **1.1** |
| 10k items | | PWC++ (best SOTA) | 34.6 ± 1.0 | 7.6 ± 0.2 | 19.8 ± 0.1 | 30.3 ± 0.4 |
| | Model scenarios | AdaMine_sem | 207.3 ± 3.9 | 1.4 ± 0.3 | 5.7 ± 0.3 | 9.6 ± 0.3 |
| | | AdaMine_ins | 15.4 ± 0.5 | 13.3 ± 0.2 | 32.1 ± 0.7 | 42.6 ± 0.8 |
| | | AdaMine_ins+cls | 14.8 ± 0.4 | 13.6 ± 0.2 | 32.7 ± 0.4 | 43.2 ± 0.3 |
| | | AdaMine_avg | 24.6 ± 0.8 | 10.0 ± 0.2 | 25.9 ± 0.4 | 35.7 ± 0.5 |
| | | AdaMine_ingr | 52.8 ± 1.2 | 6.5 ± 0.2 | 17.9 ± 0.2 | 25.8 ± 0.3 |
| | | AdaMine_instr | 39.0 ± 0.9 | 6.4 ± 0.1 | 18.9 ± 0.4 | 27.6 ± 0.5 |
| | | AdaMine | **13.2** ± **0.4** | **14.9** ± **0.3** | **35.3** ± **0.2** | **45.2** ± **0.2** |

better), while the addition of a classification head (**AdaMine_ins+cls**), proposed by Salvador et al. 2017, improves these results to 1.1 and 1.2. Removing the classification head and adding a semantic loss (**AdaMine**) further improves the results to 1 for both retrieval tasks which further validates Hypothesis H3 in 2.3.1.

• The adaptive sampling strategy described in Section 2.3.3 strongly contributes to the good results of **AdaMine**. With **AdaMine_avg**, we test the same setup of **AdaMine**, replacing the adaptive strategy with the average one. The importance of removing triplets that are not contributing to the loss becomes evident when the scores for both strategies are compared: 24.6 and 24.0 of MedR (lower is better)

Table 2.3 – State-of-the-art comparison for the recipe-to-image retrieval setting. MedR means Median Rank (lower is better). R@K means Recall at K (between 0% and 100%, higher is better). The mean and std values over 10 (resp. 5) bags of 1k (resp. 10k) pairs each are reported for the top (resp. bottom) table. Items marked with a star (*) are our reimplementation of the cited methods.

| | | | Textual recipe to Image | | | |
|---|---|---|---|---|---|---|
| | | | MedR | R@1 | R@5 | R@10 |
| 1k items | SOTA | Random | 499 | 0.0 | 0.0 | 0.0 |
| | | CCA (Salvador et al. 2017) | 24.8 | 9.0 | 24.0 | 35.0 |
| | | PWC (Salvador et al. 2017) | 5.1 | 25.0 | 52.0 | 65.0 |
| | | PWC* (Salvador et al. 2017) | 5.3 ± 0.4 | 21.2 ± 1.2 | 48.0 ± 1.1 | 60.4 ± 1.4 |
| | | PWC++ | 3.5 ± 0.5 | 24.8 ± 1.1 | 55.0 ± 1.8 | 67.1 ± 1.2 |
| | Model scenarios | AdaMine_sem | 21.1 ± 1.9 | 8.2 ± 0.9 | 25.5 ± 1.0 | 36.2 ± 0.9 |
| | | AdaMine_ins | 1.6 ± 0.5 | 36.1 ± 1.6 | 66.6 ± 1.3 | 76.8 ± 1.5 |
| | | AdaMine_ins+cls | 1.2 ± 0.4 | 37.5 ± 1.4 | 67.7 ± 1.2 | 77.3 ± 1.0 |
| | | AdaMine_avg | 2.2 ± 0.3 | 30.6 ± 1.8 | 60.6 ± 1.1 | 71.9 ± 1.1 |
| | | AdaMine_ingr | 5.0 ± 0.6 | 21.5 ± 1.4 | 47.7 ± 2.1 | 59.8 ± 1.8 |
| | | AdaMine_instr | 3.7 ± 0.5 | 23.6 ± 1.7 | 52.7 ± 1.6 | 65.5 ± 1.5 |
| | | AdaMine | **1.0 ± 0.1** | **40.2 ± 1.6** | **68.1 ± 1.2** | **78.7 ± 1.3** |
| 10k items | | PWC++ (best SOTA) | 35.0 ± 0.9 | 6.8 ± 0.2 | 21.5 ± 0.2 | 28.8 ± 0.3 |
| | Model scenarios | AdaMine_sem | 205.4 ± 3.2 | 1.4 ± 0.1 | 5.4 ± 0.2 | 9.1 ± 0.4 |
| | | AdaMine_ins | 15.8 ± 0.7 | 12.3 ± 0.3 | 31.1 ± 0.5 | 41.7 ± 0.6 |
| | | AdaMine_ins+cls | 15.2 ± 0.4 | 12.9 ± 0.3 | 31.8 ± 0.3 | 42.5 ± 0.2 |
| | | AdaMine_avg | 24.0 ± 0.6 | 9.2 ± 0.4 | 25.4 ± 0.5 | 35.3 ± 0.4 |
| | | AdaMine_ingr | 53.8 ± 0.7 | 5.8 ± 0.3 | 17.3 ± 0.2 | 25.0 ± 0.2 |
| | | AdaMine_instr | 39.2 ± 0.7 | 5.7 ± 0.4 | 17.9 ± 0.6 | 26.6 ± 0.5 |
| | | AdaMine | **12.2 ± 0.4** | **14.8 ± 0.3** | **34.6 ± 0.3** | **46.1 ± 0.3** |

for **AdaMine_avg**, and 13.2 and 12.2 for **AdaMine**, an improvement of roughly 46.34% and 49.17%.

• **AdaMine** combines the information coming from the image and all the parts of the recipe (instructions and ingredients), attaining high scores. When compared to the degraded models **AdaMine_ingr** and **AdaMine_instr**, we conclude that both textual information are complementary and necessary for correctly identifying the recipe of a plate. While **AdaMine** achieves MedRs of 13.2 and 12.2 (lower is better), the scenarios without instructions or without ingredients achieve 52.8 and 53.8, and 39.0 and 39.2, respectively.

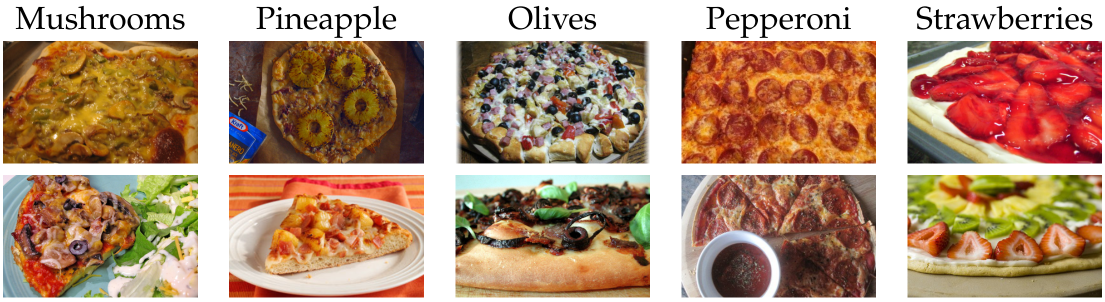


Figure 2.9 – Illustration of the ingredient-to-image retrieval ability. Within the class *Pizza*, we provide two images from the top 20 results when searching for an ingredient such as *Mushrooms*, *Pineapple*, *Olives*, *Pepperoni* and *Strawberries*.

### 2.4.3 Qualitative study on downstream tasks

In this subsection, we discuss the potential of our model for promising cooking-related application tasks. We particularly focus on downstream tasks in which the current setting might be applied. We provide illustrative examples issued from the testing set of our evaluation process. For better readability, we always show the results as images, even for text recipes for which we display their corresponding original picture.

**Ingredient To Image** An interesting ability of our model is to map ingredients into the latent space. For instance, it can retrieve recipes containing specific ingredients that could be visually identified. This is particularly useful to get a list of recipes given available aliments from a fridge. To demonstrate this process, we create each recipe query as follows: 1) for the ingredients part, we use a single word which corresponds to the ingredient we want to retrieve; 2) for the instructions part, we use the average of the instruction embeddings over all the training set. Then, we project our query into the multi-modal space and retrieve the nearest neighbors among 10,000 images randomly picked from the testing set. We show on Figure 2.9 examples of retrieved images when searching for different ingredients while constraining the results to the class *pizza*. Searching for *pineapple* or *olives* results in different types of pizzas. An interesting remark is that searching for *strawberries* inside the class *pizza* yields images of *fruit pizza* containing strawberries, *i.e.*, images that are visually similar to pizzas while containing the required ingredient. This shows the fine-grain structure of the latent space in which recipes and images are organized by visual or semantic similarity inside the different classes.

**Textual query**

Set of ingredients

- Oregano
- Zucchini
- Tofu
- Bell pepper
- Onions
- ~~**Broccoli**~~
- Olive Oil

List of instructions

1. Cut all ingredients into small pieces.
2. ~~Put **broccoli** in hot water for 10 min.~~
3. Heat olive oil in pan and put oregano in it.
4. Put cottage cheese and saute for 1 minute.
5. ~~Put onion, bell pepper, **broccoli**, zucchini.~~
6. Put burnt chilli garlic dressing with salt.
7. Saute for 1 minutes.

**Top 4 retrieved images**

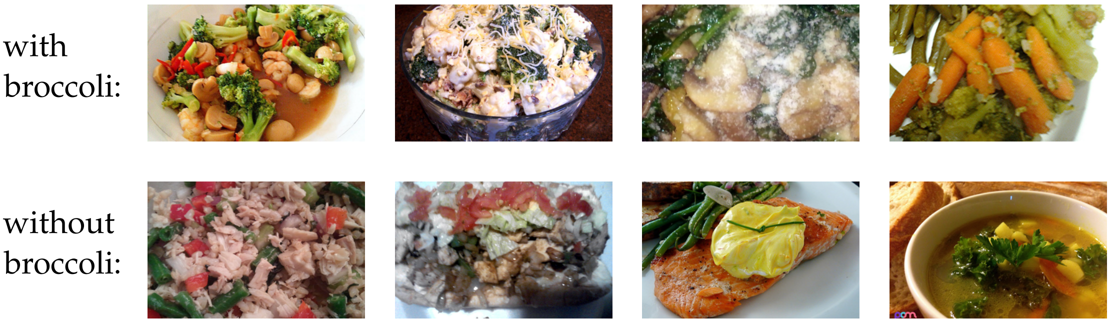


Figure 2.10 – Illustration of the latent space consistency by removing ingredients. We display the four most similar images to a recipe with (top row) or without (bottom row) broccoli in the set of ingredients. Instructions containing the targeted ingredients are also removed.

**Removing ingredients** The capacity of finely model the presence or absence of specific ingredients may be interesting for generating menus, especially for users with dietary restrictions (for instance, peanut or lactose intolerance, or vegetarians and vegans). To do so, we randomly select a recipe having *broccoli* in its ingredients list (Figure 2.10, first column) and retrieve the top 4 closest images in the embedding space from 1000 recipe images (Figure 2.10, top row). Then we remove the *broccoli* in the ingredients and remove the instructions having the *broccoli* word. Finally, we retrieve once again the top 4 images associated with this "modified" recipe (Figure 2.10, bottom row). The retrieved images using the original recipe have broccoli, whereas the retrieved images using the modified recipe do not have broccoli. This reinforces our previous statement, highlighting the ability of our latent space to correctly discriminate items with respect to ingredients.

**Images retrieval in the multi-modal space** The first task relies on multi-modal retrieval, for which a user requests items in any available format given a query

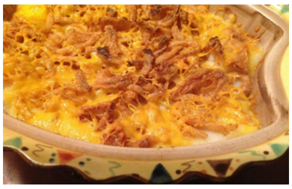
| | **Ingredients** | **Cooking instructions** | **Image** |
|---|---|---|---|
| Crunchy Onion Potato Bake | *Milk, Water, Butter, Mashed potatoes, Corn, Cheddar cheese, French-fried onions* | *Preheat oven to 350 degrees Fahrenheit. Spray pan with non stick cooking spray. Heat milk, water and butter to boiling; stir in contents of both pouches of potatoes; let stand one minute. Stir in corn. Spoon half the potato mixture in pan. Sprinkle half each of cheese and onions; top with remaining potatoes. Sprinkle with remaining cheese and onions. Bake 10 to 15 minutes until cheese is melted. Enjoy !* |  |

Figure 2.11 – Query used for the visualization provided in Figure 2.12. From left to right: a recipe category, a set of ingredients, a list of cooking instructions, an image associated with the recipe.

item in a specific format. This results in image-to-text, text-to-image, text-to-text, and image-to-image retrieval scenarios, referred to as *"multi-modal retrieval"*. In the long term, this could be useful when the user needs the recipe of a meal eaten in a restaurant or identifying similar recipes if they would like to replace a meal in their menu. In our case, solving this task leads to retrieving the most similar items in the semantic space (*i.e.*, items with the smallest distances). For illustrating our intuition, we test the four retrieval scenarios on the query shown in Figure 2.11.

Regarding the *image-to-image* scenario (see Figure 2.12), we can see that the top retrieved images look similar to the query image not only in terms of colors, shapes, and textures, but also semantically. For instance, the first, third and fourth images have gratin cheese on top, and the second image also has a plate that looks similar to the one from the query. When looking at their corresponding recipe, all five include a similar set of ingredients containing potatoes, milk, butter, cheese, and onion. Small variations in the ingredients are observed, for example, the second image has rice instead of potatoes. As for the instructions, all shown results are baked in a 350 degrees Fahrenheit oven for 15 to 45 minutes depending on the recipe.

In the *image-to-recipe* scenario (see Figure 2.12), most of the results are shared with the image-to-image search which indicates that the embeddings of matching image-recipe pairs are very close. However, we also obtain results that are less visually similar, but are close to the recipe associated with the query, either in terms of ingredients or in cooking instructions.

Starting from a text recipe query, *recipe-to-image* in Figure 2.12 shows the retrieved pictures. We are able to find images similar to the picture associated with the query recipe, although no visual information was used for the querying.

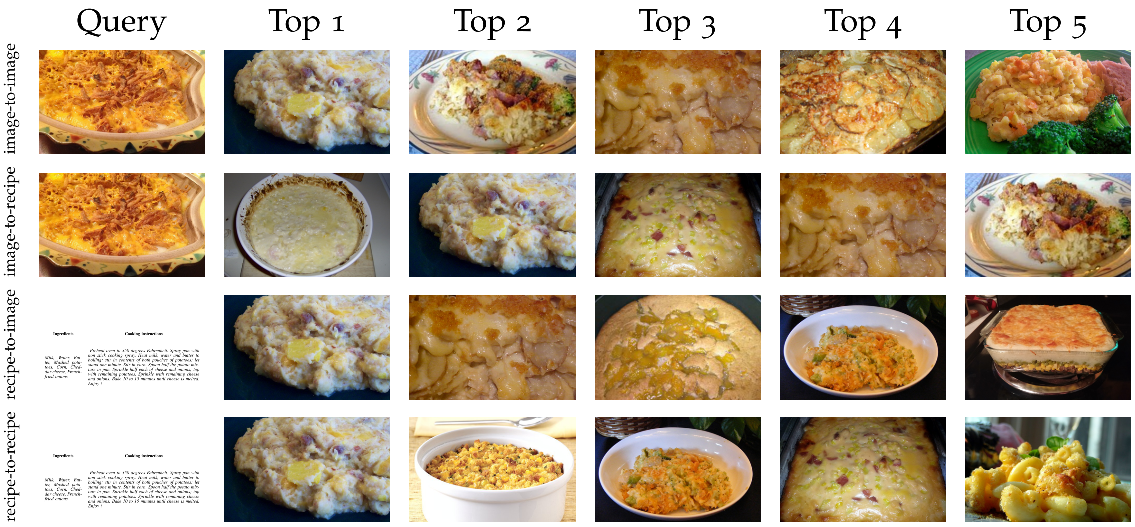

Query
Top 1
Top 2
Top 3
Top 4
Top 5
image-to-image
image-to-recipe
recipe-to-image
recipe-to-recipe
Ingredients
Cooking instructions
Milk, Water, Butter, Mashed potatoes, Corn, Cheddar cheese, French-fried onions
Preheat oven to 350 degrees Fahrenheit. Spray pan with non stick cooking spray. Heat milk, water and butter to boiling; stir in contents of both pouches of potatoes; let stand one minute. Stir in corn. Spoon half the potato mixture in pan. Sprinkle half each of cheese and onions; top with remaining potatoes. Sprinkle with remaining cheese and onions. Bake 10 to 15 minutes until cheese is melted. Enjoy !
Ingredients
Cooking instructions
Milk, Water, Butter, Mashed potatoes, Corn, Cheddar cheese, French-fried onions
Preheat oven to 350 degrees Fahrenheit. Spray pan with non stick cooking spray. Heat milk, water and butter to boiling; stir in contents of both pouches of potatoes; let stand one minute. Stir in corn. Spoon half the potato mixture in pan. Sprinkle half each of cheese and onions; top with remaining potatoes. Sprinkle with remaining cheese and onions. Bake 10 to 15 minutes until cheese is melted. Enjoy !

Figure 2.12 – Visualization of four different search processes over the multimodal space. Both image and textual recipe used as query come from the same image-text pair. First row, the image-to-image search where an image is used as query to retrieve the five most similar images. Second row, the image-to-recipe search where the five most similar textual recipes are retrieved and their associated images are shown. Third row, the recipe-to-image search where a recipe is used as query and the images are retrieved. Fourth row, the recipe-to-recipe search where an image is used as query and the recipes are retrieved.

Finally, the *recipe-to-recipe* scenario (see Figure 2.12) highlights that although the ingredients and the cooking instructions of the retrieved recipes are similar to those of the query, we observe more visual diversity among the results.

### 2.4.4 Implementation details

**Software, hardware and pretrained models** We use a single NVidia Titan X Pascal to learn our model. A single experiment lasts for 30 hours. We also improved the efficiency of the **PWC** baseline, initially implemented in Torch. From four NVidia Titan X Pascal for 3 days to run a single experiment, we reach the same performance on a single GPU for 30 hours of training only. Our code and pretrained models can be found on github:

- `github.com/Cadene/recipe1m.bootstrap.pytorch`

**Hyper-parameter choices** Our model **AdaMine** is a combination of the adaptive bidirectional instance and semantic triplet losses. Its margin $\alpha$ and the weight $\lambda$ for the semantic cost $\mathcal{L}_{sem}$ are determined using cross-validation with values varying between 0.1 and 1, and step of 0.1. We finally retained 0.3 for both $\alpha$ and $\lambda$. The parameter $\lambda$ was further analyzed in Section 2.4.2 and in Figure 2.8.

**Triplet sampling** As is common with triplet based losses in deep learning, we adopt a per-batch sampling strategy for estimating $\mathcal{L}_{ins}$ and $\mathcal{L}_{sem}$ (see Section 2.3.3). The set of multi-modal (image-recipe) matching pairs in the train (resp. validation) set are split in 2383 (resp. 513) mini-batches of 100 pairs. Following the dataset structure in which half of the pairs are not labeled by class meta-data, those 100 pairs are split into: 1) 50 randomly selected pairs among those not associated with class information; 2) 50 labeled pairs for which we respect the distribution over all classes in the training set (resp. validation set).

Within each mini-batch, we then build the set of double-triplets fitting with our joint retrieval and semantic loss functions. Each item in the 100 pairs is iteratively seen as the query. The main issue is to build positive and negative sets with respect to this query. For the retrieval losses, the item in the other modality associated with the query is assigned to the positive set while the remaining items in the other modality (namely, 99 items) are assigned to the negative instance set. For the semantic loss, we randomly select, as the positive set, one item in the other modality that does not belong to the matching pair while sharing the query class. For the negative set, we consider the remaining items in the other modality that do not belong to the query class. For a fair comparison between queries over the mini-batch, we limit the size of the negative sets over each query to the smallest negative ensemble size inside the batch.

**Optimization process** As adopted by Salvador et al. 2017, we use the Adam (Kingma et al. 2014) optimizer with a learning rate of $10^{-4}$. Besides, we propose a simpler training scheme: At the beginning of the training phase, we freeze the ResNet-50 weights, optimizing only the text-processing branch, as well as the weights of the mapping of the visual processing branch. After 20 epochs, the weights of the ResNet-50 are unfrozen and the whole architecture is fine-tuned for 60 more epochs. For the final model selection, we evaluate the MedR on the validation set at the end of each training epoch, and we keep the model with the best MedR on validation.

## 2.5 Conclusion

We tackled one of the core problems in multimodal learning which is to connect the visual and textual modalities to perform crossmodal retrieval. Our target application consisted in a large-scale search engine for cooking oriented retrieval tasks (image-to-recipe, and recipe-to-image). We proposed the AdaMine approach which is based on a novel metric learning scheme to learn a crossmodal similarity function. We introduced a joint retrieval and classification learning framework based on crossmodal triplet losses to align both modalities in the same representation space. Contrarily to proceedings approaches, the semantic information is directly injected to structure the retrieval space. This allows refining the multimodal latent space by limiting the number of parameters to be learned. To tackle the issue of gradient vanishing in each triplet loss, we proposed an adaptive strategy for triplet mining.

We validated our approach on Recipe1M, the largest English dataset of nearly 1 million pictures of dishes and their recipes. Without any computing overhead, AdaMine provided significant improvements over the best state-of-the-art approaches[2]. To ensure the correctness of our evaluation protocol, we reproduced state-of-the-art performances and even provided a stronger baseline based on a positive margin. We also validated our contributions by evaluating the performances of different ablated models. Finally, we provided qualitative studies on several downstream tasks to better illustrate the ability of AdaMine to connect textual concepts with visual ones.

[2] at the time of submission

Chapter 3

# MULTIMODAL FUSIONS AND ARCHITECTURES FOR VISUAL QUESTION ANSWERING

## Contents

### *Chapter abstract*

*In Visual Question Answering (VQA), a major challenge consists in merging the vision and language modalities in order to answer a question about an image. We introduce a theoretically grounded multimodal fusion framework based on factorized bilinear models. We derive from this framework two fusion modules, MUTAN and BLOCK. These fusion modules model fine and rich interactions between the image and the question while maintaining a tractable number of free parameters.*

*Another important challenge consists in incorporating the right inductive biases and priors into reasoning architectures. We move away from the*

*classical multi-glimpse attention architecture in order to propose a relational and iterative architecture. It allows taking the compositional nature of the questions into account and to make the multimodal representations more aware of the visual and textual context. We leverage our fusion modules to sequentially combine the two modalities as well as to model the pairwise relationships between visual regions in the context of the question.*

*The work in this chapter, in collaboration with Hedi Ben-Younes, has led to the publication of three conference papers and two workshop papers:*

- Hedi Ben-Younes*, Rémi Cadène*, Nicolas Thome, and Matthieu Cord (2017b). "MUTAN: Multimodal Tucker Fusion for Visual Question Answering". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. URL: https://arxiv.org/abs/1705.06676

- Hedi Ben-Younes, Rémi Cadène, 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 (AAAI)*. URL: https://arxiv.org/abs/1902.00038

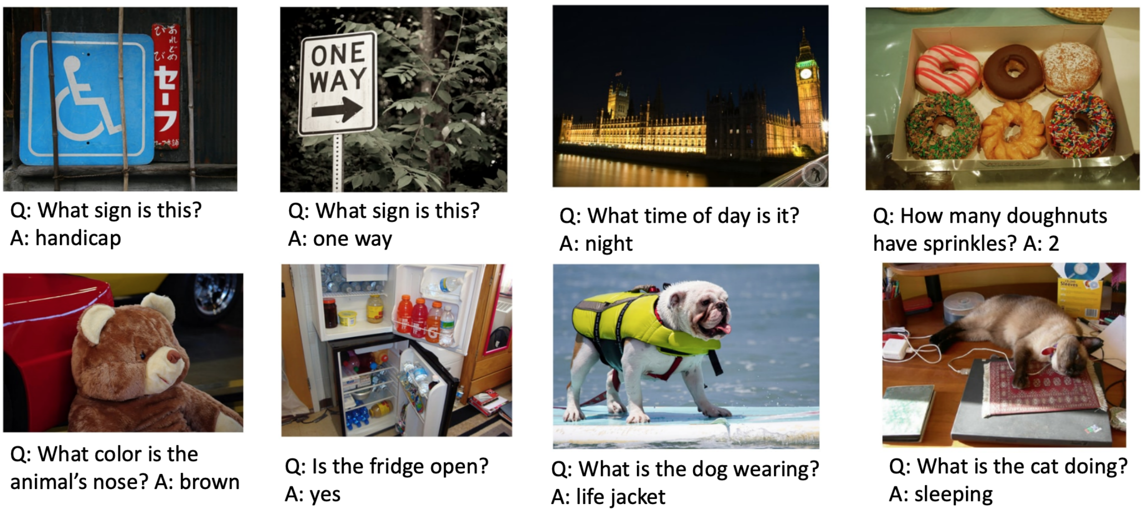
- Rémi Cadène*, Hedi Ben-Younes*, Nicolas Thome, and Matthieu Cord (2019). "MUREL: Multimodal Relational Reasoning for Visual Question Answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. URL: https://arxiv.org/abs/1902.09487

- Hedi Ben-Younes*, Remi Cadene*, Nicolas Thome, and Matthieu Cord (2017a). "VQA Challenge Workshop: MUTAN 2.0". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). VQA Challenge and Visual Dialog Workshop*

- Hedi Ben-Younes*, Remi Cadene*, Nicolas Thome, and Matthieu Cord (2018). "VQA Challenge Workshop: Bilinear Superdiagonal Fusion". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). VQA Challenge and Visual Dialog Workshop*

## 3.1 Introduction

In this chapter, we tackle the multimodal fusion problem in the context of Visual Question Answering (VQA) (Malinowski et al. 2014a; Antol et al. 2015). As illustrated in Figure 3.1, VQA consists in answering a question about the visual content of an image. To produce the correct answer, a system would need to understand a very large set of human concepts, to ground the concepts behind words in the visual space, to decompose the question into sub-tasks and to address them one after the other.



Q: What sign is this? A: handicap

Q: What sign is this? A: one way

Q: What time of day is it? A: night

Q: How many doughnuts have sprinkles? A: 2

Q: What color is the animal's nose? A: brown

Q: Is the fridge open? A: yes

Q: What is the dog wearing? A: life jacket

Q: What is the cat doing? A: sleeping

Figure 3.1 – Examples of image-question-answer triplets that are used to build and evaluate VQA systems.

VQA is considered as a visual Turing test (Malinowski et al. 2014b) aiming at evaluating the advances in visual understanding. Solving this task could have a tremendous impact on real-world applications such as aiding visually impaired users in understanding their physical and online surroundings (Gurari et al. 2018). It could also set the basis for more natural language interfaces to access multimedia data. Furthermore, VQA can be extended to dialog settings (Das et al. 2017) and thus could have an impact on the quality of smart assistants.

The current best approaches on this task are based on deep multimodal neural networks. Their parameters are optimized over a large-scale training set composed of numerous image-question-answer triplets (M. Ren et al. 2015; Y. Zhu et al. 2016; Antol et al. 2015; Goyal et al. 2017; Kafle et al. 2017) A core component to their ability to provide the appropriate answer is their multimodal fusion module (Antol et al. 2015; Fukui et al. 2016; J.-H. Kim et al. 2017). At some point in the modeling process, both modalities need to be fused into a multimodal representation that encodes high-level interactions between heterogeneous concepts.

Another critical component of the VQA models is the reasoning architecture, or in other words, the inductive biases and priors in the modeling. Previous VQA models for real data struggled to integrate an explicit reasoning process while keeping competitive performances (Antol et al. 2015). Instead, they often relied on an attentional framework which reduces the reasoning to a selection of visual regions and sometimes words that are relevant to answer the question (Fukui et al. 2016; Ben-Younes* et al. 2017b; J.-H. Kim et al. 2017; Z. Chen et al. 2017).

We tackle the VQA task by introducing two main contributions. First, we propose a theoretically grounded multimodal fusion framework based on factorized bilinear models. We derive from this framework two fusion modules: MUTAN

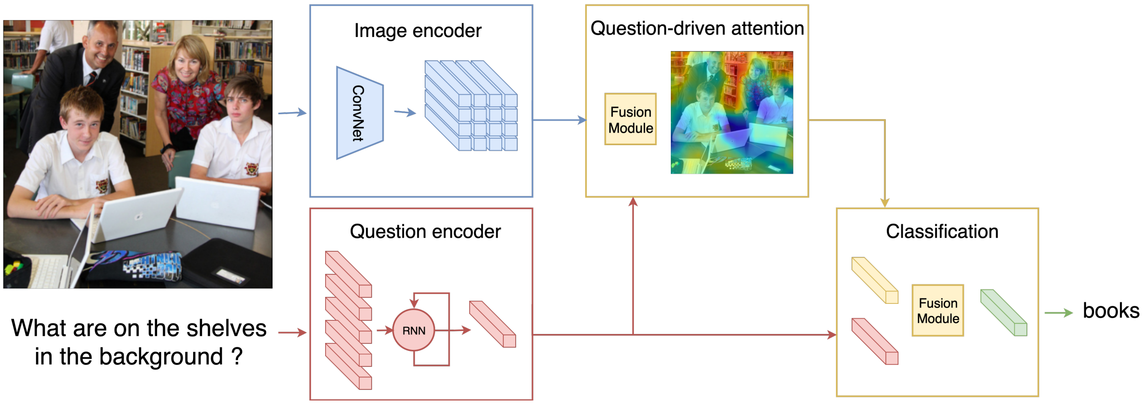


Figure 3.2 – The question-driven attentional neural architecture for VQA. It is based on the prior that the question can be fused with each visual region to determine which regions are useful to answer the question.

and BLOCK. Each of them allows modeling fine and rich interactions between the image and the question while maintaining a tractable number of free parameters. We validate the superiority of our fusion modules against state-of-the-art solutions commonly used in VQA.

Secondly, we propose a multimodal relational architecture called MuRel network that fuses the image and question modalities in a step-by-step process. We call its atomic reasoning primitive the MuRel cell. It produces vector representations of the visual regions fused with the question. These representations explicitly contain contextual information about the relations between the different regions. We apply the MuRel cell iteratively to progressively refine the multimodal representations. We also evaluate them on the most recent and commonly used VQA datasets.

## 3.2 Related work

VQA (Malinowski et al. 2014a; Antol et al. 2015) is formally tackled as a classification problem where the goal is to optimize the parameters $\theta$ of a model so that the predicted answer $\hat{a}$ matches the correct answer $a^\star$ for each image-question item. More formally,

$$\hat{a} = \arg\max_{a \in d_a} p_\theta \left(a|v, q\right) \tag{3.1}$$

As illustrated in Figure 3.2, VQA architectures leverage fusion modules that merge a vector $\mathbf{v} \in \mathbb{R}^{d_v}$ and a vector $\mathbf{q} \in \mathbb{R}^{d_q}$ to produce a vector $\mathbf{y} \in \mathbb{R}^{d_a}$. In Section 3.2.1, we review the state-of-the-art fusion modules. In Section 3.2.2, we review the VQA architectures in which these fusion modules are embedded.

### 3.2.1 Multimodal fusion modules

**Linear models** A naive way to fuse $\mathbf{v}$ and $\mathbf{q}$ is through a concatenation. Then a linear model composed of a weight matrix $W \in \mathbb{R}^{(d_q+d_v)\times d_a}$ learned on top of it to model first-order interactions between each modality. We can mathematically express the linear models as follows:

$$\mathbf{y} = \boldsymbol{W}[\mathbf{v}, \mathbf{q}] \tag{3.2}$$

That way, one dimension of the resulting multimodal vector representation $\mathbf{y}$ is obtained by a sum of each dimension of $\mathbf{v}$ and $\mathbf{q}$ weighted by the corresponding parameters of the linear model. For instance, the IMG+BOW model proposed by M. Ren et al. 2015 fuses a global image representation and a vector representation of the question using concatenation and a linear model. A limitation of the linear fusion models is their inability to model second-order interactions between the two modalities such as a simple logic AND gate.

**Multi-layers neural networks** A more expressive alternative consists in learning a fully connected neural network on top of the concatenation of the two vectors $\mathbf{v}$ and $\mathbf{q}$. For instance, the BoW Q+I model proposed by Antol et al. 2015 extends the IMG+BOW model (M. Ren et al. 2015) by learning two fully connected layers with tanh non-linearity. An intuitive limitation of this kind of fusion model is the lack of priors in the modeling.

**Bilinear models** As illustrated in Figure 3.3, a more powerful way to model interactions between two vectors $\mathbf{v}$ and $\mathbf{q}$ consists in learning a bilinear model. We can mathematically express the bilinear fusion model as follows:

$$\mathbf{y} = (\mathcal{T} \times_1 \mathbf{v}) \times_2 \mathbf{q} \tag{3.3}$$

where $\mathcal{T} \in \mathbb{R}^{d_q \times d_v \times d_a}$ and the operator $\times_i$ is the *i-mode* product between a tensor and a matrix (here a vector). Each component of $\mathbf{y}$ is a quadratic form of the inputs: $\forall k \in [1, d_a]$,

$$y_k = \sum_{i=1}^{d_q} \sum_{j=1}^{d_v} \mathcal{T}_{ijk} q_i v_j \tag{3.4}$$

Learning every parameter of $\mathcal{T}$ is only tractable when the dimensions of $\mathbf{v}$ and $\mathbf{q}$ are low enough. Because of today's hardware limitation, going higher than one hundred becomes possible when $\mathcal{T}$ is factorized or specifically structured to reduce memory consumption and compute.

A common structure is a sparse diagonalization which requires $\mathcal{T}$ to be cubical such that $d_q = d_v = d_a$. Its elements can be defined such as $\forall (i, j, k) \in \{1, d_q\}^3$:

$$\begin{aligned} \text{if } i = j = k, \text{ then } \mathcal{T}_{ijk} &= 1 \\ \text{else } \mathcal{T}_{ijk} &= 0 \end{aligned} \tag{3.5}$$

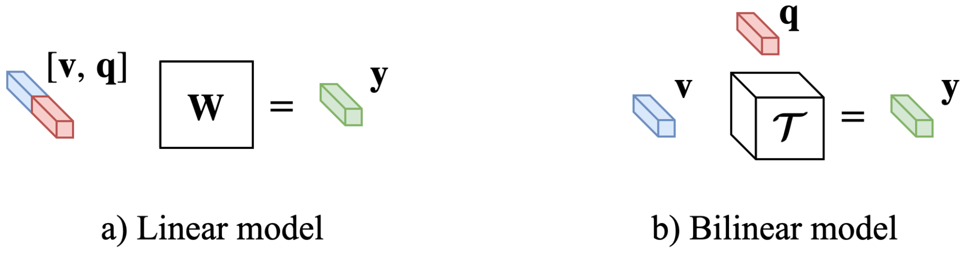


Figure 3.3 – The linear model learns first-order interactions between the two input modalities which are concatenated, whereas the bilinear model learns second-order interactions.

With this structure, Equation 3.3 simplifies such as:

$$\mathbf{y} = \mathbf{v} * \mathbf{q} \tag{3.6}$$

where $*$ is the element-wise product between two vectors. This structure dramatically reduces memory consumption and compute allowing for much higher dimensionality of input vectors. For instance, the Long-Short Term Memory (LSTM) Q + norm I model from Antol et al. 2015 as well as the models proposed by J.-H. Kim et al. 2016, R. Li et al. 2016 and J.-H. Kim et al. 2017 use an element-wise product to fuse the two vectors $\mathbf{v}$ and $\mathbf{q}$ before learning at least one linear embedding. Another efficient fusion for VQA that can be expressed as a structuration of the tensor $\mathcal{T}$ is the Multimodal Compact Bilinear pooling (MCB) (Fukui et al. 2016). It leverages the count-sketch projection (Charikar et al. 2002) to project the outer product $\mathbf{q} \otimes \mathbf{v}$ on a lower-dimensional space. While these fusions allow for larger input dimensions, they also highly constraint the ability of modeling interactions between modalities.

### 3.2.2 Neural network architectures

**Visual representations** The type and quality of the visual representations have a large impact on performances and on the kind of VQA architectures that can be used. Early works (Malinowski et al. 2015; Antol et al. 2015) have been using vector representations extracted from pretrained convolutional neural networks on ImageNet (Russakovsky et al. 2015b) such as VGG16 (Simonyan et al. 2015) or ResNet152 (Simonyan et al. 2015). Later, fixed-grid representations computed from the same Convolutional Neural Network (ConvNet) using upscaled images have been found to perform better. They allowed for region-based modeling. More recently, the object-based representations called *bottom-up features* have improved the performances furthermore (Anderson et al. 2018; Yu Jiang* et al. 2018). They are produced by a Faster-RCNN pretrained on the Visual Genome dataset (Krishna et al. 2017) to detect the bounding boxes of objects, their classes and their attributs. Finally, both fixed-grid and object-based representations can be combined to reach higher accuracies (Yu Jiang* et al. 2018).

**Question representations** Similarly, the question representations have a large impact on performances. VQA approaches commonly represent each word using embedding matrices that are pretrained with word2vec (Tomas Mikolov et al. 2013b) or GloVe (Pennington et al. 2014). (M. Ren et al. 2015) represents the question by summing each word embeddings. Other approaches learn a more powerful representation using Recurrent Neural Network (RNN) such as a LSTM (Antol et al. 2015; Fukui et al. 2016; J.-H. Kim et al. 2016; R. Li et al. 2016) proposed by Hochreiter et al. 1997. Notably, J.-H. Kim et al. 2017 use a Gated Recurrent Unit (GRU) (Chung et al. 2014) pretrained with the *skip-thought* method (Kiros et al. 2015).

**Architectures for real datasets** An intuitive way to model the iterative process of answering a question is to use a bimodal LSTM (M. Ren et al. 2015; Antol et al. 2015). This LSTM takes both the visual and word representations until the sequence of words has been fully encoded in its internal state. Orthogonally, a large set of approaches rely on a soft attention mechanism (Bahdanau et al. 2015; Xu et al. 2015). Shih et al. 2016 propose to calculate a similarity score between the representations of each image region and the concatenation of the question and the answer. These scores are used to weight the multimodal representations associated with each region before doing an averaging operation. J.-H. Kim et al. 2017; Z. Yu et al. 2017; Z. Yu et al. 2018 calculate a similarity score between the representations of each image region and the question, before doing a weighted sum over the visual representations. Finally, the resulting representations are fused with the question. This question-driven attention process can be done in parallel with an aggregation of multiple glimpses of attention (Fukui et al. 2016), or can be done sequentially (Yang et al. 2016). Lu et al. 2016a extend this approach by proposing a sequence of co-attention between the image regions and each word. A bilinear attentional architecture (J.-H. Kim et al. 2018) has also been proposed to simultaneously focus over regions and word tokens. More complex attention strategies have been explored such as graphical structures between regions (Z. Chen et al. 2017; Zhang et al. 2018; Norcliffe-Brown et al. 2018). However, they do not reach state-of-the-art performances of simpler attentional models based on *bottom-up features*.

**Architectures for toy datasets** The research efforts towards VQA models that are able to reason about a visual scene is mainly conducted using the CLEVR dataset (Johnson et al. 2017a). This artificial dataset provides questions that require spatial and relational reasoning on simple images coming from a visual world with low variability. An important line of work attempts to solve this task through explicit reasoning. In such methods (Johnson et al. 2017b; R. Hu et al. 2017; Mascharka et al. 2018), a neural network reads the question and generates a program, corresponding to a graph of elementary neural operations that process the image. However, there are two major downsides to these techniques. First, their performance strongly depends on program annotations which are used to learn the program generator. Secondly, they can be matched or surpassed by simpler models that implicitly learn to reason without requiring program annotation. In particular, Featurewise Linear Modulation (FiLM) (Perez et al. 2018) modulates the visual feature map with an affine transformation whose parameters depend on the question. In more recent work, the Memory, Attention, and Composition (MAC) network (Hudson et al. 2018) draws inspiration from the Model-View-Controller paradigm to design the trainable MAC cell on which the network iterates. Finally, Santoro et al. 2017 proposed to reason over all the possible pairs of objects in the picture, thus introducing relationship modeling in VQA. These architectures for toy datasets are nonetheless far from reaching state-of-the-art performances on real datasets.

In Section 3.2.1, we showed that current fusion modules for VQA lack the ability to control the trade-off between high dimensionality of the unimodal representations and complexity of the multimodal interactions. In Section 3.3, we propose our fusion modules based on theoretically grounded factorizations of bilinear models. They are able to model complex inter-modal interactions while keeping a large enough input dimension. Our fusion modules are agnostic to the VQA architecture. We validate them in an attentional reasoning architecture illustrated in Figure 3.2. It uses the *bottom-up features* from Anderson et al. 2018 and the *skip-thought* GRU from J.-H. Kim et al. 2017. We also propose MuRel, a new reasoning architecture, which is described in Section 3.4. MuRel is a relational and iterative architecture that goes beyond the question-driven attentional reasoning. It is designed to take the compositional nature of the questions into account and to make the multimodal representations more aware of the visual and textual context. Contrarily to previous work, this contextualized relationship modeling between regions of the image allows reaching competitive performances on real datasets.

## 3.3 Our fusion modules

As seen in Section 3.1, one of the critical components of VQA models is the fusion module for merging question and image modalities. Now, we present our fusions modules which are based on bilinear models. The latter are defined by their associated tensor $\mathcal{T} \in \mathbb{R}^{I \times J \times K}$. Its number of parameters can be calculated as $d_v d_q d_a$. We propose different factorizations of the tensor $\mathcal{T}$ which aim at lowering the number of free parameters that are learned during training.

### 3.3.1 MUTAN fusion

**Tucker decomposition** In order to reduce the number of parameters and constrain the complexity of the model, we express $\mathcal{T}$ using the Tucker decomposition (Tucker 1966) as a tensor product between *factor matrices* $\boldsymbol{W}_q, \boldsymbol{W}_v$ and $\boldsymbol{W}_a$, and a *core tensor* $\mathcal{T}_c$ such as:

$$\mathcal{T} = \left(\left(\mathcal{T}_c \times_1 \boldsymbol{W}_q\right) \times_2 \boldsymbol{W}_v\right) \times_3 \boldsymbol{W}_a \tag{3.7}$$

with $\boldsymbol{W}_q \in \mathbb{R}^{d_q \times t_q}$, $\boldsymbol{W}_v \in \mathbb{R}^{d_v \times t_v}$ and $\boldsymbol{W}_a \in \mathbb{R}^{d_a \times t_a}$, and $\mathcal{T}_c \in \mathbb{R}^{t_q \times t_v \times t_a}$. Each parameter of $\mathcal{T}$ becomes a function of a restricted number of parameters such as $\forall i \in [1, d_q], j \in [1, d_v], k \in [1, d_o]$

$$\mathcal{T}[i,j,k] = \sum_{l \in [1,t_q], m \in [1,t_v], n \in [1,t_a]} \mathcal{T}_c[l,m,n] \boldsymbol{W}_v[i,l] \boldsymbol{W}_q[j,m] \boldsymbol{W}_a[k,n] \tag{3.8}$$

The number of free parameters $n_{Tucker}$ in $\mathcal{T}$ with the Tucker decomposition can be calculated as:

$$n_{Tucker} := t_v t_q t_a + d_v t_v + d_q t_q + d_a t_a \tag{3.9}$$

When the dimensionality of $t_v$, $t_q$ and $t_a$ is chosen to be low enough, the number of free parameters is significantly reduced. As an example, when the number of dimensions of $d_v$, $d_q$ and $d_a$ is 1000, $\mathcal{T}$ has 1 billion parameters. However, when the number of dimensions of $t_v$, $t_q$ and $t_a$ are 100, $\mathcal{T}$ only contains 1,300,000 free parameters. Even though, this fusion allows learning complex inter-modal interactions between the question and the image, the dimensions of their respective unimodal representations in the spaces $t_v$ and $t_q$ are constrained to be more than 10 times smaller compared to the state-of-the-art fusions.

**Sparsity constraint** We propose MUTAN to further balance between expressivity and complexity of the interactions modeling. It consists in adding a structured sparsity constraint based on the rank of the slice matrices in $\mathcal{T}_c$. As illustrated in Figure 3.4, we impose the rank of each slice to be equal to a constant $R$. Thus, we express each slice $\mathcal{T}_c[:,:,k]$ as a sum of $R$ rank-one matrices:

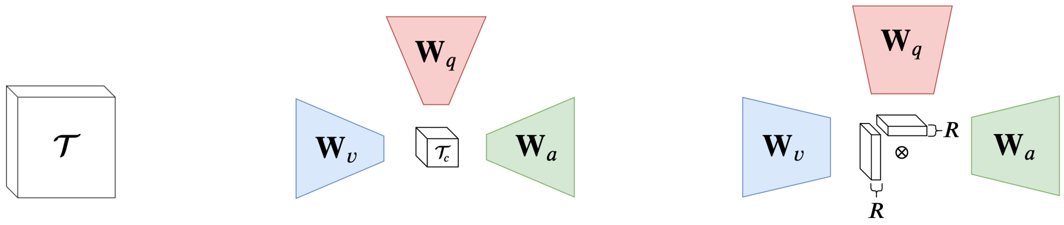
$$\mathcal{T}_c[:,:,k] = \sum_{r=1}^{R} \mathbf{m}_r^k \otimes \mathbf{n}_r^{k\top} \tag{3.10}$$

with $\mathbf{m}_r^k \in \mathbb{R}^{t_q}$ and $\mathbf{n}_r^k \in \mathbb{R}^{t_v}$.

The number of free parameters $n_{MUTAN}$ in $\mathcal{T}$ with the added sparsity constraint can be calculated as:

$$n_{MUTAN} := t_v t_a R + t_q t_a R + d_v t_v + d_q t_q + d_a t_a \tag{3.11}$$

For instance, by choosing R to be 5, we can increase the dimensions of $t_v$, $t_q$ and $t_a$ to 400 to reach 1,600,000 free parameters. This amount is comparable to what we could obtain with the Tucker fusion while allowing for higher unimodal dimensionalities.



a) Full bilinear b) Tucker c) Mutan

Figure 3.4 – The Tucker decomposition allows factorizing the tensor $\mathcal{T}$. To be tractable, this decomposition requires to project each modality in low dimensionality spaces. To reduce this constraint of dimensionality, our proposed MUTAN fusion adds an additional factorization of the core tensor $\mathcal{T}_c$ by constraining each matrix slice to be of rank $R$. It corresponds to an outer product between each matrix slice of two sub-tensors.

**Practical use** In practice, MUTAN allows avoiding calculating and storing in memory any tensor $\mathcal{T}$ or $\mathcal{T}_c$. In fact, we can decompose the bilinear fusion model from Equation 3.3 into a sequence of steps. First, input vectors $\mathbf{q}$ and $\mathbf{v}$ are projected into the unimodal spaces $\mathbb{R}^{t_q}$ and $\mathbb{R}^{t_v}$ respectively, such as:

$$\begin{aligned}\tilde{\mathbf{q}} &= \mathbf{q}^T \boldsymbol{W}_q \\ \tilde{\mathbf{v}} &= \mathbf{v}^T \boldsymbol{W}_v\end{aligned} \tag{3.12}$$

By applying the sparsity constraint, we fuse $\tilde{q}$ and $\tilde{v}$ into a multimodal vector $\mathbf{z} \in \mathbb{R}^{t_a}$, such as:

$$\mathbf{z}[k] = \sum_{r=1}^{R} (\tilde{\mathbf{q}}^T \mathbf{m}_r^k)(\tilde{\mathbf{v}}^T \mathbf{n}_r^k) \tag{3.13}$$

We then stack the rank-one matrices $\mathbf{m}_r^k$ and $\mathbf{n}_r^k$into the matrices $\mathbf{M}_r \in \mathcal{R}^{t_q \times t_a}$ and $\mathbf{N}_r \in \mathcal{R}^{t_v \times t_a}$, such that $\mathbf{M}_r[:,k] = \mathbf{m}_r^k$, and $\mathbf{N}_r[:,k] = \mathbf{n}_r^k$. This allows simplifying the previous equation such as:

$$\begin{aligned}\mathbf{z}_r &= (\tilde{q}^T \mathbf{M}_r^k) * (\tilde{v}^T \mathbf{N}_r^k) \\ \mathbf{z} &= \sum_{r=1}^{R} \mathbf{z}_r\end{aligned} \tag{3.14}$$

where $*$ is the element-wise multiplication between two vectors. Inspired by Perronnin et al. 2010 and Tsung-Yu Lin et al. 2015, we add a normalization of $\mathbf{z}_r$ that consists in a combination of signed square-root and L2-normalization to reduce sparsity in high-dimensional vectors such as:

$$\mathbf{z} = \sum_{r=1}^{R} \frac{sign(\mathbf{z}_r)\sqrt{|\mathbf{z}_r|}}{||\mathbf{z}_r||_2} \tag{3.15}$$

Finally, we project the resulting vector $\mathbf{z}$ into the answer space:

$$\mathbf{y} = \mathbf{z}^T \boldsymbol{W}_a \tag{3.16}$$

**Interpretation** We can interpret $\mathbf{z}$ as modeling an OR interaction over multiple AND gates ($R$ in MUTAN) between projections of $\tilde{\mathbf{q}}$ and $\tilde{\mathbf{v}}$. $\mathbf{z}[k]$ can be described in terms of logical operators as:

$$\mathbf{z}_r[k] = \left(\tilde{\mathbf{q}} \text{ similar to } \mathbf{m}_r^k\right) \text{ AND } \left(\tilde{\mathbf{v}} \text{ similar to } \mathbf{n}_r^k\right) \tag{3.17}$$

$$\mathbf{z}[k] = \mathbf{z}_1[k] \text{ OR ... OR } \mathbf{z}_R[k] \tag{3.18}$$

This decomposition gives a very clear insight into the logical operations that our fusion can model.

### 3.3.2 BLOCK fusion

**Block-term decomposition** Additionnaly, we propose to decompose $\mathcal{T}$ using the block-term decomposition (De Lathauwer 2008) such as:

$$\mathcal{T} := \sum_{r=1}^{R_b} \mathcal{T}_r \times_1 \boldsymbol{W}_{v_r} \times_2 \boldsymbol{W}_{q_r} \times_3 \boldsymbol{W}_{a_r} \tag{3.19}$$

where $\forall r \in [1, R_b]$, $\mathcal{T}_r \in \mathbb{R}^{t_v \times t_q \times t_a}$, $\boldsymbol{W}_{v_r} \in \mathbb{R}^{d_v \times t_v}$, $\boldsymbol{W}_{q_r} \in \mathbb{R}^{d_q \times t_q}$ and $\boldsymbol{W}_{a_r} \in \mathbb{R}^{d_a \times t_a}$. This decomposition is called *block-term* because it can be written as

$$\mathcal{T} = \mathcal{T}^{bd} \times_1 \boldsymbol{W}_v \times_2 \boldsymbol{W}_q \times_3 \boldsymbol{W}_a \tag{3.20}$$

where $\boldsymbol{W}_v = [\boldsymbol{W}_{v_1}, ..., \boldsymbol{W}_{v_{R_b}}]$ (same for $\boldsymbol{W}_q$ and $\boldsymbol{W}_a$), and $\mathcal{T}^{bd} \in \mathbb{R}^{d_v t_v \times d_q t_q \times d_a t_a}$ the block-superdiagonal tensor of $\{\mathcal{T}_r\}_{1 \leq r \leq R_b}$. Note that this decomposition is equivalent to the Tucker decomposition when R is set to 1.

To simplify the calculus, let us consider a block-term decomposition where $\forall rin[1, R_b]$, $\mathcal{T}_r$, $\boldsymbol{W}_{v_r}$, $\boldsymbol{W}_{q_r}$ and $\boldsymbol{W}_{a_r}$ have the same dimensionality. The number of free parameters $n_{BlockTerm}$ in $\mathcal{T}$ can be calculated as:

$$n_{BlockTerm} := \frac{t_v t_q t_a}{R_b^2} + d_v t_v + d_q t_q + d_a t_a \tag{3.21}$$

This decomposition allows for more fine-grained control over the complexity of the inter-modal interactions and the size of each unimodal projection. As illustrated in Figure 3.5, $\mathcal{T}_r$, $\boldsymbol{A}_r$, $\boldsymbol{B}_r$ and $\boldsymbol{C}_r$ may have different dimensionality with respect to $r$ to allow for much higher control over these quantities.

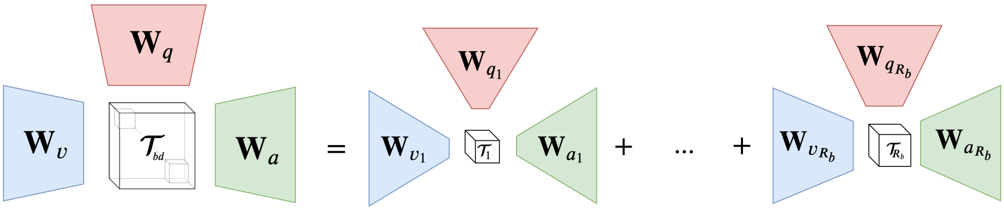


Figure 3.5 – The Block-term decomposition can be expressed as a sum of several Tucker decompositions which may have different dimensionality.

**Complexity of bilinear interactions** Multiple algebraic concepts can be used to constrain the complexity of the bilinear interactions $\mathcal{T}$. The Candecomp/PARAFAC (CP) decomposition (Carroll et al. 1970; Harshman et al. 2001) decomposes $\mathcal{T}$ such as:

$$\mathcal{T} := \sum_{r=1}^{R_{CP}} \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r \tag{3.22}$$

with $\otimes$ the outer-product between two vectors, $\mathbf{a}_r \in \mathbb{R}^{d_q}$, $\mathbf{b}_r \in \mathbb{R}^{d_v}$, and $\mathbf{c}_r \in \mathbb{R}^{d_a}$. The rank of $\mathcal{T}$ is defined by the minimal number $R_{CP}$ of triplet vectors so that Equation 3.22 is true. This decomposition is notably used in J.-H. Kim et al. 2017.

The Tucker decomposition (Tucker 1966) provides a different notion of the rank of $\mathcal{T} \in \mathbb{R}^{d_v \times d_q \times d_a}$. It is defined as a triplet of *mode-ranks* $(t_v, t_q, t_q)$ which constraint the three unfolded matrices of $\mathcal{T}$ to be of their corresponding rank, such that:

$$\begin{aligned} \text{Rank}(\mathcal{T}_{d_q d_a \times d_v}) &= t_v \\ \text{Rank}(\mathcal{T}_{d_v d_a \times d_q}) &= t_q \\ \text{Rank}(\mathcal{T}_{d_v d_q \times d_a}) &= t_a \end{aligned} \tag{3.23}$$

The Block-term decomposition (De Lathauwer 2008) generalizes the two decompositions and constraint the tensor $\mathcal{T}$ using a combination of both concepts. It allows for more fine-grained control over the complexity of the interactions between modalities. The Candecomp/PARAFAC decomposition (Carroll et al. 1970) can be seen as a special case where $t_v = t_q = t_a = 1$, reducing $\mathcal{T}^{bd}$ to a super-diagonal identity tensor. Similarly, the Tucker decomposition (Tucker 1966) can be seen as a special case of the Block-term decomposition where $R_b = 1$, constraining $\mathcal{T}^{bd}$ to be made of a single block. As illustrated in Figure 3.6, fusions based on the Block-term decomposition combine the high-dimensional monomodal projections capability of the Candecomp/PARAFAC based fusions with the rich inter-model interactions capability of the Tucker based fusions.

**Sparsity constraint** To further reduce the number of parameters in the model, we add a constraint on the rank of each third-order slice matrix of the blocks $\mathcal{T}_r$, as it was done in the MUTAN fusion. This different factorization of $\mathcal{T}$ is called BLOCK fusion.

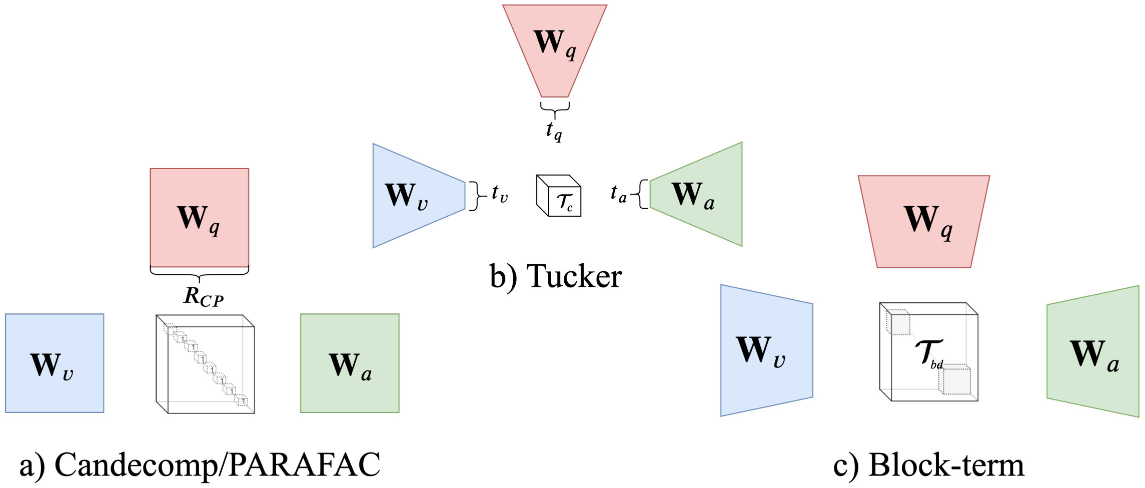


Figure 3.6 – The Candecomp/PARAFAC decomposition imposes a rank constraint over the tensor $\mathcal{T}$ by fixing the dimension of the three projections to be of size $R_{CP}$. The Tucker decomposition imposes a different rank constraint by fixing each dimension to be of size $t_v$, $t_q$ and $t_a$. The block-term decomposition generalizes these two decompositions, allowing for a fine-grained control over the complexity of the bilinear interactions.

When $\forall r \in [1, R_b]$, $\mathcal{T}_r$, $\boldsymbol{W}_{v_r}$, $\boldsymbol{W}_{q_r}$ and $\boldsymbol{W}_{a_r}$ have the same dimensionality, the number of free parameters $n_{BLOCK}$ in $\mathcal{T}$ can be calculated as:

$$n_{BLOCK} := \frac{t_v t_a R + t_q t_a R}{R_b^2} + d_v t_v + d_q t_q + d_a t_a \tag{3.24}$$

where $R$ is the rank constraint over the matrix slices of the core tensors expressed in MUTAN.

## 3.4 Our reasoning architecture

After having described our fusion modules, we propse a novel reasoning architecture for VQA, which is called MuRel, and leverages our previous fusion modules. As illustrated in Figure 3.7, MuRel iteratively merges visual information with a novel attentional strategy modeling interactions between visual regions. In Section 3.4.1, we present the MuRel network that encompasses an iterative scheme to reason about the scene with respect to a question. It is based on the MuRel cell, presented in Section 3.4.2, which is a neural module that learns to perform elementary reasoning operations by blending question information into the set of spatially-grounded visual representations.

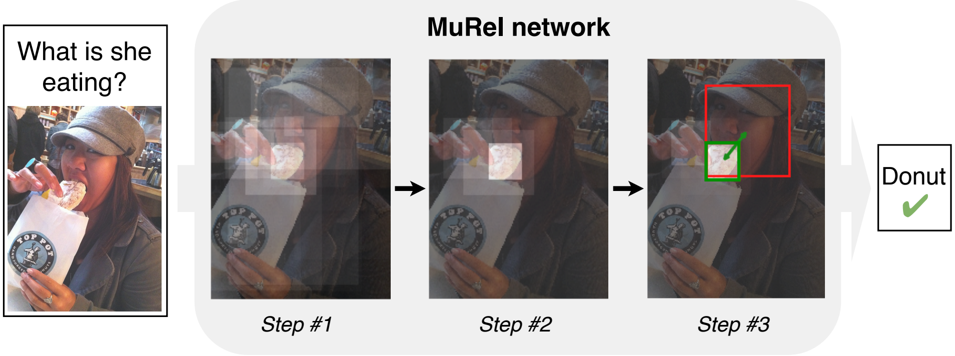


Figure 3.7 – Our MuRel network for VQA is an iterative process based on a rich vectorial representation between the question and visual information explicitly modeling pairwise region relations. MuRel is thus able to express complex analysis primitives beyond attention maps: here the two regions corresponding to the head and the donuts are selected based on their visual cues and semantic relations to properly answer the question "what is she eating?"

### 3.4.1 MuRel network

The MuRel network is a novel reasoning architecture for VQA. As illustrated in Figure 3.8, its image encoder can be instantiated by a Faster-RCNN object detector (S. Ren et al. 2015). It provides a set of vectors $\{\boldsymbol{v}_i\}_{i\in[1,N]}$, where each $\boldsymbol{v}_i \in \mathbb{R}^{d_v}$ corresponds to an object detected in the picture. It also outputs the spatial coordinates of each region $\boldsymbol{b}_i = [x, y, w, h]$, where $(x, y)$ are the coordinates of the top-left point of the box, and $h$ and $w$ correspond to the height and the width of the box. Note that $x$ and $w$ (respectively $y$ and $h$) are normalized by the width (resp. height) of the image. Its question encoder can be instantiated by a GRU. It provides a vector representation of the question $\boldsymbol{q} \in \mathbb{R}^{d_q}$. Its multimodal fusion scheme is composed of a MuRel cell which iteratively updates the region state vectors $\{\boldsymbol{s}_i\}$, each time refining the representations with contextual and question information. More specifically, for each step $t = 1..T$ where $T$ is the total number of steps, a MuRel cell processes and updates the state vectors following Equation (3.25):

$$\{\boldsymbol{s}_i^t\} = \text{MurelCell}\left(\{\boldsymbol{s}_i^{t-1}\}; \{\boldsymbol{b}_i\}, \boldsymbol{q}\right) \tag{3.25}$$

where MurelCell is detailed in Section 3.4.2. The state vectors are initialized with the features outputted by the object detector; for each region $i$, $\boldsymbol{s}_i^0 = \boldsymbol{v}_i$.

The MuRel network represents each visual region regarding the question and the visual context in which this region is located. This representation is done iteratively, through multiple steps of a MuRel cell. The residual nature of this module makes it possible to align multiple cells without being subject to gradient

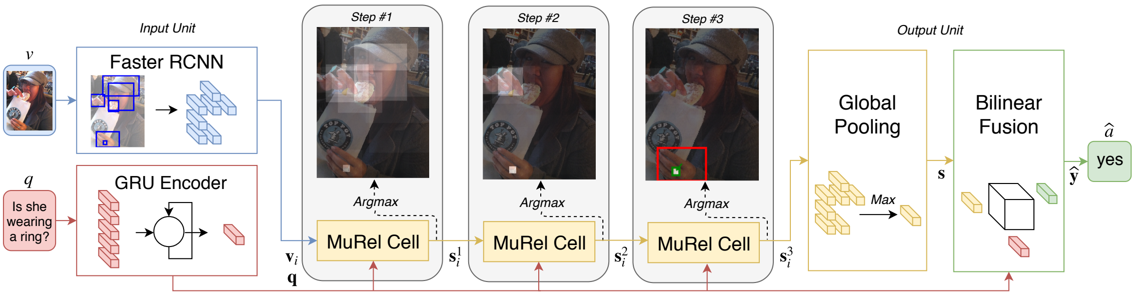


Figure 3.8 – The MuRel network merges the question embedding $\boldsymbol{q}$ into spatially-grounded visual representations $\{\boldsymbol{v}_i\}$ by iterating through a single MuRel cell. This module takes as input a set of localized vectors $\{\boldsymbol{s}_i\}$ and updates their representation using a multimodal fusion component. Moreover, it models all the possible pairwise relations between regions by combining spatial and semantic information. To construct the importance map at step $t$, we count the number of time each region provides the maximal value of $\max_i\{\boldsymbol{s}_i^t\}$ (over the 2048 dimensions).

vanishing. Moreover, the weights of our model are shared across the cells, which enables compact parametrization and good generalization.

At step $t = T$, the representations $\{\boldsymbol{s}_i^T\}$ are aggregated with a global max-pooling operation to provide a single vector $\boldsymbol{s} \in \mathbb{R}^{d_v}$. This scene representation contains information about the objects, the spatial and semantic relations between them, with respect to a particular question.

The scene representation $\boldsymbol{s}$ is merged with the question embedding $\boldsymbol{q}$ to compute a score for every possible answer $\hat{\boldsymbol{y}} = \mathrm{B}\left(\boldsymbol{s}, \boldsymbol{q}; \Theta_y\right)$, where B is a given fusion module such as BLOCK. Finally, $\hat{a}$ is the answer with a maximum score in $\hat{\boldsymbol{y}}$.

## 3.4.2 MuRel cell

The proposed MuRel cell takes as input a bag of $N$ visual features $\boldsymbol{s}_i \in \mathbb{R}^{d_v}$, along with their bounding box coordinates $\boldsymbol{b}_i$. As shown in Figure 3.9, it is a residual function consisting of two modules. First, an efficient bilinear fusion module merges question and region feature vectors to provide a local multimodal embedding. This fusion is directly followed by a pairwise modeling component, designed to update each multimodal representation with respect to its own spatial and visual context.

**Multimodal fusion** In classical attention models, the fusion between image region and question features $\boldsymbol{s}$ and $\boldsymbol{q}$ only learns to encode whether a region is relevant. In the MuRel cell, the local multimodal information is represented within

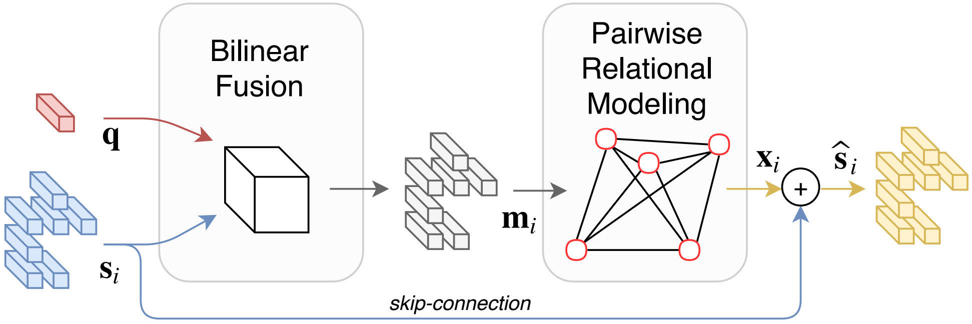


Figure 3.9 – In the MuRel cell, the bilinear fusion represents rich and fine-grained interactions between question and region vectors $\boldsymbol{q}$ and $\boldsymbol{s}_i$. All the resulting multimodal vectors $\boldsymbol{m}_i$ pass through a pairwise modeling block to provide a context-aware embedding $\boldsymbol{x}_i$ per region. The cell's output $\hat{\boldsymbol{s}}_i$ is finally computed as a sum between $\boldsymbol{s}_i$ and $\boldsymbol{x}_i$, acting as a residual function of $\boldsymbol{s}_i$.

a richer vector form $\boldsymbol{m}_i$ which can encode more complex correlations between both modalities. This allows storing more specific information about what precise characteristic of a particular region is important in a given textual context.

**Pairwise interactions** To answer certain types of questions, it can be necessary to reason over multiple objects that interact together. More generally, we want each representation to be aware of the spatial and semantic context around it. Given that our features are structured as a bag of localized vectors (Anderson et al. 2018), modeling the visual context of each region is not straightforward. Similarly to the recent work of (Norcliffe-Brown et al. 2018), we opt for a pairwise relationship modeling where each region receives a message based on its relations to its neighbors. In their work, a region's neighbors correspond to the $K$ most similar regions, whereas in the MuRel cell the neighborhood is composed of every region in the image. Besides, instead of using scalar pairwise attention and graph convolutions with Gaussian kernels as they do, we merge spatial and semantic representations to build relationship vectors. In particular, we compute a context vector $\check{e}_i$ for every region. It consists in an aggregation of all the pairwise links $\boldsymbol{r}_{i,j}$ coming into $i$. We define it as $\check{e}_i = \max_j \boldsymbol{r}_{i,j}$, where $\boldsymbol{r}_{i,j}$ is a vector containing information about the content of both regions, but also about their relative spatial positioning. We use the max operator in the aggregation function to reduce the noise that can be induced by average or sum pooling, which oblige all the regions to interact with each other. To encode the relationship vector, we use the following formulation:

$$\boldsymbol{r}_{i,j} = \mathrm{B}\left(\boldsymbol{b}_i, \boldsymbol{b}_j; \Theta_b\right) + \mathrm{B}\left(\boldsymbol{m}_i, \boldsymbol{m}_j; \Theta_m\right) \tag{3.26}$$

Through the $\mathrm{B}(.,.;\Theta_b)$ operator, the cell is free to learn spatial concepts such as *on top of, left, right, etc.* In parallel, $\mathrm{B}(.,.;\Theta_s)$ encodes correlations between multimodal vectors $(\boldsymbol{s}_i, \boldsymbol{s}_j)$, corresponding to semantic visual concepts conditioned on the question representation. By summing up both spatial and semantic fusions, the network can learn high-level relational concepts such as *wear, hold, etc.*

The context representation $\check{\boldsymbol{e}}_i$ that contains an aggregation of the messages $\boldsymbol{r}_{i,j}$ provided by its neighbors updates the multimodal vector $\boldsymbol{m}_i$ in an additive manner:

$$\boldsymbol{x}_i = \boldsymbol{m}_i + \check{\boldsymbol{e}}_i \tag{3.27}$$

This formulation of the pairwise modeling is actually closer to the Graph Networks (Battaglia et al. 2018), where the notion of relational inductive biases is formalized.

Finally, the MuRel cell's output is computed as a residual function of its input, to avoid the vanishing gradient problem. Each visual feature $\boldsymbol{s}_i$ is updated as: $\hat{\boldsymbol{s}}_i = \boldsymbol{s}_i + \boldsymbol{x}_i$.

The chain of operations that updates the set of localized region embeddings $\{\boldsymbol{s}_i\}_{i \in [1,N]}$ using the multimodal fusion with $\boldsymbol{q}$ and the pairwise modeling operator is noted:

$$\{\hat{\boldsymbol{s}}_i\} = \text{MurelCell}\left(\{\boldsymbol{s}_i\}; \{\boldsymbol{b}_i\}, \boldsymbol{q}\right) \tag{3.28}$$

## 3.5 Experiments

### 3.5.1 Experimental setup

**VQA v2 dataset** We validate our contributions on three recent datasets. First, we use VQA v2 (Goyal et al. 2017), which is the most used dataset. Its images come from the MS-COCO dataset (Tsung-Yi Lin et al. 2014). Its questions and answers have been annotated on Amazon Mechanical Turk (AMT). Its training set is composed of 248,349 image-question pairs. Its validation set is composed of 121,512 image-question pairs. Each one of these questions has been answered by 10 annotators, yielding a list of 10 ground-truth answers. Its testing set is composed of 244,302 image-question pairs and is called *test-std*. One must submit their predictions to an evaluation server to get the scores on the testing set. Note that the evaluation server makes it possible to submit ten predictions per day on *test-dev*, which is a half-size version of *test-std*. To avoid hyperparameters overfitting on the testing set, the whole submission on *test-std* can only be done five times per account. Scores reported on *test-std* or *test-std* are produced by models trained on *trainval*, which is the aggregation of the training and validation sets minus a small 5% subset used for early-stopping.

**TDIUC dataset** Then, we use the TDIUC dataset (Kafle et al. 2017), which is the current biggest VQA dataset. Its images come from the MS-COCO dataset and the VisualGenome dataset (Krishna et al. 2017). Its training set is composed of 1,157,917 image-question pairs. Its testing set is composed of 496,250 image-question pairs. We use a small 5% subset of the training set for early-stopping. Around two-thirds of its question-answer pairs have been generated using semantic knowledge attached to the images. It comes with different metrics to compensate for over-represented question-types. TDIUC allows us to construct a more detailed analysis of our model's performance on 12 well-defined types of questions.

**VQA-CP v2 dataset** Finally, we use the VQA Changing Priors v2 (VQA-CP v2) dataset (Agrawal et al. 2018). VQA-CP v2 has been built using the training and validation sets of VQA v2. It comes with different training and validation splits. Both sets possess a different distributions of answers per question-type. We use it to valide the robustness of our approach to question-based overfitting.

**Unimodal architectures** For the image encoder, we use the recent Bottom-up features provided by Anderson et al. 2018 to represent our image as a set of 36 localized regions. For the question encoder, we use the pretrained skip-thought encoder from Kiros et al. 2015.

**Training protocol** We use standard features extraction, preprocessing and loss function (Fukui et al. 2016). Inspired by recent works, we use Adam as optimizer (Kingma et al. 2014) with a learning scheduler (Yu Jiang* et al. 2018). More details can be found in Section 3.5.6.

### 3.5.2 Fusion modules validation

**Fusion analysis** In Table 3.1, we compare our MUTAN and BLOCK fusion modules against different fusion schemes available in the literature on the commonly used VQA v2 dataset (Goyal et al. 2017). We embed each fusion module in the same question-driven attentional architecture proposed by Fukui et al. 2016. For each model, we run a grid search over its hyperparameters and report the best results on the validation set. We report the size of the model, corresponding to the number of parameters between the attended image features, the question embedding, and the answer prediction. We briefly describe the different fusion schemes used for the comparison:

(1) *linear model*: the two vectors are projected on a common space, and their summation is projected to predict the answer;

(2) *multi-layer model*: the vectors are concatenated and passed at the input of a 3-layer Multi-Layer Perceptron (MLP);

Table 3.1 – Comparison of the fusion modules on VQA v2 *test-dev* set. $|\Theta|$ is the number of parameters learned in the fusion modeling. *Overall* is the overall Open Ended accuracy (higher is better). *Yes/no*, *Numbers* and *Others* are subsets that correspond to answers types.

| | Model | $\|\Theta\|$ | Overall | Answer type | | |
|---|---|---|---|---|---|---|
| | | | | Yes/No | Number | Other |
| (1) | Sum | 8M | 58.48 | 71.89 | 36.56 | 52.09 |
| (2) | Concat MLP | 13M | 63.85 | 81.34 | 43.75 | 53.48 |
| (3) | MCB (Fukui et al. 2016) | 32M | 61.23 | 79.73 | 39.13 | 50.45 |
| (4) | Tucker (Ben-Younes* et al. 2017b) | 14M | 64.21 | 81.81 | 42.28 | 54.17 |
| (5) | MLB (J.-H. Kim et al. 2017) | 16M | 64.88 | 81.34 | 43.75 | 53.48 |
| (6) | MFB (R. Yu et al. 2017) | 24M | 65.56 | 82.35 | 41.54 | 56.74 |
| (7) | MUTAN (Ben-Younes* et al. 2017b) | 14M | 65.19 | 82.22 | 42.1 | 55.94 |
| (8) | MFH (Z. Yu et al. 2018) | 48M | 65.72 | 82.82 | 40.39 | 56.94 |
| (9) | BLOCK | 18M | **66.41** | **82.86** | **44.76** | **57.3** |

(3) *bilinear model*: a bilinear interaction based on a count-sketching technique that projects the outer product of between inputs on a multimodal space (Fukui et al. 2016);

(4) *bilinear model*: a bilinear interaction where the tensor is expressed as a Tucker decomposition which corresponds to our Mutan fusion without the sparsity constraint;

(5) *bilinear model*: a bilinear interaction where the tensor is expressed as a CP decomposition (J.-H. Kim et al. 2017);

(6) *bilinear model*: a bilinear interaction where each 3rd mode slice matrix of the tensor is constrained by its rank (Z. Yu et al. 2017);

(7) *bilinear model*: our MUTAN fusion module;

(8) *higher-order model*: a higher-order fusion composed of cascaded of (6). (Z. Yu et al. 2018);

(9) *bilinear model*: our BLOCK fusion module.

From the results in Table 3.1, we see that the simple sum fusion (1) provides a very low baseline. We also note that the MLP (2) doesn't provide the best results, despite its non-linear structure. As the MLP should be able to find that two different modalities are used and that it needs to look for interactions between them, this is in practice difficult to obtain. Instead, top-performing methods are based on a bilinear model. The structure imposed on the parameters highly influences the final performance. We can see that (3), which simplifies the bilinear model using random projections, has efficiency issues due to the count-sketching technique. These issues are alleviated in the other bilinear methods, which use the tensor decomposition framework to practically implement the interaction. Our BLOCK method (9) gives the best results. As we saw, the block-term decomposition generalizes both CP and Tucker decompositions, which is why it is not surprising to see it surpass them. Moreover, the fact that it integrates the 3rd order slices rank constraint gives it the advantages of (6) and (7). Interestingly, it even surpasses (8) which is based on a higher-order interaction modeling, while using 30M fewer parameters. This strongly indicates that controlling a bilinear model through its block-term ranks provides an efficient trade-off between modeling capacities and number of parameters. To further validate this hypothesis, we evaluate a BLOCK fusion with only 3M parameters. This model obtained 64.91%. Unsurprisingly, it does not surpass all the methods against which we compare. However, it obtains competitive results, improving over 5 out of 8 methods that all use far more parameters.

**State-of-the-art comparison** We compare our best fusion module embedded in an attentional architecture against state-of-the-art approaches[1] on two datasets: the widely used VQA v2 dataset (Goyal et al. 2017) and TDIUC (Kafle et al. 2017). On this more recent dataset, evaluation metrics are provided to assess the robustness of the model with respect to answer imbalance, as well as to account for performance homogeneity across the different question types.

Table 3.2 – State-of-the-art comparison of BLOCK on the TDIUC dataset. (*) are reported from Kafle et al. 2017.

| Model | Accuracy | A-MPT | H-MPT | A-NMPT | H-NMPT |
|---|---|---|---|---|---|
| Most common answer (Kafle et al. 2017) | 51.15 | 31.11 | 17.53 | 15.63 | 0.83 |
| Question only (Kafle et al. 2017) | 62.74 | 39.31 | 25.93 | 21.46 | 8.42 |
| NMN* (Andreas et al. 2016) | 79.56 | 62.59 | 51.87 | 34.00 | 16.67 |
| MCB* (Fukui et al. 2016) | 81.86 | 67.90 | 60.47 | 42.24 | 27.28 |
| RAU* (Noh et al. 2016) | 84.26 | 67.81 | 59.00 | 41.04 | 23.99 |
| BLOCK | **85.96** | **71.84** | **65.52** | **58.36** | **39.44** |

[1] at time of submission

Table 3.3 – State-of-the-art comparison of BLOCK on VQA v2 *test-dev* set. The models were trained on the union of VQA v2 *trainval* split and VisualGenome (Krishna et al. 2017) train split. *Overall* is the overall OpenEnded accuracy (higher is better). *Yes/no*, *Numbers* and *Others* are subsets that correspond to answers types. Only single model scores are reported. (*) are reported from Goyal et al. 2017.

| Model | Overall | Answer type | | |
|---|---|---|---|---|
| | | Yes/no | Num. | Other |
| TipsAndTricks (Teney et al. 2018) | 65.32 | 81.82 | 44.21 | 56.05 |
| MFH (Z. Yu et al. 2018) | 65.80 | - | - | - |
| Counter (Zhang et al. 2018) | **68.09** | 83.14 | **51.62** | **58.97** |
| BLOCK | 67.58 | **83.6** | 47.33 | 58.51 |

As we show in Table 3.2, our model is able to outperform the preceding ones on TDIUC by a large margin for each metric, especially those which account for bias in the data. We notably report a gain of +1.7 in accuracy, +3.95 in A-MPT, +5.05 in H-MPT, +16.12 in A-NMPT, +15.45 in H-NMPT, over the best scoring model in each metric. The high results in the harmonic metrics (H-MPT and H-NMPT) suggest that BLOCK performs well across all question types, while the high scores in the normalized metrics (A-NMPT and H-NMPT) denote that our model is robust to answer imbalance type of bias in the dataset.

In Table 3.3 and Table 3.4, we see that our model obtains competitive results on VQA v2 compared to previously published approaches. Our model is notably outperformed by Counter proposed by Zhang et al. 2018. However, their approach relies on a specialized module that increases the counting ability of VQA model. We believe that our contribution is orthogonal to them. Still, our model performs better than (Teney et al. 2018) and (Z. Yu et al. 2018), with whom we share the global VQA architecture. In further detail, we point out that BLOCK surpasses (Z. Yu et al. 2018) reaching a +1.78 improvement in the overall accuracy on *test-dev*, even though the latter encompasses the current state-of-the-art fusion scheme. Furthermore, we use the same image features than Teney et al. 2018 and are able to achieve a +2.26 gain on *test-dev* and +2.25 on *test-std*.

### 3.5.3 Reasoning architecture validation

**Comparison to attention-based model** In Table 3.5, we compare MuRel against a strong multi-glimpses attentional architecture (Fukui et al. 2016) based on our BLOCK fusion module. The goal of these experiments is to compare our approach with strong baselines for real VQA in controlled conditions. In addition to using

Table 3.4 – State-of-the-art comparison of BLOCK on VQA v2 *test-std* set. The models were trained on the union of VQA v2 *trainval* split and VisualGenome (Krishna et al. 2017) train split. *Overall* is the overall OpenEnded accuracy (higher is better). *Yes/no*, *Numbers* and *Others* are subsets that correspond to answers types. Only single model scores are reported. (*) are reported from Goyal et al. 2017

| Model | Overall | Answer type | | |
|---|---|---|---|---|
| | | Yes/no | Num. | Other |
| Most common answer (Goyal et al. 2017) | 25.98 | 61.20 | 0.36 | 1.17 |
| Question only (Goyal et al. 2017) | 44.26 | 67.01 | 31.55 | 27.37 |
| Deep LSTM* (Lu et al. 2015) | 54.22 | 73.46 | 35.18 | 41.83 |
| MCB* (Fukui et al. 2016) | 62.27 | 78.82 | 38.28 | 53.36 |
| ReasonNet (Ilievski et al. 2017) | 64.61 | 78.86 | 41.98 | 57.39 |
| TipsAndTricks (Teney et al. 2018) | 65.67 | 82.20 | 43.90 | 56.26 |
| Counter (Zhang et al. 2018) | **68.41** | 83.56 | **51.39** | **59.11** |
| BLOCK | 67.92 | **83.98** | 46.77 | 58.79 |

Table 3.5 – Comparison of MuRel against a strong attention-based architecture on the VQA v2 *val*, VQA-CP v2 and TDIUC datasets. The accuracy is reported (higher is better). Both models have an equivalent number of parameters (∼60 million) and are trained on the same features following the same experimental setup.

| Model | VQA v2 | VQA CP v2 | TDIUC |
|---|---|---|---|
| Attention baseline | 63.44 | 38.04 | 86.96 |
| MuRel | **65.14** | **39.54** | **88.20** |

the same Bottom-up features, which are crucial for fair comparisons, we also dimension the attention-based baseline to have an equivalent amount of learned parameters than MuRel (∼60 million including those from the GRU encoder). Also, we train it following the same experimental setup to ensure competitiveness. MuRel reaches a higher accuracy on the three datasets. We report a significant gain of +1.70 on VQA v2 and +1.50 on VQA CP v2. Not only these results validate the ability of MuRel to better model interactions between the question and the image, but also to generalize when the distribution of the answers per question is different between the training and validation sets as in VQA CP v2. A gain of +1.24 on TDIUC demonstrates the richer modeling capacity of MuRel in a fine-grained context of 12 well-delimited question types.

Table 3.6 – State-of-the-art comparison of MuRel on the VQA v2 dataset. Results on *test-dev* and *test-std* splits. All these models were trained on the same training set (VQA v2 *train+val*), using the Bottom-up features provided by Anderson et al. 2018. No ensembling methods have been used. † have been trained by Bai et al. 2018.

| Model | *test-dev* | | | | *test-std* |
|---|---|---|---|---|---|
| | Overall | Answer type | | | Overall |
| | | Yes/no | Num. | Other | |
| Bottom-up<br>(Anderson et al. 2018) | 65.32 | 81.82 | 44.21 | 56.05 | 65.67 |
| Graph Att.<br>(Norcliffe-Brown et al. 2018) | - | - | - | - | 66.18 |
| MUTAN†<br>(Ben-Younes* et al. 2017b) | 66.01 | 82.88 | 44.54 | 56.50 | 66.38 |
| MLB†<br>(J.-H. Kim et al. 2017) | 66.27 | 83.58 | 44.92 | 56.34 | 66.62 |
| DA-NTN<br>(Bai et al. 2018) | 67.56 | 84.29 | 47.14 | 57.92 | 67.94 |
| Pythia<br>(Yu Jiang* et al. 2018) | 68.05 | - | - | - | - |
| Counter<br>(Zhang et al. 2018) | **68.09** | 83.14 | **51.62** | **58.97** | **68.41** |
| MuRel | 68.03 | **84.77** | 49.84 | 57.85 | **68.41** |

**State-of-the-art comparison on VQA v2** In Table 3.6, we compare MuRel to the most recent contributions on the VQA v2 dataset. For fairness considerations, all the scores correspond to models trained on the VQA v2 *trainval* split, using the Bottom-up visual features (Anderson et al. 2018). Interestingly, our model surpasses both MUTAN (Ben-Younes* et al. 2017b) and MLB (J.-H. Kim et al. 2017), which correspond to some of the latest development in visual attention and bilinear models. This tends to indicate that VQA models can benefit from retaining local information in multimodal vectors instead of scalar coefficients. Moreover, our model greatly improves over the recent method proposed in (Norcliffe-Brown et al. 2018) where the regions are structured using pairwise attention scores, which are leveraged through spatial graph convolutions. This shows the interest of our spatial-semantic pairwise modeling between all possible pairs of regions. Finally, even though we did not extensively tune the hyperparameters of our model, our overall score on the *test-dev* split is highly competitive with state-of-the-art

Table 3.7 – State-of-the-art comparison of MuRel on the TDIUC dataset. (*) are reported by Kafle et al. 2017.

| | RAU*<br>(Noh et al. 2016) | MCB*<br>(Fukui et al. 2016) | QTA<br>(Y. Shi et al. 2018) | MuRel |
|---|---|---|---|---|
| Bottom-up | ✗ | ✗ | ✓ | ✓ |
| Scene Reco. | 93.96 | 93.06 | 93.80 | **96.11** |
| Sport Reco. | 93.47 | 92.77 | 95.55 | **96.20** |
| Color Attr. | 66.86 | 68.54 | 60.16 | **74.43** |
| Other Attr. | 56.49 | 56.72 | 54.36 | **58.19** |
| Activity Reco. | 51.60 | 52.35 | 60.10 | **63.83** |
| Pos. Reasoning | 35.26 | 35.40 | 34.71 | **41.19** |
| Object Reco. | 86.11 | 85.54 | 86.98 | **89.41** |
| Absurd | 96.08 | 84.82 | **100.00** | 99.8 |
| Util. and Afford. | 31.58 | **35.09** | 31.48 | 21.43 |
| Object Presence | 94.38 | 93.64 | 94.55 | **95.75** |
| Counting | 48.43 | 51.01 | 53.25 | **61.78** |
| Sentiment | 60.09 | **66.25** | 64.38 | 60.65 |
| Overall (A-MPT) | 67.81 | 67.90 | 69.11 | **71.56** |
| Overall (H-MPT) | 59.00 | **60.47** | 60.08 | 59.30 |
| Overall Accuracy | 84.26 | 81.86 | 85.03 | **88.20** |

approaches[2]. In particular, we report comparable results to Pythia (Yu Jiang* et al. 2018) which won the VQA Challenge 2018. Please note that we do not report their best scoring model which improves the overall scores up to 70.01% by including multiple types of visual features and more training data. Also, we do not report the score of 69.52% obtained by BAN (J.-H. Kim et al. 2018) which is trained on extra data from the Visual Genome dataset (Krishna et al. 2017).

**State-of-the-art comparison on TDIUC** One of the core aspect of VQA models lies in their ability to address different tasks. The TDIUC dataset enables a detailed analysis of the strengths and limitations of a model by evaluating its performance on different types of questions. We show in Table 3.7 a detailed comparison of recent models to our MuRel. We obtain state-of-the-art results on the Overall Accuracy and the arithmetic mean of per-type accuracies (A-MPT), and surpass by a significant margin the second-best model proposed by Y. Shi et al. 2018. Interestingly, we improve over this model even though it uses a combination of Bottom-up and fixed-grid features, as well as a supervision on the question types (hence its 100% result on the *Absurd* task). MuRel notably surpasses all

[2] at the time of submission

Table 3.8 – State-of-the-art comparison of MuRel on the VQA-CP v2 dataset. We train the Attention model using the Bottom-up features.

| Model | Bottom-up | Overall | Answer type | | |
|---|---|---|---|---|---|
| | | | Yes/no | Num. | Other |
| HAN (Malinowski et al. 2018) | ✗ | 28.65 | 52.25 | **13.79** | 20.33 |
| GVQA (Agrawal et al. 2018) | ✗ | 31.30 | **57.99** | 13.68 | 22.14 |
| Attention | ✓ | 38.04 | 41.56 | 12.19 | 43.29 |
| MuRel | ✓ | **39.54** | 42.85 | 13.17 | **45.04** |

previous methods on the Positional reasoning (+5.9 over MCB), Counting (+8.53 over QTA) questions. These improvements are likely due to the pairwise structure induced within the MuRel cell, which makes the answer prediction depend on the spatial and semantic relations between regions. The effectiveness of our per-region context modeling is also demonstrated by the improvement on Scene recognition questions. For these questions, representing the image as a collection of independent objects shows lower performance than replacing each of them in its spatial and semantic context. Interestingly, our results on the harmonic mean of per-type accuracies (H-MPT) are lower than state-of-the-art. For MuRel, this harmonic metric is significantly harmed by our low score of 21.43% on the *Utility and Affordances* task. This is due to the fact that this task concerns the possible usages of objects present in the scene (such as *Can you eat the yellow object?*). They do not require a contextualized visual understanding of the scene.

**State-of-the-art comparison on VQA-CP v2** This dataset has been proposed to evaluate and reduce the question-oriented bias in VQA models. In particular, the distributions of answers with respect to question types differ from *train* to *val* splits. In Table 3.8, we report the scores of two recent baselines (Agrawal et al. 2018; Malinowski et al. 2018), on which we improve significantly. In particular, we demonstrate an important gain over GVQA (Agrawal et al. 2018), whose architecture is designed to focus on Yes/No questions. However, since both methods do not use the Bottom-up features, the fairness of the comparison can be questioned. So we also train an attention model similar to (Ben-Younes* et al. 2017b) using these Bottom-up region representation. We observe that MuRel provides a substantial gain over this strong attention baseline. Given the distribution mismatch between *train* and *val* splits, models that only focus on linguistic biases to answer the question are systematically penalized on their *val* scores. This property of VQA-CP v2 implies that the pairwise iterative structure of MuRel is less prone to question-based overfitting than classical attention architectures.

Table 3.9 – Ablation study of MuRel. Validation of the pairwise module and the iterative processing on the VQA v2 *val* set, VQA-CP v2 and TDIUC datasets.

| Pairwise | Iter. | VQA v2 | VQA CP v2 | TDIUC |
|---|---|---|---|---|
| ✗ | ✗ | 64.13 | 38.88 | 87.50 |
| ✓ | ✗ | 64.57 | 39.12 | 87.86 |
| ✗ | ✓ | 64.72 | 39.37 | 87.92 |
| ✓ | ✓ | **65.14** | **39.54** | **88.20** |

### 3.5.4 Further analysis

**Ablation study** In Table 3.9, we compare three ablated instances of MuRel to its complete form. First, we validate the benefits of the pairwise module. Adding it to a vanilla MuRel without iterative process leads to higher accuracy on every dataset. In fact, between lines 1 and 2, we report a gain of +0.44 on VQA v2, +0.24 on VQA CP v2 and +0.36 on TDIUC. Secondly, we validate the interest of the iterative process. Between line 1 et 3, we report a gain of +0.59 on VQA v2, +0.49 on VQA CP v2 and +0.42 on TDIUC. Notably, this modification does not add any parameters, because we iterate over a single MuRel cell. Unsharring the weights by using a different MuRel cell for each step gives similar results. Finally, the pairwise module and the iterative process are added to create the complete MuRel network. This instance (in line 4) reaches the highest accuracy on the three datasets. Interestingly, the gains provided by the combination of the two methods are sometimes larger than those of each one separately. For instance, we report a gain of +1.01 on VQA v2 between lines 1 and 4. This attests to the complementary of the two modules.

**Number of reasoning steps** In Figure 3.10, we perform an analysis of the iterative process. We train four different MuRel networks on the VQA v2 *train* split, each with a different number of iterations over the MuRel cell. Performance is reported on *val* split. Networks with two and three steps respectively provide a gain of +0.30 and +0.57 in overall accuracy on VQA v2 over the network with a single step. An interesting aspect of the iterative process of MuRel is that the four networks have exactly the same number of parameters, but the accuracy significantly varies with respect to the number of steps. While the accuracy for the answer type involving numbers keeps increasing, we report a decrease in overall accuracy at four reasoning steps. Counting is a challenging task: not only does the model need to detect every occurrence of the desired object, but also the representation computed after the final aggregation must keep the information of the number of detected instances. The complexity of this question may require

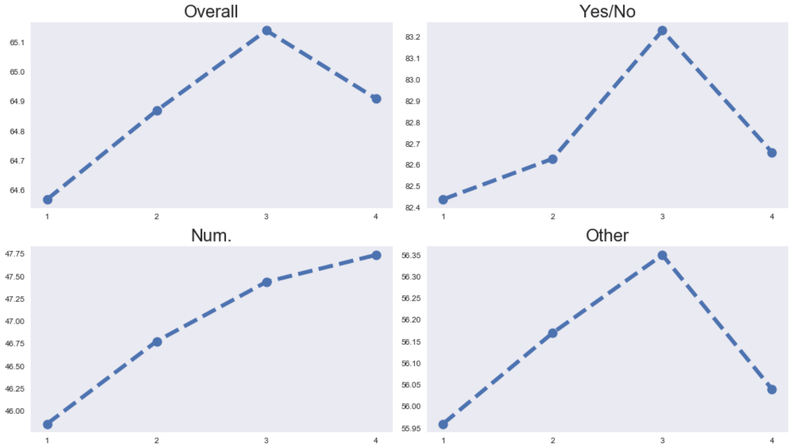


Figure 3.10 – Impact of the number of iterations on the overall accuracy and the accuracy of the three question type of VQA v2 *val*.

deeper relational modeling, and thus benefit from a higher number of iterations over the MuRel cell.

**Entropy of the implicit attention maps** We quantitatively study the behavior of our MuRel network after each reasoning step. To do so, we first compute the implicit attention maps during training as described in 3.8. These maps provide a score going from 0 to 1 for each visual region, and these scores sum to 1 similarly to a probability distribution over regions. Then we calculate the entropy of these distributions. As shown in figure 3.11, the mean entropy converges to different levels depending on the cell. The highest entropy is reached by the first cell, while the lowest entropy is reached by the last cell. This suggests that our model gradually discards visual regions.

**Validation of the architecture** We furthermore validate our MuRel network by introducing architectural modifications inspired by the literature and by evaluating these modified networks on the VQA v2 dataset. First, we replaced our BLOCK fusion inside MuRel cells by an affine transformation inspired by FiLM(Perez et al. 2018), but the latter significantly reduced the accuracy in this setup. Secondly, we investigated different mechanisms to integrate the visual context. Instead of using the coordinates of each region in the pairwise module, we concatenated their coordinates to their associated features similarly to R. Liu et al. 2018. Then, we replaced the pairwise module by a self-attention module incorporating a global pooling over all regions similarly to Jie Hu et al. 2018. These two modifications reduced the accuracy. Finally, we removed the question shortcut used by our multimodal bilinear fusion which outputs the predictions. Instead, we used an unimodal three layers feed-forward network. This modification significantly reduced the accuracy.

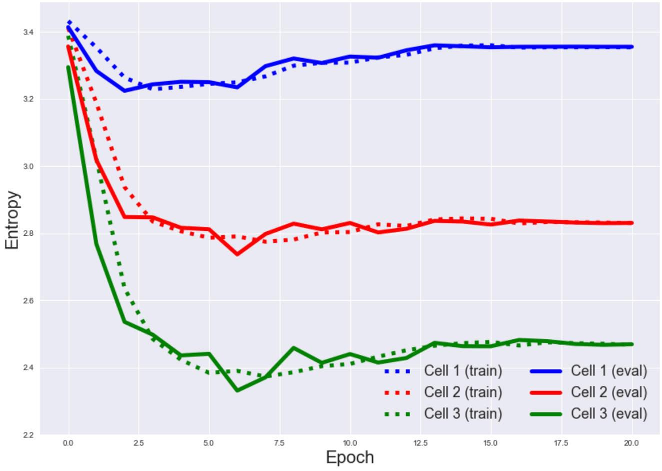


Figure 3.11 – Entropies of the attention maps computed using the outputs of each MuRel cell. The highest entropy is reached by the first cell, while the lowest entropy is reached by the last cell. This suggests that our MuRel network gradually discards visual regions.

### 3.5.5 Qualitative results

**Visualizing MuRel network** Our model can also be leveraged to define visualization schemes finer than mere attention maps. Especially, we can highlight important relations between image regions for answering a specific question. At the end of the MuRel network, the visual features $\{\boldsymbol{s}_i^T\}$ are aggregated using a max operation, yielding a $d_v-$dimensional vector $\boldsymbol{s}$. Thus, we can compute a *contribution map* by measuring to what extent each region contributes to the final vector. To do so, we compute the point-wise $\boldsymbol{c} = \text{argmax}_i\{\boldsymbol{s}_i^T\} \in [1, N]^{d_v}$, and measure the occurrence frequency of each region in this vector $\boldsymbol{c}$. This provides a value for each region that estimates its contribution to the final vector. Interestingly, this process can be done after each cell, and not exclusively at the last one. Intuitively, it measures what the contribution map would have been if the iterative process had stopped at this point. As we can see in Figures 3.7,3.8,3.12, these relevance scores match human intuition and can be used to explain the model's decision, even if the network has not been trained with any selection mechanism.

Similarly, we are able to visualize the pairwise relationships involved in the prediction of the MuRel cell. The first step is to find $i^\star$, which is the region that is the most impacted by the pairwise modeling. It is the region such that $\|\frac{\check{e}_i}{\boldsymbol{x}_i}\|_2$ is maximal (cf. Equation (3.27)). This bounding box is shown in green in all our visualizations. We then measure the contribution of every other region to $i^\star$ using the occurrence frequencies in $\text{argmax}_j\, \boldsymbol{r}_{i,j}$. We show in red the regions whose contribution to $i^\star$ is above a certain threshold (0.2 in our visualizations). If there is no such region, the green box is not shown.

**Qualitative results** In Figure 3.12 we illustrate the behavior of a MuRel network with three shared cells. Iterations through the MuRel cell tend to gradually discard regions, keeping only the most relevant ones. As explained in Section 3.4.1, the regions that are most involved in the pairwise modeling process are shown in green and red. Both region contributions and pairwise links match human intuition. In the first row, the most relevant relations according to our model are between the player's hand, containing the WII controller, and the screen, which explains the prediction *bowling*. In the third row, the model answers *kite* using the relation between the man's hand and the kite he is holding. Finally, in the last row, our model is able to address a third question on the same image than in Figure 4.1 and 3.8. Here, the relation between the head of the woman and her hat is used to provide the right answer. As VQA models are often subject to linguistic bias (Goyal et al. 2017; Agrawal et al. 2018), this type of visualization shows that the MuRel network actually relies on the visual information to answer questions.

### 3.5.6 Implementation details

**Software, hardware and pretrained models** We use pytorch 1.1.0 to implement our algorithms in order to benefit from the GPU acceleration. We use four NVidia Titan Xp GPU in this study. We use a single GPU for each experiment. We use a dedicated Solid State Drive to load the visual features using multiple threads. Our code and pretrained models can be found on github:

- `github.com/Cadene/block.bootstrap.pytorch`
- `github.com/Cadene/murel.bootstrap.pytorch`

**Image encoder** We use the pretrained Faster-RCNN (S. Ren et al. 2015) by Anderson et al. 2018 on the Visual Genome dataset (Krishna et al. 2017) to extract object features from each image. We use the setup that extracts 36 regions for each image. We do not fine-tune the image extractor.

**Question encoder** We use the same preprocessing as Fukui et al. 2016, which apply a lower case transformation and remove all the punctuation. We only consider the questions that are associated with the 3000 most occurring answers (1480 for the TDIUC dataset) while containing less than 26 words. We use a pretrained skip-thought GRU encoder by Kiros et al. 2015. For MuRel experiments only, we use the same two-glimpses self-attention mechanism proposed by Z. Yu et al. 2017 to represent our question in a 4800-dimensional space. We fine-tune every parameters of the pretrained skip-thought including the embedding layer.

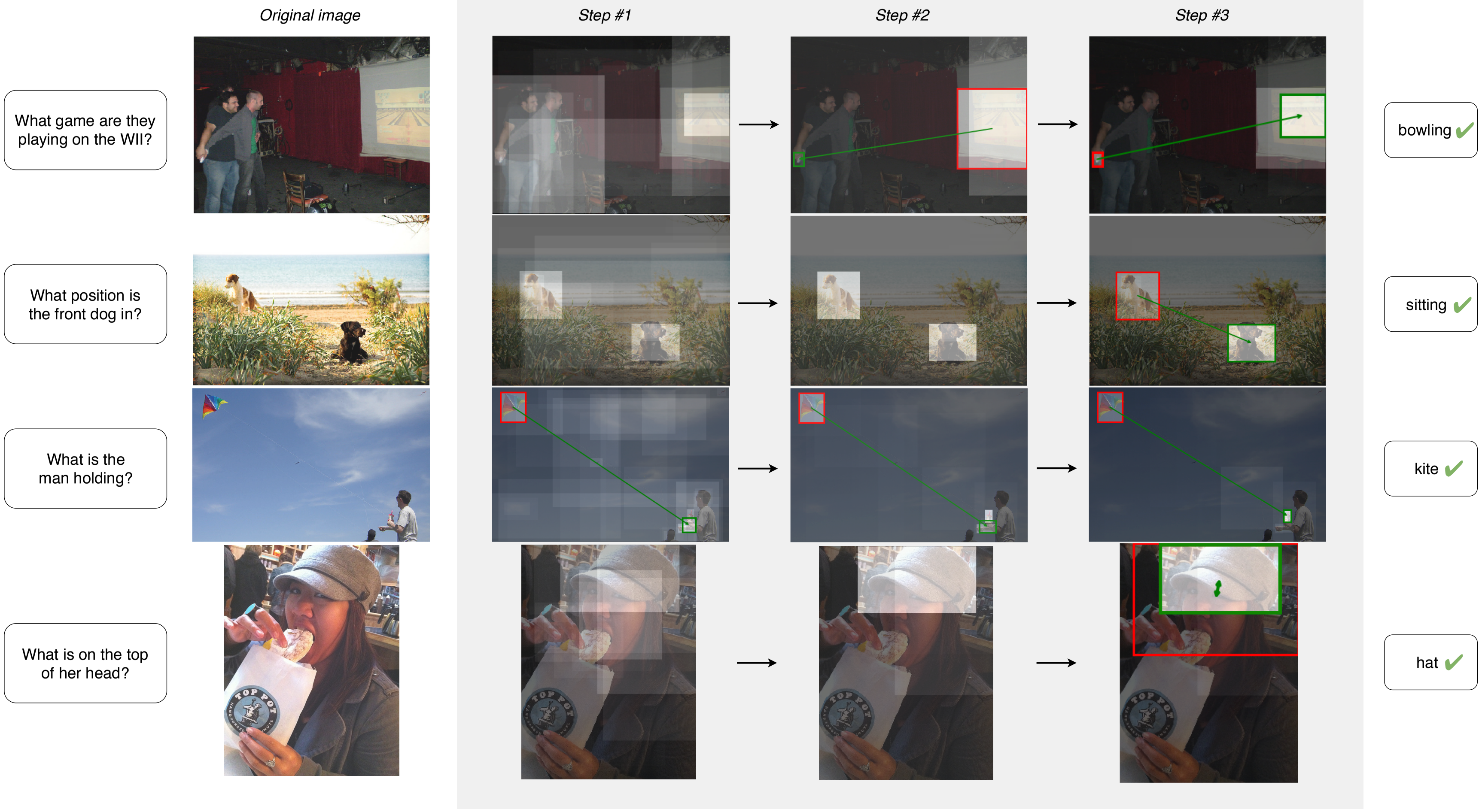

Original image
Step #1
Step #2
Step #3
What game are they playing on the WII?
What position is the front dog in?
What is the man holding?
What is on the top of her head?
bowling ✓
sitting ✓
kite ✓
hat ✓

Figure 3.12 – Qualitative results of MuRel. Visualization of the importance maps with colored regions related to the relational mechanism. As in Figure 3.8, the most selected regions by the implicit attentional mechanism are shown in brighter. The green region is the most impacted by the pairwise modeling, while the red regions impact the green regions the most. These colored regions are only represented if they are greater than a certain threshold.

**Multimodal architecture** Our MuRel network is composed of three MuRel cells sharing their parameters. The BLOCK fusions inside a MuRel cell and inside the classification module use a rank of 15, a dimension of 1600 and 20 chunks. The two BLOCK fusions inside the Pairwise module (coordinates and visual features) use a rank of 10, a dimension of 500 and 10 chunks.

**Optimization process** We use Adam as optimizer (Kingma et al. 2014) with a learning rate of $5 * 10^{-5}$ and a batch size of 256. During the first 7 epochs, we linearly increase the learning rate to $2 * 10^{-4}$. After the epoch 14, we decrease it by a factor 0.25 every two epochs until convergence. We also apply a gradient clipping of 0.25. We use early stopping based on the validation accuracy. This process is inspired from Yu Jiang* et al. 2018 and J.-H. Kim et al. 2018.

## 3.6 Conclusion

We addressed the problem of fusing the visual and textual modalities for the VQA task which consists in learning to answer a question about an image. We evaluated our contributions against the state-of-the-art on several widely used datasets such as VQA v2, VQA-CP v2 and TDIUC. First, we proposed two multimodal fusion modules, MUTAN and BLOCK, which are based on different factorization of bilinear models. MUTAN is based on the Tucker decomposition. BLOCK is based on the Block-term decomposition and generalizes MUTAN. Both leverage a sparsity constraint. We found our fusions to be competitive with the best fusions from the literature while allowing for a much lower number of free parameters.

Secondly, we proposed the MuRel network which is a multimodal architecture that leverages our fusion modules. MuRel iteratively merges object-based visual representations with the question representation while sharing its parameters between each of its MuRel cells. Each MuRel cell also includes a pairwise modeling between visual regions which adds relational and contextual information in the multimodal representations. Instead of relying on the widely used question-driven attention to remove the spatial dimensions, MuRel uses a simple averaging between each visual region representations. We found that MuRel significantly surpasses the challenging question-driven attention baseline on the three tested datasets with constant gain.Finally, we provided further analysis of MuRel to better understand its reasoning process. We notably found with a mean entropy calculation that MuRel iteratively refines its focus on object regions.

Chapter 4

# REDUCING UNIMODAL BIASES FOR VISUAL QUESTION ANSWERING

## Contents

**Chapter abstract**

*An important problem of current Machine Learning (ML) models is that they tend to learn unwanted biases by giving too much importance to some predictive features. In multimodal learning, a model can be easily biased towards a certain modality due to the heterogeneous nature of the data. In particular, it has been found that Visual Question Answering (VQA) models often exploit unimodal biases to provide the correct answer without using the visual information. As a result, they suffer from a huge drop in performance when evaluated on data outside their training set distribution. This critical issue makes them unsuitable for real-world settings.*

*We propose RUBi, a new learning strategy to reduce biases in any VQA model. It reduces the importance of the most biased examples, i.e. examples that can be correctly classified without looking at the image. It implicitly forces the VQA model to use the two input modalities instead of relying on statistical regularities between the question and the answer. We leverage a question-only model that captures the language biases by identifying when a given question can be answered without looking at the image. It prevents the base*

*VQA model from learning them by influencing its predictions. This leads to dynamically adjusting the loss in order to compensate for biases. We validate our RUBi learning strategy with three different architectures on VQA-CP v1 and VQA-CP v2. These datasets are designed to penalize VQA models that are biased towards the question modality. We also show that models trained with RUBi obtain competitive results on the original VQA2 v2 dataset. Finally, we provide an experimental analysis of the grounding ability of our models.*

*The work in this chapter, at equal contribution with Corentin Dancette, has led to the publication of a conference paper:*

- Remi Cadene*, Corentin Dancette*, Hedi Ben-Younes, Matthieu Cord, and Devi Parikh (2019). "RUBi: Reducing Unimodal Biases for Visual Question Answering". In: *Advances in Neural Information Processing Systems (NeurIPS)*. URL: https://arxiv.org/abs/1906.10169.

## 4.1 Introduction

In Chapter 3, we introduced fusion modules and reasoning architectures for the Visual Question Answering (VQA) task. Now, we focus on the learning aspect in the context of the current VQA datasets. In fact, an important issue of Machine Learning (ML) models is that they tend to learn unwanted dataset biases. By giving too much importance to some predictive features, they fail to learn the correct behaviors that allow them to generalize outside their training set distribution. In multimodal learning, a model can be easily biased towards a certain modality due to the heterogeneous nature of the data. Especially, it has been found that some of the best VQA models heavily exploit unimodal biases (Agrawal et al. 2016; Agrawal et al. 2018; Ramakrishnan et al. 2018; Johnson et al. 2017a; Hudson et al. 2019). Instead of providing an answer based on the question and the image, they often use the question only. As illustrated in Figure 4.1, a biased VQA model towards the question modality would only read the question to answer *yellow* because most of the bananas are yellow in the training set. Instead of learning the correct behavior which consists in locating the banana in the image and finding the term describing its color, models tend to rely on the statistical shortcut linking the words *what*, *color* and *bananas* with the most occurring answer *yellow*.

An efficient way to quantify the potential amount of statistical shortcuts that can be leveraged for each modality is to train unimodal models. For instance, a question-only model trained on the widely used VQA v2 dataset (Goyal et al. 2017) predicts the correct answer approximately 44% of the time over the testing set. VQA models are not discouraged to exploit these statistical shortcuts from the question modality, because their training set often follows a very similar distribution as their testing set. However, when evaluated on a testing set that displays different statistical regularities, they usually suffer from a significant

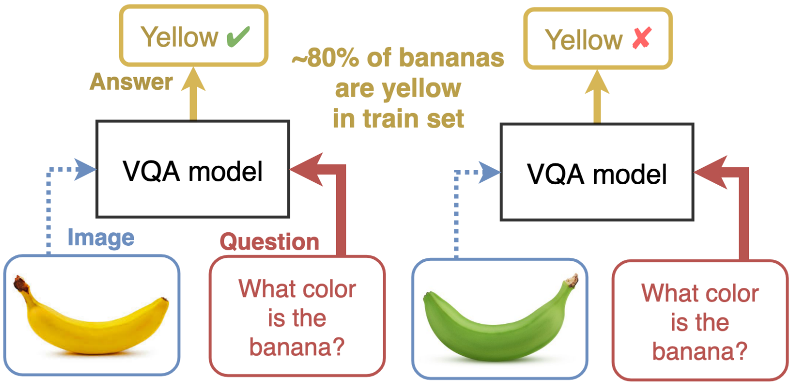

Yellow ✔
Answer
~80% of bananas are yellow in train set
Yellow ✘
VQA model
VQA model
Image
Question
What color is the banana?
What color is the banana?

Figure 4.1 – Current VQA models often rely on unwanted statistical correlations between the question and the answer instead of using both modalities. A biased model will answer *yellow* without looking at the image, because it learned that the presence of the words *what*, *color* and *banana* is highly predictive of the answer *yellow*.

drop in accuracy (Agrawal et al. 2018). Unfortunately, these statistical regularities are hard to avoid when collecting real datasets. Also, the process of carefully balancing the dataset statistics can be made impossible by a lack of precise annotations over the image (Hudson et al. 2019). Thus, there is a crucial need to develop new strategies to reduce the amount of unwanted biases in order to learn better behaviors.

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

In Section 4.2, we review recent approaches to detect and reduce unwanted biases, in particular in the context of VQA. Then, we introduce our RUBi approach in Section 4.3. Finally, we evaluate our training strategy in Section 4.4. We report results of several VQA architectures (Yang et al. 2016; Anderson et al. 2018) trained on the VQA-CP v1 and VQA-CP v2 (Agrawal et al. 2018) datasets which penalize models that are biased towards the question modality, as well as on the VQA v2 dataset (Goyal et al. 2017). Also, we evaluate the behaviors induced by our learning strategy using the grounding ability as a proxy on the VQA-HAT dataset (Das et al. 2016).

## 4.2 Related work

Real-world datasets display some form of inherent biases due to their collection process (Gordon et al. 2013; Chao et al. 2018; Torralba et al. 2011). As a result, machine learning models tend to reflect these biases because they often capture undesirable correlations between the inputs and the ground truth annotations (Stock et al. 2018; Jia et al. 2018; Manjunatha et al. 2019). Procedures exist to identify certain kinds of biases and to reduce them. For instance, some methods are focused on gender biases (Hendricks et al. 2018; Zhao et al. 2017), some others on the human reporting biases (Misra et al. 2016), and also on the shift in distribution between lab-curated data and real-world data (Gupta et al. 2018). In the language and vision context, some works evaluate unimodal baselines (Anand et al. 2018; Thomason et al. 2019) or leverage language priors (Rohrbach et al. 2018). In the following, we discuss related work that assess and reduce unimodal biases learned by VQA models.

**Assessing unimodal biases in datasets and models** Despite being designed to merge the two input modalities, it has been found that VQA models often rely on superficial correlations between inputs from one modality and the answers without considering the other modality (Jabri et al. 2016; Manjunatha et al. 2019). An interesting way to quantify the amount of unimodal biases that can potentially be learned by a VQA model consists in training models using only one of the two modalities (Antol et al. 2015; Goyal et al. 2017). The question-only model is a particularly strong baseline because of the large amount of statistical regularities that can be leveraged from the question modality. Unfortunately, biased models that exploit statistical shortcuts from one modality usually reach impressive accuracy on most of the current benchmarks. VQA-CP v2 and VQA-CP v1 (Agrawal et al. 2018) were recently introduced as diagnostic datasets containing different answer distributions for each question-type between train and test splits. Consequentially, models biased towards the question modality suffer from a huge drop in accuracy on these benchmarks.

**Balancing datasets to avoid unimodal biases** Once the unimodal biases have been identified, one method to overcome these biases is to create more balanced datasets. For instance, the synthetic datasets for VQA (Johnson et al. 2017a; Hudson et al. 2019) minimize question-conditional biases via rejection sampling within families of related questions to avoid simple shortcuts to the correct answer. Doing rejection sampling in real VQA datasets is usually not possible due to the cost of annotations. Another solution is to collect complementary examples to increase the difficulty of the task. For instance, VQA v2 (Goyal et al. 2017) has been introduced to weaken language priors in the VQA v1 dataset (Antol et al. 2015) by identifying complementary images. For a given VQA v1 question, VQA v2 also contains a similar image with a different answer to the same question. However, even with this additional balancing, statistical biases from the question remain and can be leveraged (Agrawal et al. 2018).

**Architectures and learning strategies to reduce unimodal biases** In parallel to these previous works on balancing datasets, an important effort has been carried out to design VQA models to overcome biases from datasets. Agrawal et al. 2018 proposed a hand-designed architecture called Grounded VQA model (GVQA). It breaks the task of VQA down into a first step of locating and recognizing the visual regions needed to answer the question, and a second step of identifying the space of plausible answers based on a question-only branch. This approach requires training multiple sub-models separately. The most recent approach proposed by Ramakrishnan et al. 2018 introduces a learning strategy to overcome language priors in VQA models. An adversary question-only branch takes as input the question encoding from the VQA model and produces a question-only loss. They use a gradient negation of this loss to discourage the question encoder to capture unwanted biases that could be exploited by the VQA model. They also propose a loss based on the difference of entropies between the VQA model and the question-only branch output distributions. These two losses are only backpropagated to the question encoder.

In contrast to GVQA Agrawal et al. 2018, our learning strategy is end-to-end. Their complex design is not straightforward to apply on different architectures while our approach is model-agnostic. In contrast to the recent approach by Ramakrishnan et al. 2018, our proposed learning strategy targets the full VQA model parameters to reduce the impact of unwanted biases more effectively. It also takes advantage of the question-only model to prevent VQA models from learning question biases. However, instead of relying on these two additional losses, we use the question-only branch to dynamically adapt the value of the classification loss. By doing so, we reduce the importance of certain examples, similarly to the rejection sampling approach, while increasing the importance of complementary examples, which are already in the training set. A visual comparison between the training strategy proposed by Ramakrishnan et al. 2018 and RUBi can be found in Figure 4.5.

## 4.3 Reducing Unimodal Biases approach

In this section, we present our RUBi approach to reduce biases in VQA. We follow the same formalism introduced in Chapter 3 while introducing specific notations to describe different parts of the VQA architectures on which RUBi is applied. We recall that the VQA is tackled as a multi-class classification problem. Given a dataset $\mathcal{D}$ consisting of $n$ triplets $(v_i, q_i, a_i)_{i\in[1,n]}$ with $v_i \in \mathcal{V}$ an image, $q_i \in \mathcal{Q}$ a question in natural language and $a_i \in \mathcal{A}$ an answer, one must optimize the parameters $\theta$ of the function $f : \mathcal{V} \times \mathcal{Q} \rightarrow \mathbb{R}^{|\mathcal{A}|}$ to produce accurate predictions. For a single example, VQA models use an image encoder $e_v : \mathcal{V} \rightarrow \mathbb{R}^{n_v \times d_v}$ to output a set of $n_v$ vectors of dimension $d_v$, a question encoder $e_q : \mathcal{Q} \rightarrow \mathbb{R}^{n_q \times d_q}$ to output a set of $n_q$ vectors of dimension $d_q$, a multimodal fusion $m : \mathbb{R}^{n_v \times d_v} \times \mathbb{R}^{n_q \times d_q} \rightarrow \mathbb{R}^{d_m}$, and a classifier $c : \mathbb{R}^{d_m} \rightarrow \mathbb{R}^{|\mathcal{A}|}$. These functions are composed as follows:

$$f(v_i, q_i) = c(m(e_v(v_i), e_q(q_i))) \tag{4.1}$$

Each one of them can be defined to instantiate most of the state-of-the-art models, such as Yang et al. 2016; Lu et al. 2016b; J.-H. Kim et al. 2018; Ben-Younes* et al. 2017b; Ben-Younes et al. 2019; Z. Yu et al. 2018; Cadène* et al. 2019 to cite a few.

**Classical learning strategy and pitfall** The classical learning strategy of VQA models, depicted in Figure 4.2, consists in minimizing the standard cross-entropy criterion over a dataset of size $n$.

$$\mathcal{L}(\theta; \mathcal{D}) = -\frac{1}{n}\sum_{i=1}^{n} \log(\text{softmax}(f(v_i, q_i)))[a_i] \tag{4.2}$$

VQA models are inclined to learn unimodal biases from the datasets (Agrawal et al. 2018). This can be shown by evaluating models on datasets that have different distributions of answers for the test set, such as VQA-CP v2. In other words, they overrely on statistical regularities from one modality to provide accurate predictions. As an extreme example, strongly biased models towards the question modality always output *yellow* to the question *what color is the banana*. They do not learn to use the image information because there are too few examples in the dataset where the banana is not yellow. Once trained, their inability to use the two modalities adequately makes them inoperable on data coming from different distributions such as real-world data. Our contribution consists in modifying this cost function to avoid the learning of these biases.

### 4.3.1 RUBi learning strategy

**Capturing biases with a question-only branch** One way to measure the unimodal biases in VQA datasets is to train a unimodal model which takes only

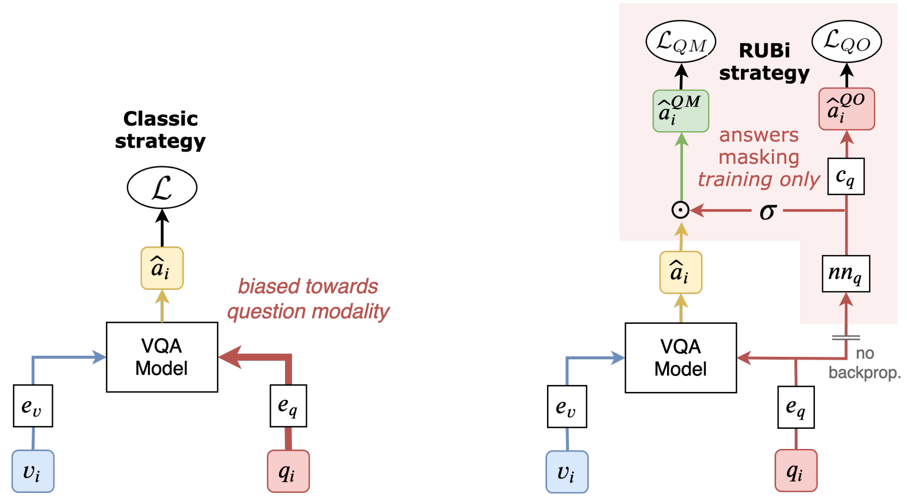


Figure 4.2 – Visual comparison between the classical learning strategy of a VQA model and our RUBi learning strategy. The red highlighted modules are removed at the end of the training. The output $\hat{a}_i$ is used as the final prediction.

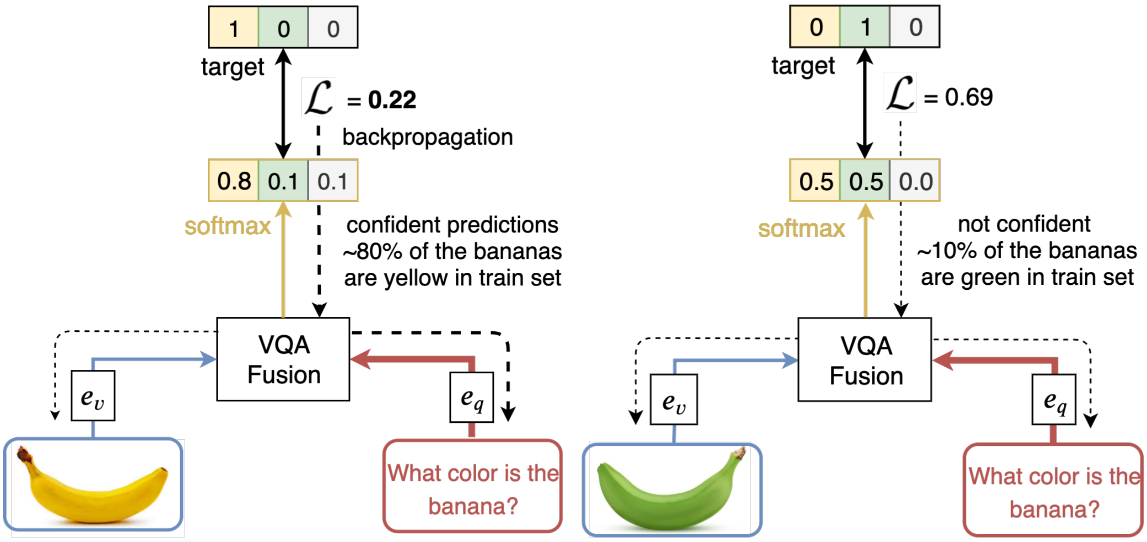
one of the two modalities as input. The key idea of our approach, depicted in Figure 4.2, is to adapt a question-only model as a branch of our VQA model, that will alter the main model's predictions. By doing so, the question-only branch captures the question biases, allowing the VQA model to focus on the examples that cannot be answered correctly using the question modality only. The question-only branch can be formalized as a function $f_Q : \mathcal{Q} \to \mathbb{R}^{|\mathcal{A}|}$ parameterized by $\theta_Q$, and composed of a question encoder $e_q : \mathcal{Q} \to \mathbb{R}^{n_q \times d_q}$ to output a set of $n_q$ vectors of dimension $d_q$, a neural network $nn_q$: $\mathbb{R}^{n_q \times d_q} \to \mathbb{R}^{|\mathcal{A}|}$ and a classifier $c_q$: $\mathbb{R}^{|\mathcal{A}|} \to \mathbb{R}^{|\mathcal{A}|}$.

$$f_Q(q_i) = c_q(nn_q(e_q(q_i))) \tag{4.3}$$

During training, the branch acts as a proxy preventing any VQA model of the form presented in Equation (4.1) from learning biases. At the end of the training, we simply remove the branch and use the predictions from the base VQA model.

**Preventing biases by masking predictions** Before passing the predictions of our base VQA model to the loss function defined in Equation (4.2), we merge them with a mask of length $|\mathcal{A}|$ containing a scalar value between 0 and 1 for each answer. This mask is obtained by passing the output of the neural network $nn_q$ through a sigmoid function $\sigma$. The goal of this mask is to dynamically alter the loss by modifying the predictions of the VQA model. To obtain the new predictions, we simply compute an element-wise product $\odot$ between the mask and the original predictions as defined in the following equation.

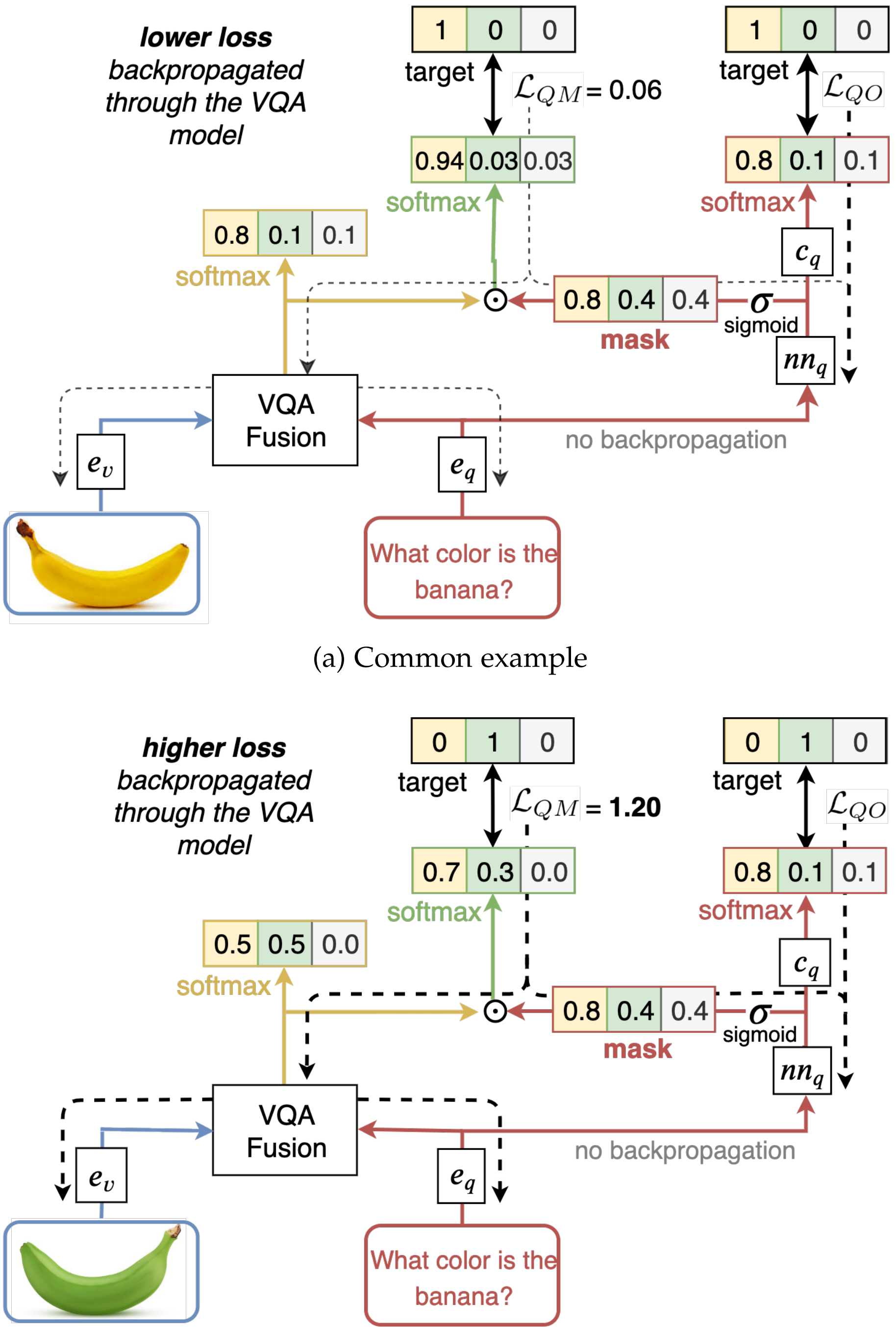
$$f_{QM}(v_i, q_i) = f(v_i, q_i) \odot \sigma(nn_q(e_q(q_i)))) \tag{4.4}$$



(a) Common example (b) Rare example

Figure 4.3 – Illustration of the classic learning strategy which consists in backpropagating the loss $\mathcal{L}$ through the VQA model and the unimodal encoders. The trained model fails at learning the correct behaviors of using both modalities, because it over relies on the question modality.

Our approach modifies the predictions in this specific way to prevent the VQA model to learn biases from the question. It is to be compared to the classic learning strategy illustrated in Figure 4.3. To better understand the impact of our approach on the learning, we examine two scenarios. First, we reduce the importance of the most biased examples, i.e. examples that can be correctly classified without using the image modality. To do so, the question-only branch outputs a mask to increase the score of the correct answer while decreasing the scores of the others. As a result, the loss is much lower for these biased examples. In other words, the gradients backpropagated through the VQA model are smaller, thereby reducing the importance of these examples in the learning. As illustrated in the first row of Figure 4.4, given the question *what color is the banana*, the mask takes a high value of 0.8 for the answer *yellow* which is the most likely answer for this question in the training set. On the other hand, the value for the other answers *green* and *white* are smaller. We see that the mask influences the VQA model to produce new predictions where the score associated with the answer *yellow* increases from 0.8 to 0.94. Compared to the classical learning approach, the loss is smaller with RUBi and decreases from 0.22 to 0.06. Secondly, we increase the importance of examples that cannot be answered without using both modalities. For these examples, the question-only branch outputs a mask that increases the score of the wrong answer. As a result, the loss is much higher and the VQA model is encouraged to learn from these examples. We illustrate this behavior in the second row of Figure 4.4



(a) Common example

(b) Rare example

Figure 4.4 – Detailed illustration of the RUBi impact on the learning. In the first row, we illustrate how RUBi reduces the loss for common examples that can be correctly answered without looking at the image. In the second row, we illustrate how RUBi increases the loss for rare examples that cannot be answered without using both modalities.

for the same question about the color of the banana. When the image contains a green banana, RUBi increases the loss from 0.69 to 1.20.

**Joint learning procedure** As explained in Section 1.2, our learning framework consists in minimizing a loss function. The loss computed for each example is backpropagated to obtain gradients with respect to each parameter of the VQA model. Then, these parameters are optimized using the Stochastic Gradient Descent (SGD) algorithm to minimize the loss. The cross-entropy loss is usually used to tackle multiclass classification problems. However, we introduce a new loss to reduce biases learned by VQA models. The main loss $\mathcal{L}_{QM}$ refers to the cross-entropy loss associated with the predictions of $f_{QM}(v_i, q_i)$ from Equation 4.4. We backpropagate this loss to optimize all the parameters $\theta_{QM}$ which contributed to this loss. $\theta_{QM}$ is the union of the parameters of the base VQA model, the encoders, and the neural network $nn_q$ of the question-only branch. In our setup, we share the parameters of the question encoder $e_q$ between the VQA model and the question-only branch. The question-only loss $\mathcal{L}_{QO}$ is a cross-entropy loss associated with the predictions of $f_Q(q_i)$ from Equation 4.3. We use this loss to only optimize $\theta_{QO}$, which is the union of the parameters of $c_q$ and $nn_q$. By doing so, we further improve the question-only branch ability to capture biases. Note that we do not backpropagate this loss to the question encoder $e_q$ preventing it from directly learning question biases. We obtain our final loss $\mathcal{L}_{\text{RUBi}}$ by summing the two losses together in the following equation:

$$\mathcal{L}_{\text{RUBi}}(\theta_{QM}, \theta_{QO}; \mathcal{D}) = \mathcal{L}_{QM}(\theta_{QM}; \mathcal{D}) + \mathcal{L}_{QO}(\theta_{QO}; \mathcal{D}) \tag{4.5}$$

Finally, we jointly optimize the parameters of the base VQA model and its question-only branch using the gradients computed from the two losses.

### 4.3.2 Baseline architecture

Most state-of-the-art VQA architectures are compatible with our RUBi learning strategy. To test our strategy, we design a fast and simple architecture inspired from the MuRel network (Cadène* et al. 2019) defined in Chapter 3. This baseline architecture is detailed at the end of the experiments section. As common in the state of the art, our baseline architecture encodes the image as a bag of $n_v$ visual features $\mathbf{v}_i \in \mathbb{R}^{d_v}$ using the pretrained Faster R-CNN by Anderson et al. 2018, and encodes the question as a vector $\mathbf{q} \in \mathbb{R}^{d_q}$ using a GRU, pretrained on the skipthought task Kiros et al. 2015. The VQA model consists of a bilinear BLOCK fusion (Ben-Younes et al. 2019) defined in Section 3.3.2. It merges the question representation $\mathbf{q}$ with the features $\mathbf{v}_i$ of each region of the image. The output is aggregated using a max-pooling on the $n_v$ regions. The resulting vector is then fed into a MLP classifier which outputs the final predictions. While most of our experiments are done with this fast and simple baseline architecture, we

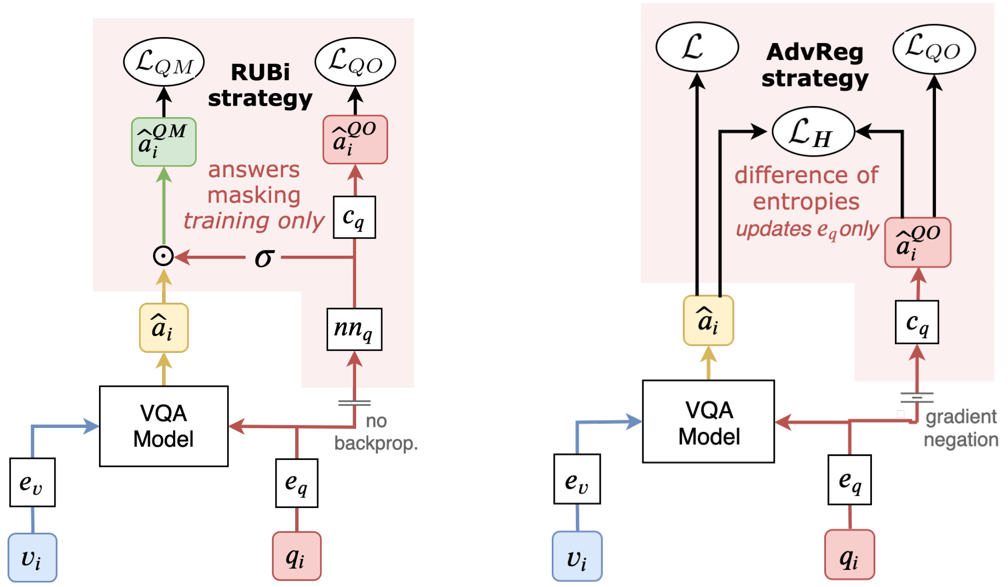


Figure 4.5 – Visual comparison between RUBi and the training strategy proposed by Ramakrishnan et al. 2018.

experimentally demonstrate that the RUBi learning strategy is effective on other VQA architectures.

## 4.4 Experiments

**Experimental setup** We train and evaluate our models on VQA-CP v2 and VQA-CP v1(Agrawal et al. 2018). These datasets were developed to evaluate the models' robustness to question biases. We follow the same training and evaluation protocol as Ramakrishnan et al. 2018, who also propose a learning strategy to reduce biases. A visual comparison between RUBi and the approach of Ramakrishnan et al. 2018 is illustrated in Figure 4.5. For each model, we report the standard VQA evaluation metric (Antol et al. 2015). We also evaluate our models on the standard VQA v2 dataset (Goyal et al. 2017). Further implementation details are included at the end of this section.

### 4.4.1 Validation of RUBi

**State-of-the-art comparison on VQA-CP v2** We compare our approach consisting of our baseline architecture trained with RUBi (Baseline + RUBi) on VQA-CP v2 against state-of-the-art approaches[1]. In Table 4.1, we only report approaches that use the richer object-based visual features proposed by Anderson

[1] at the time of submission

Table 4.1 – Performance on VQA-CP v2 `test`. All reported models use the same object-level features proposed by Anderson et al. 2018. (*) are reported from Ramakrishnan et al. 2018. (**) are reported from Shrestha et al. 2019.

| Model | Overall | Answer type | | |
|---|---|---|---|---|
| | | Yes/No | Number | Other |
| Question-Only (Agrawal et al. 2018) | 15.95 | 35.09 | 11.63 | 7.11 |
| UpDn** (Anderson et al. 2018) | 38.01 | . | . | . |
| RAMEN (Shrestha et al. 2019) | 39.21 | . | . | . |
| BAN** (J.-H. Kim et al. 2018) | 39.31 | . | . | . |
| MuRel (Cadène* et al. 2019) | 39.54 | 42.85 | 13.17 | 45.04 |
| UpDn* (Anderson et al. 2018) | 39.74 | 42.27 | 11.93 | **46.05** |
| UpDn + AdvReg (Ramakrishnan et al. 2018) | 41.17 | 65.49 | 15.48 | 35.48 |
| Balanced Sampling | 40.38 | 57.99 | 10.07 | 39.23 |
| Q-type Balanced Sampling | 42.11 | 61.55 | 11.26 | 40.39 |
| Baseline (ours) | 38.46 | 42.85 | 12.81 | 43.20 |
| Baseline + RUBi (ours) | **47.11** | **68.65** | **20.28** | 43.18 |
| SAN (Yang et al. 2016) | 24.96 | 38.35 | 11.14 | 21.74 |
| SAN + RUBi | **37.63** | **59.49** | **13.71** | **32.74** |
| UpDn (Anderson et al. 2018) | 39.74 | 42.27 | 11.93 | **46.05** |
| UpDn + RUBi | **44.23** | **67.05** | **17.48** | 39.61 |

et al. 2018. We report the average accuracy over 5 experiments with different random seeds. Our approach reaches an average overall accuracy of 47.11% with a low standard deviation of ±0.51. This accuracy corresponds to a gain of +5.94 percentage points over the current state-of-the-art UpDn + AdvReg. AdvReg corresponds to the Q-Adv + DoE strategy reported from Ramakrishnan et al. 2018. RUBi leads to a +8.65 improvement over our baseline model trained with the classical cross-entropy. In comparison, the second-best approach UpDn + AdvReg only achieves a +1.43 gain in overall accuracy over their baseline UpDn. In addition, our approach does not significantly reduce the accuracy over our baseline for the answer type *Other*, while the second-best approach reduces it by 10.57 point. Contrarily to other reported methods, GVQA (Agrawal et al. 2018) does not use the rich object-based visual features proposed by Anderson et al. 2018. However, it has been specifically designed for VQA-CP. Baseline + RUBi leads to a significant gain of +15.88 over GVQA (Agrawal et al. 2018).

Table 4.2 – Performance on VQA-CP v1. We report the overall accuracy top1 results and the one for each answer type (yes/no, number and other).

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

**Additionnal baseline on VQA-CP v2** We also compare our baseline architecture trained with RUBi on VQA-CP v2 against two sampling-based training strategies inspired by standard methods to handle imbalanced datasets. In the *Balanced Sampling* method, we sample the questions such that the answer distribution is uniform. In the *Question-Type Balanced Sampling* method, we sample the questions such that for every question type, the answer distribution is uniform, but the question type distribution remains the same overall. Both methods are tested with our baseline architecture. In Table 4.1, we report that the *Question-Type Balanced Sampling* improves the overall accuracy from 38.46 to 42.11. This gain is already +0.94 higher than the previous state-of-the-art method (Ramakrishnan et al. 2018), but remains significantly lower than our proposed method.

**State-of-the-art comparison on VQA-CP v1** We compare our baseline architecture trained with RUBi (Baseline + RUBi) on the VQA-CP v1 dataset (Agrawal et al. 2018) against state-of-the-art approaches[2]. In Table 4.2, our approach reaches the overall accuracy of 46.93 which corresponds to a significant gain of +3.5 against the best scoring approach SAN + AdvReg Ramakrishnan et al. 2018. We also compare the two training strategies on the same SAN architecture. We report a gain of +2.65 using RUBi against AdvReg. Finally, we report a gain of +7.7 against when comparing our best scoring approach against GVQA (Agrawal et al. 2018). The latter has been specifically designed for VQA-CP datasets.

**Architecture agnostic abilities on VQA-CP v2 and VQA-CP v1** We compare the impact of RUBi training on three different architectures to evaluate its ability

[2] at the time of submission

Table 4.3 – Comparison with or without RUBi learning strategy on VQA v2 `val` and `test-dev` splits. We report the overall accuracy top1 results.

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

to be architecture agnostic. In Table 4.1 and in Table 4.2, we report results on VQA-CP v2 and VQA-CP v1 respectively. RUBi leads to overall accuracy gains of +8.65 and +9.8 respectively on our baseline architecture, +12.67 and +19.2 on SAN, and +4.49 and +7.66 for UpDn.

### 4.4.2 Further analysis

**Impact on VQA v2** We report the impact of our method on the standard VQA v2 dataset in Table 4.3. VQA v2 train, val and test sets follow the same distribution, contrarily to VQA-CP v2 train and test sets. In this context, we usually observe a drop in accuracy using approaches focused on reducing biases. This is due to the fact that exploiting unwanted correlations from the VQA v2 train set is not discouraged and often leads to a higher accuracy on the test set. Nevertheless, our RUBi approach leads to a comparable drop to what can be seen in the state-of-the-art. We report a drop of 1.94 percentage points with respect to our baseline, while Agrawal et al. 2018 report a drop of 3.78 between GVQA and their SAN baseline. Ramakrishnan et al. 2018 report drops of 0.05, 0.73 and 2.95 for their three learning strategies with the UpDn architecture which uses the same visual features as RUBi. As shown in this section, RUBi improves the accuracy on VQA-CP v2 from a large margin, while maintaining competitive performance on the standard VQA v2 dataset compared to similar approaches.

**Validation of the masking strategy** We compare different fusion techniques to combine the output of $nn_q$ with the output from the VQA model. We report a drop of 7.09 accuracy points on VQA-CP v2 by replacing the sigmoid with a ReLU on our best scoring model. Using an element-wise sum instead of an element-wise product leads to a further performance drop. These results confirm the effectiveness of our proposed masking method which relies on a sigmoid and an element-wise sum.

**Validation of the question-only loss** We validate the ability of the question-only loss $\mathcal{L}_{QO}$ to reduce the question biases. The absence of $\mathcal{L}_{QO}$ implies that the question-only classifier $c_q$ is never used, and $nn_q$ only receives gradients from the main loss $\mathcal{L}_{QM}$. Using $\mathcal{L}_{QO}$ leads to consistent gains on all three architectures. In

Table 4.4 – Ablation study of the question-only loss $\mathcal{L}_{QO}$ on VQA-CP v2. We report the overall accuracy top1 results and the one for each answer type (yes/no, number and other).

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

Table 4.4, we report gains of +0.89 for our baseline architecture, +0.22 for SAN, +4.76 for UpDn.

**Grounding ability on VQA-HAT** We conduct additional studies to evaluate the grounding ability of models trained with RUBi. We follow the experimental protocol of VQA-HAT (Das et al. 2016). We train our models on VQA v1 train set and evaluate them using rank-correlation on the VQA-HAT val set, which is a subset of the VQA v1 val set. This metric compares attention maps computed from a model against human annotations indicating which regions humans found relevant for answering the question. In Table 4.5, we report a gain of +0.012 with our baseline architecture trained with RUBi, a gain of +0.019 with SAN and a loss of -0.003 with UpDn architecture. This preliminary work need to be extended to assess the real impact on grounding induced by RUBi.

### 4.4.3 Qualitative analysis

**Visualization of the impact of RUBi on VQA-CP v2** To better understand the impact of our RUBi approach, we compare in Figure 4.6 the answer distribution on VQA-CP v2 for some specific question patterns. We also display interesting behaviors on some examples using attention maps extracted using the method proposed in Cadène* et al. 2019 and defined at the end of Section 3.4.1.

In the first row, we show the ability of RUBi to reduce biases for the *is this person skiing* question pattern. Most examples in the train set have the answer *yes*, while in the test set, they have the answer *no*. Nevertheless, RUBi outputs 80% of *no*, while the baseline almost always outputs *yes*. Interestingly, the best scoring region from the attention map of both models is localized on the shoes. To get the answer right, RUBi seems to reason about the absence of skis in this region. It

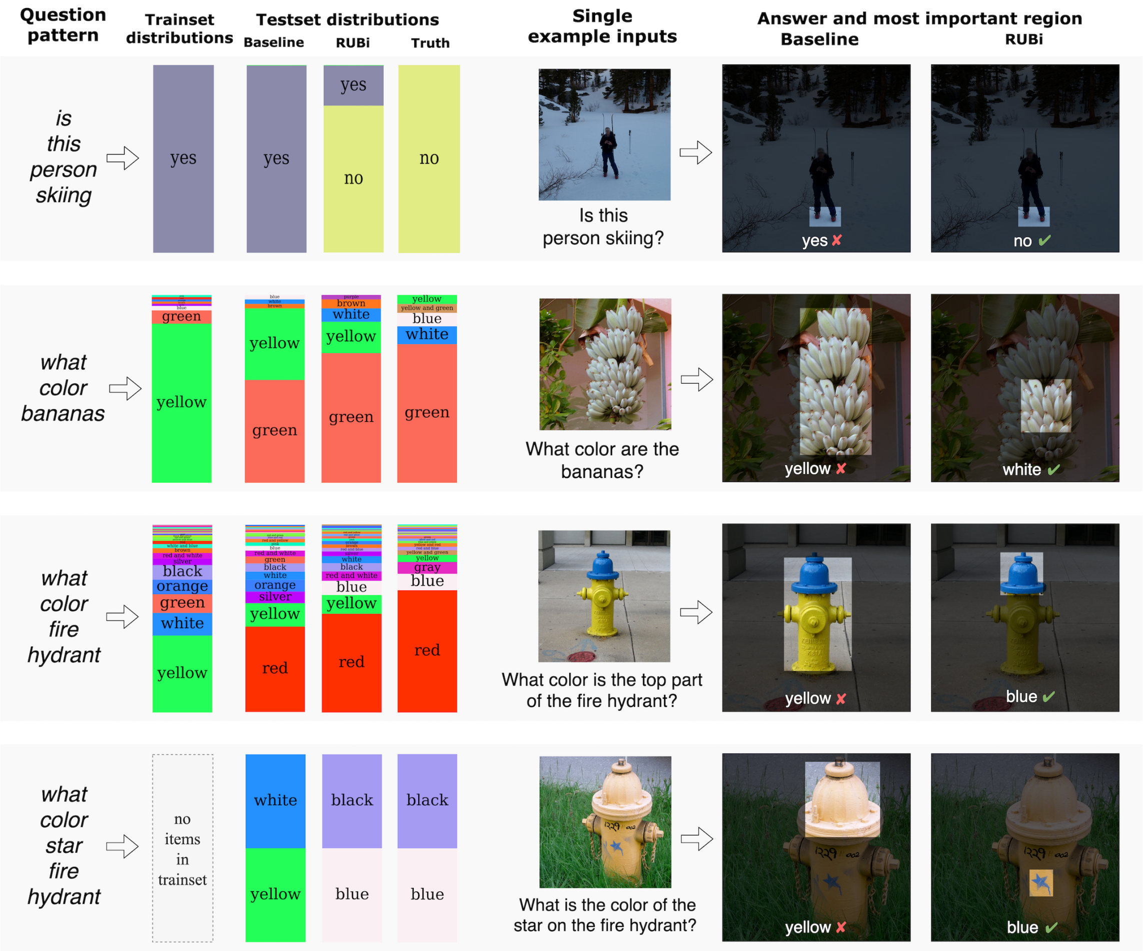


Figure 4.6 – Qualitative comparison between the outputs of RUBi and our baseline on VQA-CP v2 `test`. On the left, we display distributions of answers for the train set, the baseline evaluated on the test set, RUBi on the test set and the ground truth answers from the test set. For each row, we filter questions in a certain way. In the first row, we keep the questions that exactly match the string *is this person skiing*. In the three other rows, we filter questions that respectively include the following words: *what color bananas*, *what color fire hydrant* and *what color star hydrant*. On the right, we display examples that contain the pattern from the left. For each example, we display the answer of our baseline and RUBi, as well as the best scoring region from their attention map.

Table 4.5 – Correlation with Human Attention Maps on VQA-HAT val set (Das et al. 2016), with or without RUBi learning strategy for three architectures (Baseline, SAN and UpDn).

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

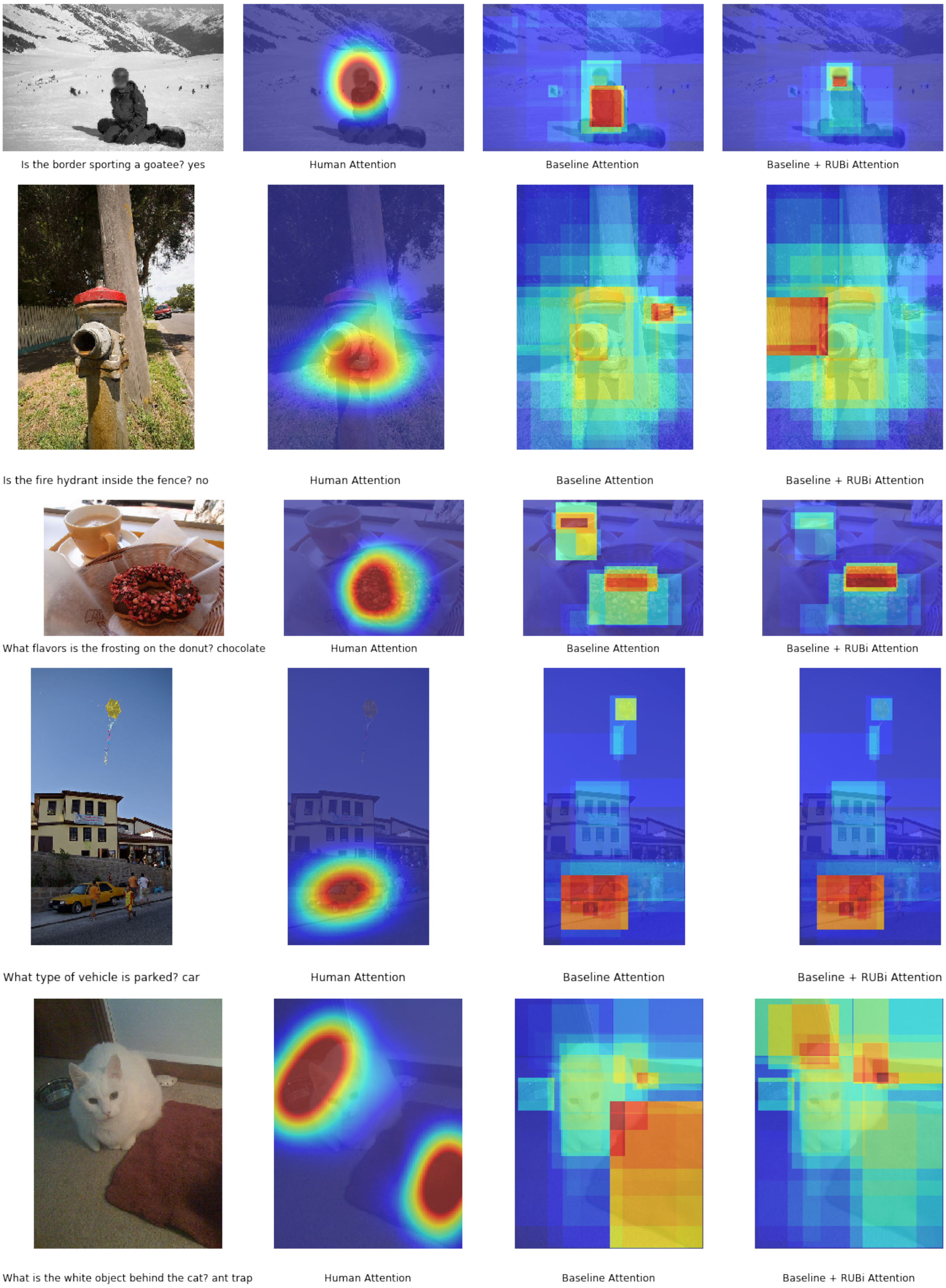
seems that our baseline gets it wrong by not seeing that the skis are not locked under the ski boots. This unwanted behavior could be due to the question biases.

In the second row, similar behaviors occur for the *what color are the bananas* question pattern. 80% of the answers from the train set are *yellow*, while most of them are *green* in the test set. We show that the amount of *green* and *white* answers from RUBi are much closer to the ones from the test set than with our baseline. In the example, it seems that RUBi relies on the color of the banana, while our baseline misses it.

In the third row, it seems that RUBi is able to ground the textual concepts such as *top part of the fire hydrant* and *color* on the right visual region, while the baseline relies on the correlations between the fire hydrant, the yellow color of its core and the answer *yellow*.

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

**Visualization of the impact of RUBi on VQA-HAT** We display in Figure 4.7 and Figure 4.8 some manually selected VQA triplets associated with the human attention maps provided by VQA-HAT (Das et al. 2016) and the attention maps computed from our baseline architecture when trained with and without RUBi. In Figure 4.7, we observe that the attention maps with RUBi are closer to the human



Is the border sporting a goatee? yes — Human Attention — Baseline Attention — Baseline + RUBi Attention

Is the fire hydrant inside the fence? no — Human Attention — Baseline Attention — Baseline + RUBi Attention

What flavors is the frosting on the donut? chocolate — Human Attention — Baseline Attention — Baseline + RUBi Attention

What type of vehicle is parked? car — Human Attention — Baseline Attention — Baseline + RUBi Attention

What is the white object behind the cat? ant trap — Human Attention — Baseline Attention — Baseline + RUBi Attention

Figure 4.7 – Examples of better grounding ability on VQA-HAT implied by RUBi. From the left column to the right: image-question-answer triplet, human attention map introduced by Das et al. 2016, attention map from our baseline, attention map from our baseline trained with RUBi.

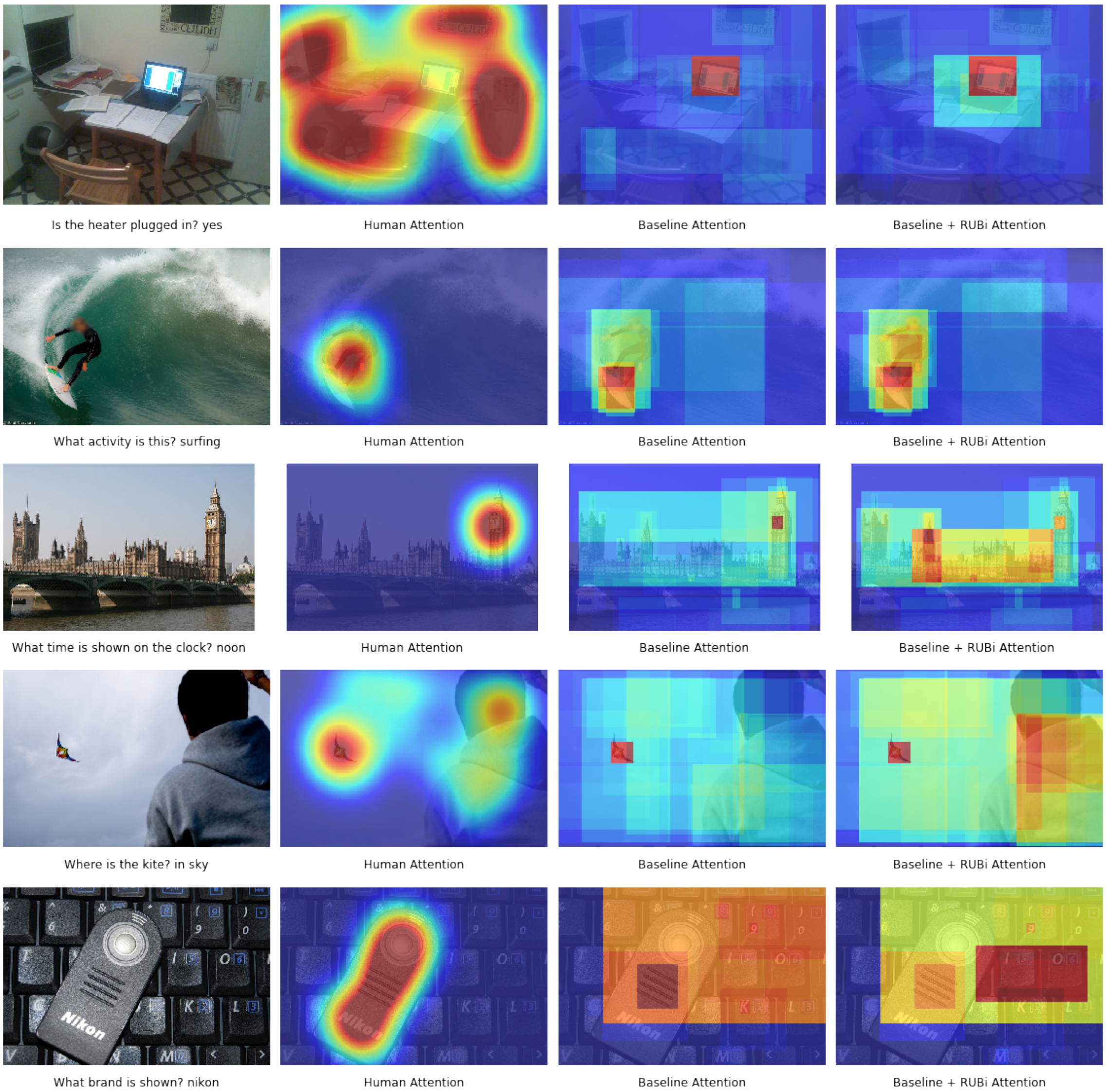

Is the heater plugged in? yes
Human Attention
Baseline Attention
Baseline + RUBi Attention
What activity is this? surfing
Human Attention
Baseline Attention
Baseline + RUBi Attention
What time is shown on the clock? noon
Human Attention
Baseline Attention
Baseline + RUBi Attention
Where is the kite? in sky
Human Attention
Baseline Attention
Baseline + RUBi Attention
What brand is shown? nikon
Human Attention
Baseline Attention
Baseline + RUBi Attention

Figure 4.8 – Examples of failure to improve grounding ability on VQA-HAT. From the left column to the right: image-question-answer triplet, human attention map introduced by Das et al. 2016, attention map from our baseline, attention map from our baseline trained with RUBi.

attention maps than without RUBi. On the contrary, we observe in Figure 4.8 some failure to improve grounding ability.

### 4.4.4 Implementation details

**Software, hardware and pretrained models** We use pytorch 1.1.0 to implement our algorithms in order to benefit from the GPU acceleration. We use four NVidia Titan Xp GPU in this study. We use a single GPU for each experiment. We use a dedicated SSD to load the visual features using multiple threads. A single experiment from Table 4.1 with the baseline architecture trained with or without RUBi takes less than five hours to run. Our code and pretrained models can be found on github:

- `github.com/cdancette/rubi.bootstrap.pytorch`

**Image encoder** We use the pretrained Faster R-CNN (S. Ren et al. 2015) by Anderson et al. 2018 on the Visual Genome dataset (Krishna et al. 2017) to extract object features. We use the setup that extracts 36 regions for each image. We do not fine-tune the image extractor.

**Question encoder** We use the same preprocessing as Cadène* et al. 2019. We apply a lower case transformation and remove the punctuation. We only consider the most frequent 3000 answers for both VQA v2 and VQA CP v2. We then use a pretrained Skip-thought encoder with a two-glimpses self-attention mechanism. The final embedding is of size 4800. We fine-tune the question encoder during training.

**Baseline architecture** Our baseline architecture is a simplified version of the MuRel network (Cadène* et al. 2019) defined in Section 3.4.1. First, it computes a bilinear fusion between the question vector and the visual features for each region. This fusion module is a BLOCK (Ben-Younes et al. 2019) defined in Section 3.3.2 and composed of 15 chunks, each of rank 15. The dimension of the projection space is 1000, and the output dimension is 2048. The output of the bilinear fusion is aggregated using a max-pooling over $n_v$ regions. The resulting vector is then fed into a MLP classifier composed of three layers of size (2048, 2048, 3000), with ReLU activations. It outputs the predictions over the space of the 3000 answers.

**Question-only branch** The RUBi question-only branch feeds the question into a first MLP composed of three layers, of size (2048, 2048, 3000), with ReLU activations. First, this output vector goes through a sigmoid to compute the mask that will alter the predictions of the VQA model. Secondly, this same output vector goes through a single linear layer of size 3000. We use these question-only predictions to compute the question-only loss.

**Optimization process** We train all our models with the Adam optimizer and the standard cross-entropy loss function for multi-class classification problems. We use a learning rate of $1.5 \times 10^{-4}$ and a batch size of 256. During the first 7 epochs, we linearly increase the learning rate to $6 \times 10^{-4}$. After epoch 14, we apply a learning rate decay strategy which multiplies the learning rate by 0.25 every two epochs. We train our models until convergence as we do not have a validation set for VQA-CP v2. For the UpDn and SAN architectures, we follow the optimization procedure described in Ramakrishnan et al. 2018.

## 4.5 Conclusion

ML models tend to learn unwanted biases by giving too much importance to some predictive features. We addressed this problem in the multimodal context of the VQA task where models overrely on the question modality. We proposed RUBi, a strategy that reduces the unimodal biases learned during training. RUBi leverages a question-only model that captures the language biases by identifying when a given question can be answered without looking at the image. It prevents the base VQA model from learning them by influencing its predictions. This leads to dynamically adjusting the loss in order to compensate for biases.

Compared to the classical training strategy, RUBi led to significant gains when used on three different VQA architectures and evaluated on datasets that penalize biased models towards the question modality. Interestingly, RUBi provided competitive performances on the original VQA v2 dataset while reducing the use of biases. Compared to the state-of-the-art approaches, RUBi also provided significant gains. Finally, we provided qualitative examples and further analysis on the VQA-HAT dataset to evaluate the grounding ability of models trained with RUBi.

Chapter 5

# CONCLUSION

## Contents

## 5.1 Summary of contributions

In this dissertation, we developed multimodal neural architectures and training procedures to connect visual and textual modalities. We summarize our contributions which address major challenges of multimodal learning.

**Multimodal alignment** One of the core challenges of multimodal learning consists in aligning two modalities in a shared space to perform crossmodal retrieval. In Chapter 2, we tackled this alignment challenge by proposing AdaMine. Our approach is based on two crossmodal triplet losses to align matching image-text pairs. Contrarily to previous works, we added two other crossmodal triplet losses that leverage the additional semantic information to align image-text pairs of the same class at no cost in terms of parameters. Additionally, we introduced an adaptive strategy for triplet mining, which reduces the issue of gradient vanishing in each loss. We validated our alignment contributions on Recipe1M, the largest dataset of image-text pairs associated with additional class information. AdaMine provided significant improvements over the best state-of-the-art approaches. To ensure the correctness of our evaluation protocol, we reproduced results of the state-of-the-art approaches. We also proposed a stronger baseline which is a pairwise loss with a positive margin and a negative margin. Finally, we conducted a further analysis and highlighted abilities of our models to connect textual concepts with visual ones on several downstream tasks.

**Multimodal fusion and reasoning** A second challenge consists in fusing the visual and textual modalities to create rich multimodal representations resulting from their interactions. In Chapter 3, we tackled this challenge in the context of the Visual Question Answering (VQA) task. We proposed a theoretically grounded fusion framework based on different factorization of a bilinear model. We leveraged this framework to propose two novel learnable fusion modules. MUTAN is based on the Tucker decomposition of the 3-order tensor of parameters which defines the bilinear interactions between both modalities. BLOCK is based on the Block-term decomposition and generalizes MUTAN. In the line of previous works, we embedded our fusions in attentional architectures. We identified their limitations and proposed MuRel. Our reasoning architecture iteratively fuses object-based visual representations with the question representation. It is made of several MuRel cells that share their parameters. Each cell includes a pairwise modeling between visual regions which adds relational and contextual information in the multimodal representations. Instead of relying on the widely used question-driven attention to remove the spatial dimensions, MuRel uses a simple averaging between each visual region representations.

We validated our fusion contributions on large-scale and widely used VQA datasets. We found that MUTAN and BLOCK are competitive with the best fusion modules from the literature while allowing a much lower number of free parameters. BLOCK was even able to reach a significantly better accuracy than the best fusion. We also found that MuRel reaches competitive performances against the best reasoning architectures and significantly surpasses the question-driven attention baseline. Finally, we confirmed that MuRel iteratively refines its focus on object regions.

**Unimodal biases** We addressed a last challenge regarding the unimodal biases that can be leveraged by statistical models. In Chapter 4, we proposed RUBi as a learning strategy to reduce the impact of these unwanted biases. RUBi leverages a question-only model that captures the language biases by identifying when a given question can be answered without looking at the image. It prevents the base VQA model from learning them by influencing its predictions. This leads to dynamically adjusting the loss in order to compensate for biases.

We validated our contribution on VQA datasets specifically tailored to penalize models that overrely on the question modality. On a classical VQA dataset, we found that RUBi leads to competitive performance, even though resulting models do not exploit the unwanted strong statistical correlations between the question and the answer.

Additionally, I was the core contributor to several open source libraries for accelerating research. A first one, *pretrained-models.pytorch* (Cadene 2017), contains numerous pretrained convolutional neural networks for image processing and has been used by hundreds of deep learning practitioners. It led to the publication of an academic journal (Bianco et al. 2018) in the context of an external collaboration. A second one, *skip-thoughts.torch* (Cadene 2018), contains pretrained language models with the skipthought learning strategy. A third one, *bootstrap.pytorch* (Cadene et al. 2018), is an experimental framework allowing for fast experimentation and accurate reproduction of experimental results. This framework has been presented twice at the PyTorch conference in San Francisco, CA. It has notably been used by several scholars to reproduce our results and extend our methods. These three libraries have been the foundations of the experimental framework of this thesis. Also, I had the opportunity to validate the knowledge acquired during this thesis by doing a research internship at Tesla for three months. It allowed me to participate in a research effort towards the creation of an autonomous car. Even in the absence of textual information, I could apply the same scientific framework and methodologies that I contributed to develop during this thesis.

## 5.2 Perspectives for future work

In the context of deep learning, the development of intelligent systems is in part conditioned on the amount of data and the computing power. These two factors are currently growing at exponential rates. However, fundamental research efforts are also critical for the development of the field. We focus here on research perspectives that are left to be explored in the near future, not forgetting long term goals.

### Improving multimodal representations

**Pretrained models** In Chapter 2, Chapter 3 and Chapter 4, we extensively used pretrained models for vision or text. More powerful pretrained models recently emerged with the findings of novel large-scale training procedures from weakly supervised learning (Joulin et al. 2016; Mahajan et al. 2018; Tan et al. 2019; Lu et al. 2019) or self-supervised learning (Vaswani et al. 2017; Devlin et al. 2018; He et al. 2019). Also, new approaches to use them emerged such as the processing of bigger images (Engilberge et al. 2018). The development of pretrained models will undoubtedly impact multimodal learning. However, it might not be straightforward to use them with the current multimodal methods including those that are developed in this PhD thesis. Also, these more power pretrained models could lead to the creation of new methods. More work should be done to use them in their full potential.

**Theoretical fusion framework** In Chapter 3, we proposed a theoretical framework based on the notion of rank of a 3-order tensor that parameterizes a bilinear model. However, many fusions do not fit any theoretical framework such as fusions based on higher-order interactions (Z. Yu et al. 2017; Z. Yu et al. 2018; Do et al. 2019) or fusions based on transformer co-attention (Z. Yu et al. 2019; Gao et al. 2019). For future work, it seems important to provide a thorough analysis of the state-of-the-art fusions and to push towards the elaboration of a global theoretical framework.

**Generative models** We recently saw the emergence of multimodal approaches based on generative models. For the crossmodal retrieval task which was the subject of Chapter 2, these generative models allow to better structure the multimodal retrieval space (Gu et al. 2018; H. Wang et al. 2019; B. Zhu et al. 2019). For the VQA task, they allow enriching and robustifying the multimodal representations (F. Liu et al. 2018; Yikang Li et al. 2018; Shah et al. 2019). Even though these approaches tend to significantly increase the number of hyperparameters, they could play an important role in the exploitation of unlabeled data. This could allow to reduce the annotation cost and improve the performances.

## Learning better behaviors

**Reasoning architecture** In the same vein of our works in Chapter 3, several attempts have been made to improve the reasoning ability and interpret ability of VQA models in the context of real data (Andreas et al. 2016; R. Hu et al. 2018; J. Shi et al. 2019). It seems promising to push forward the efforts of finding novel inductive biases (Battaglia et al. 2018). Especially, those allowing to generalize in the zero-shot learning context (Yuanpeng Li et al. 2018).

**Metrics and datasets** In Chapter 4, we saw that validating correct behaviors is challenging. Reporting accuracy scores on current VQA datasets is not enough to assess the correctness of the learned behaviors. It seems critical to find new methodologies to evaluate the abilities of VQA models to answer questions about images. One way could be through the creation of carefully tailored and annotated testing sets associated with behavioral metrics such as the grounding ability (Das et al. 2016).

## Advanced AI systems

Despite significant advances in the past decades, machines are still far from understanding the complexity of our visual world and far from mastering the richness of human language. For instance, explaining what is funny in a visual scene is totally out of reach for current AI systems. We could hypothesis that incredibly rich and complex multimodal representations are at the core of these high-level abilities. Solutions to the problem of forming these representations are yet to be found. Nevertheless, we can put forward a few approaches that seem promising in the future.

**Embodied interactions** Our human way to form knowledge about the world is through interacting with it. We possess an internal model of the world and the ability to form hypotheses. Similarly to the scientific approach, we acquire new knowledge through the validation or invalidation of our hypotheses. An argument can be made that the embodied interactions are critical in the process of forming these rich multimodal representations that could sustain high-level abilities. Thus, AI systems might never reach human-level behaviors and understanding of the world without being embodied and able to interact.

**Neuroscience** A complementary way to reach human-level behaviors in machines could be through neuroscience. Convolutional neural network is an example of bio-inspired model (Fukushima 1980) which led to a breakthrough in representation learning. It seems important to discover the priors and innate abilities that have been crafted by evolution to sustain survival (J. Kim et al. 2020; Linsley et al. 2020). Especially, developing a human-like ability to learn could be a critical step forward (Marblestone et al. 2016; Hassabis et al. 2017; Dehaene 2018; Zador 2019).

# BIBLIOGRAPHY

Agrawal, Aishwarya, Dhruv Batra, and Devi Parikh (2016). "Analyzing the behavior of visual question answering models". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (cit. on p. 80).

Agrawal, Aishwarya, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi (2018). "Don't just assume; look and answer: Overcoming priors for visual question answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 64, 71, 75, 80–84, 89–92).

Agrawal, Aishwarya, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh (2015). "VQA: Visual Question Answering". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on p. 4).

Anand, Ankesh, Eugene Belilovsky, Kyle Kastner, Hugo Larochelle, and Aaron Courville (2018). "Blindfold baselines for embodied qa". In: *arXiv preprint arXiv:1811.05013* (cit. on p. 82).

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: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 12, 52, 54, 62, 64, 69, 75, 81, 88–90, 98).

Andreas, Jacob, Marcus Rohrbach, Trevor Darrell, and Dan Klein (2016). "Neural module networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 66, 104).

Andrew, Galen, Raman Arora, Jeff Bilmes, and Karen Livescu (2013). "Deep canonical correlation analysis". In: *International Conference on Machine Learning (ICML)* (cit. on p. 22).

Antol, Stanislaw, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh (2015). "VQA: Visual Question Answering". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on pp. 48–53, 82, 83, 89).

Bach, Francis R and Michael I Jordan (2002). "Kernel independent component analysis". In: *Journal of machine learning research (JMLR)* 3.Jul, pp. 1–48 (cit. on p. 22).

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015). "Neural machine translation by jointly learning to align and translate". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on p. 53).

Bai, Yalong, Jianlong Fu, Tiejun Zhao, and Tao Mei (2018). "Deep Attention Neural Tensor Network for Visual Question Answering". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on p. 69).

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 pp. 63, 104).

Ben-Younes, Hedi, Rémi Cadène, 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 (AAAI)*. URL: https://arxiv.org/abs/1902.00038 (cit. on pp. 18, 48, 84, 88, 98).

Ben-Younes*, Hedi, Remi Cadene*, Nicolas Thome, and Matthieu Cord (2017a). “VQA Challenge Workshop: MUTAN 2.0”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). VQA Challenge and Visual Dialog Workshop* (cit. on pp. 18, 48).

Ben-Younes*, Hedi, Remi Cadene*, Nicolas Thome, and Matthieu Cord (2018). “VQA Challenge Workshop: Bilinear Superdiagonal Fusion”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). VQA Challenge and Visual Dialog Workshop* (cit. on pp. 18, 48).

Ben-Younes*, Hedi, Rémi Cadène*, Nicolas Thome, and Matthieu Cord (2017b). “MUTAN: Multimodal Tucker Fusion for Visual Question Answering”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. URL: https://arxiv.org/abs/1705.06676 (cit. on pp. 17, 48, 49, 65, 69, 71, 84).

Bianco, Simone, Remi Cadene, Luigi Celona, and Paolo Napoletano (2018). “Benchmark Analysis of Representative Deep Neural Network Architectures”. In: *IEEE Access* 6, pp. 64270–64277. URL: https://arxiv.org/abs/1810.00736 (cit. on p. 102).

Bishop, Christopher M (2006). *Pattern recognition and machine learning*. Springer (cit. on pp. 5, 6).

Bossard, Lukas, Matthieu Guillaumin, and Luc Van Gool (2014). “Food-101 – Mining Discriminative Components with Random Forests”. In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on p. 20).

Cadene, Remi (2017). *Pretrained convolutional neural networks library in PyTorch*. URL: http://github.com/Cadene/pretrained-models.pytorch (cit. on pp. 11, 102).

Cadene, Remi (2018). *Pretrained skipthoughts models in Torch7 and PyTorch*. URL: http://github.com/Cadene/skip-thoughts.torch (cit. on p. 102).

Cadene, Remi, Micael Carvalho, Hedi Ben-Younes, Thibaut Durand, Thomas Robert, Corentin Dancette, and Matthieu Cord (2018). “Bootstrap.pytorch: a high-level extension for deep learning projects in PyTorch”. In: *PyTorch Conference*. URL: http://github.com/Cadene/bootstrap.pytorch (cit. on p. 102).

Cadène*, Rémi, Hedi Ben-Younes*, Nicolas Thome, and Matthieu Cord (2019). "MUREL: Multimodal Relational Reasoning for Visual Question Answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. URL: https://arxiv.org/abs/1902.09487 (cit. on pp. 18, 48, 84, 88, 90, 93, 98).

Cadene*, Remi, Corentin Dancette*, Hedi Ben-Younes, Matthieu Cord, and Devi Parikh (2019). "RUBi: Reducing Unimodal Biases for Visual Question Answering". In: *Advances in Neural Information Processing Systems (NeurIPS)*. URL: https://arxiv.org/abs/1906.10169 (cit. on pp. 18, 80).

Carroll, J. Douglas and Jih-Jie Chang (1970). "Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition". In: *Psychometrika* (cit. on p. 58).

Carvalho*, Micael, Remi Cadene*, David Picard, Laure Soulier, and Matthieu Cord (2018a). "Images & Recipes: Retrieval in the cooking context". In: *Proceedings of the IEEE International Conference on Data Engineering (ICDE). Data Engineering meets Intelligent Food and Cooking Recipe Workshop (DECOR)*. URL: https://arxiv.org/abs/1805.00900 (cit. on pp. 17, 20).

Carvalho*, Micael, Remi Cadene*, David Picard, Laure Soulier, Nicolas Thome, and Matthieu Cord (2018b). "Cross-modal retrieval in the cooking context: Learning semantic text-image embeddings". In: *ACM Conference on Research and Development in Information Retrieval (SIGIR)*. URL: https://arxiv.org/abs/1804.11146 (cit. on pp. 17, 20).

Chao, Wei-Lun, Hexiang Hu, and Fei Sha (2018). "Being negative but constructively: Lessons learnt from creating better visual question answering datasets". In: (cit. on p. 82).

Charikar, Moses, Kevin Chen, and Martin Farach-Colton (2002). "Finding Frequent Items in Data Streams". In: *International Colloquium on Automata, Languages and Programming* (cit. on p. 52).

Chechik, Gal, Varun Sharma, Uri Shalit, and Samy Bengio (2010). "Large scale online learning of image similarity through ranking". In: *Journal of machine learning research (JMLR)* 11.Mar, pp. 1109–1135 (cit. on p. 23).

Chen, Jingjing and Chong-Wah Ngo (2016). "Deep-based ingredient recognition for cooking recipe retrieval". In: *Proceedings of the 2016 ACM on Multimedia Conference* (cit. on pp. 20, 24, 32).

Chen, Jingjing, Lei Pang, and Chong-Wah Ngo (2017). "Cross-Modal Recipe Retrieval: How to Cook this Dish?" In: *MultiMedia Modeling* (cit. on p. 20).

Chen, Xinlei, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick (2015). "Microsoft coco captions: Data collection and evaluation server". In: *arXiv preprint arXiv:1504.00325* (cit. on pp. 3, 4, 24).

Chen, Zhu, Zhao Yanpeng, Huang Shuaiyi, Tu Kewei, and Ma Yi (2017). "Structured Attentions for Visual Question Answering". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on pp. 49, 53).

Chung, Junyoung, Çağlar Gülçehre, Kyunghyun Cho, and Yoshua Bengio (2014). "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling". In: *Advances in Neural Information Processing Systems (NIPS). Deep Learning workshop* (cit. on pp. 14, 22, 53).

Clinchant, Stéphane, Julien Ah-Pine, and Gabriela Csurka (2011). "Semantic combination of textual and visual information in multimedia retrieval". In: *Proceedings of the 1st ACM international conference on multimedia retrieval* (cit. on p. 25).

Coates, Adam and Andrew Y Ng (2011). "The importance of encoding versus training with sparse coding and vector quantization". In: *International Conference on Machine Learning (ICML)* (cit. on p. 24).

Cortes, Corinna and Vladimir Vapnik (1995). "Support-vector networks". In: *Machine learning* 20.3, pp. 273–297 (cit. on p. 7).

Csurka, Gabriella, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray (2004). "Visual categorization with bags of keypoints". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV). Workshop on statistical learning in computer vision* (cit. on p. 2).

Das, Abhishek, Harsh Agrawal, C. Lawrence Zitnick, Devi Parikh, and Dhruv Batra (2016). "Human Attention in Visual Question Answering: Do Humans and Deep Networks Look at the Same Regions?" In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (cit. on pp. 82, 93, 95–97, 104).

Das, Abhishek, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra (2017). "Visual dialog". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 5, 49).

De Lathauwer, L. (2008). "Decompositions of a Higher-Order Tensor in Block Terms — Part II: Definitions and Uniqueness". In: *SIAM J. Matrix Anal. Appl.* 30.3, pp. 1033–1066 (cit. on pp. 57, 58).

De Marneffe, Marie-Catherine, Bill MacCartney, Christopher D Manning, et al. (2006). "Generating typed dependency parses from phrase structure parses." In: *The International Conference on Language Resources and Evaluation (LREC)* (cit. on p. 25).

Dehaene, Stanislas (2018). *Apprendre!: Les talents du cerveau, le défi des machines*. Odile Jacob (cit. on p. 105).

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *arXiv preprint arXiv:1810.04805* (cit. on pp. 3, 22, 103).

Do, Tuong, Thanh-Toan Do, Huy Tran, Erman Tjiputra, and Quang D Tran (2019). “Compact Trilinear Interaction for Visual Question Answering”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on p. 103).

Donahue, Jeff, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell (2014). “Decaf: A deep convolutional activation feature for generic visual recognition”. In: *International conference on machine learning* (cit. on p. 11).

Duchi, John, Elad Hazan, and Yoram Singer (2011). “Adaptive subgradient methods for online learning and stochastic optimization”. In: *Journal of machine learning research (JMLR)* 12.Jul, pp. 2121–2159 (cit. on p. 10).

Durand, Thibaut (2017). “Weakly supervised learning for visual recognition”. PhD thesis. Sorbonne Université (cit. on p. 11).

Durand, Thibaut, Taylor Mordan, Nicolas Thome, and Matthieu Cord (2017). “WILDCAT: Weakly Supervised Learning of Deep ConvNets for Image Classification, Pointwise Localization and Segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 12).

Durand, Thibaut, Nicolas Thome, and Matthieu Cord (2016). “WELDON: Weakly Supervised Learning of Deep Convolutional Neural Networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 12).

Eisenschtat, Aviv and Lior Wolf (2017). “Linking image and text with 2-way nets”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 23, 25).

Elman, Jeffrey L (1990). “Finding structure in time”. In: *Cognitive science* 14.2, pp. 179–211 (cit. on p. 13).

Elsweiler, David, Christoph Trattner, and Morgan Harvey (2017). “Exploiting food choice biases for healthier recipe recommendation”. In: *ACM Conference on Research and Development in Information Retrieval (SIGIR)* (cit. on p. 5).

Engilberge, Martin, Louis Chevallier, Patrick Pérez, and Matthieu Cord (2018). “Finding beans in burgers: Deep semantic-visual embedding with localization”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 103).

Engilberge, Martin, Louis Chevallier, Patrick Pérez, and Matthieu Cord (2019). “SoDeep: a Sorting Deep net to learn ranking loss surrogates”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 24).

Faghri, Fartash, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler (2018). “VSE++: Improved Visual-Semantic Embeddings”. In: *Proceedings of the British Machine Vision Conference (BMVC)* (cit. on pp. 21, 23, 25).

Fournier, Jérôme, Matthieu Cord, and Sylvie Philipp-Foliguet (2001). “Retin: A content-based image indexing and retrieval system”. In: *Pattern Analysis & Applications* 4.2-3, pp. 153–173 (cit. on p. 2).

Frome, Andrea, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov (2013). "Devise: A deep visual-semantic embedding model". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on p. 25).

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: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (cit. on pp. 49, 52, 53, 64–68, 70, 75).

Fukushima, Kunihiko (1980). "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biological cybernetics* 36.4, pp. 193–202 (cit. on pp. 10, 105).

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: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 103).

Girshick, Ross (2015). "Fast r-cnn". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on p. 12).

Girshick, Ross, Jeff Donahue, Trevor Darrell, and Jitendra Malik (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 2, 10, 12, 25).

Glorot, Xavier and Yoshua Bengio (2010). "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the International Conference on Artificial Intelligence and Statistics (ICASP)* (cit. on p. 9).

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep learning*. MIT Press (cit. on p. 5).

Gordon, Jonathan and Benjamin Van Durme (2013). "Reporting bias and knowledge acquisition". In: *International Conference on Information and Knowledge Management. Proceedings of the 2013 workshop on Automated knowledge base construction* (cit. on p. 82).

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: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 4, 49, 63, 64, 66–68, 75, 80–83, 89).

Gu, Jiuxiang, Jianfei Cai, Shafiq R Joty, Li Niu, and Gang Wang (2018). "Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 104).

Gupta, Abhinav, Adithyavairavan Murali, Dhiraj Prakashchand Gandhi, and Lerrel Pinto (2018). "Robot learning in homes: Improving generalization and reducing dataset bias". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on p. 82).

Gurari, Danna, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham (2018). "VizWiz Grand Challenge: Answering Visual Questions from Blind People". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 5, 49).

Hadsell, R., S. Chopra, and Y. LeCun (2006). "Dimensionality Reduction by Learning an Invariant Mapping". In: *CVP)*, pp. 1735–1742 (cit. on p. 23).

Harshman, Richard A., Peter Ladefoged, Hans Reichenbach, Robert I. Jennrich, Dale Terbeek, Lee Cooper, Andrew Comrey, P. M. Bentler, Jeanne Yamane, Diane Vaughan, and Bill Jahnke (2001). "Foundations of the Parafac Procedure: Models and Conditions for an "explanatory" Multimodal Factor Analysis". In: *UCLA Working Phonetics Paper* (cit. on p. 58).

Hassabis, Demis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick (2017). "Neuroscience-inspired artificial intelligence". In: *Neuron* 95.2, pp. 245–258 (cit. on p. 105).

He, Kaiming, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick (2019). "Momentum contrast for unsupervised visual representation learning". In: *arXiv preprint arXiv:1911.05722* (cit. on p. 103).

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick (2017). "Mask R-CNN". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on pp. 2, 13).

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015a). "Deep Residual Learning for Image Recognition". In: *arXiv preprint arXiv:1512.03385* (cit. on p. 29).

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015b). "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on p. 9).

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 2, 11, 22).

Hendricks, Lisa Anne, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach (2018). "Women Also Snowboard: Overcoming Bias in Captioning Models". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on p. 82).

Hochreiter, Sepp (1991). "Untersuchungen zu dynamischen neuronalen Netzen". In: *Diploma, Technische Universität München* 91.1 (cit. on p. 11).

Hochreiter, Sepp, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. (2001). "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies". In: *A field guide to dynamical recurrent neural networks*. Ed. by S. C. Kremer and J. F. Kolen. IEEE Press (cit. on pp. 11, 14).

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780 (cit. on pp. 14, 30, 53).

Hodosh, Micah, Peter Young, and Julia Hockenmaier (2013). "Framing image description as a ranking task: Data, models and evaluation metrics". In: *Journal of Artificial Intelligence Research* 47, pp. 853–899 (cit. on pp. 3, 24).

Hotelling, Harold (1936). "Relations between two sets of variates". In: *Biometrika* 28.3/4, pp. 321–377 (cit. on pp. 22, 33).

Hu, J., J. Lu, and Y. P. Tan (2014). "Discriminative Deep Metric Learning for Face Verification in the Wild". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 23, 33).

Hu, Jie, Li Shen, and Gang Sun (2018). "Squeeze-and-excitation networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 73).

Hu, Ronghang, Jacob Andreas, Trevor Darrell, and Kate Saenko (2018). "Explainable neural computation via stack neural module networks". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on p. 104).

Hu, Ronghang, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko (2017). "Learning to Reason: End-to-End Module Networks for Visual Question Answering". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on p. 54).

Huang, Eric H, Richard Socher, Christopher D Manning, and Andrew Y Ng (2012). "Improving word representations via global context and multiple word prototypes". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1* (cit. on p. 24).

Huang, Yan, Wei Wang, and Liang Wang (2017). "Instance-aware image and sentence matching with selective multimodal LSTM". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 25).

Hudson, Drew A and Christopher D Manning (2018). "Compositional attention networks for machine reasoning". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on p. 54).

Hudson, Drew A and Christopher D Manning (2019). "GQA: a new dataset for compositional question answering over real-world images". In: *arXiv preprint arXiv:1902.09506* (cit. on pp. 80–82).

Ilievski, Ilija and Jiashi Feng (2017). "Multimodal Learning and Reasoning for Visual Question Answering". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on p. 68).

Jabri, Allan, Armand Joulin, and Laurens Van Der Maaten (2016). "Revisiting visual question answering baselines". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on p. 82).

Jia, Sen, Thomas Lansdall-Welfare, and Nello Cristianini (2018). "Right for the Right Reason: Training Agnostic Networks". In: *Lecture Notes in Computer Science*, pp. 164–174 (cit. on p. 82).

Johnson, Justin, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick (2017a). "CLEVR: A Diagnostic Dataset for CompositionalLanguage and Elementary Visual Reasoning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 53, 80, 82).

Johnson, Justin, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick (2017b). "Inferring and Executing Programs for Visual Reasoning". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on p. 54).

Johnson, Justin, Andrej Karpathy, and Li Fei-Fei (2016). "Densecap: Fully convolutional localization networks for dense captioning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 5).

Joulin, Armand, Laurens van der Maaten, Allan Jabri, and Nicolas Vasilache (2016). "Learning Visual Features from Large Weakly Supervised Data". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on p. 103).

Kafle, Kushal and Christopher Kanan (2017). "An Analysis of Visual Question Answering Algorithms". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on pp. 49, 63, 66, 70).

Karpathy, Andrej and Li Fei-Fei (2015). "Deep visual-semantic alignments for generating image descriptions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 20, 23, 25).

Karpathy, Andrej, Armand Joulin, and Li F Fei-Fei (2014). "Deep fragment embeddings for bidirectional image sentence mapping". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on p. 25).

Kawano, Yoshiyuki and Keiji Yanai (2014). "Food image recognition with deep convolutional features". In: *The ACM international joint conference on pervasive and ubiquitous computing (UbiComp)* (cit. on p. 20).

Kim, Jin-Hwa, Jaehyun Jun, and Byoung-Tak Zhang (2018). "Bilinear attention networks". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on pp. 53, 70, 77, 84, 90).

Kim, Jin-Hwa, Sang-Woo Lee, Donghyun Kwak, Min-Oh Heo, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang (2016). "Multimodal Residual Learning for Visual QA". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on pp. 52, 53).

Kim, Jin-Hwa, Kyoung Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang (2017). "Hadamard Product for Low-rank Bilinear Pooling". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on pp. 49, 52–54, 58, 65, 69).

Kim, Junkyung, Drew Linsley, Kalpit Thakkar, and Thomas Serre (2020). "Disentangling neural mechanisms for perceptual grouping". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on p. 105).

Kingma, Diederik and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv arXiv:1412.6980* (cit. on pp. 10, 45, 64, 77).

Kiros, Ryan, Ruslan Salakhutdinov, and Richard S Zemel (2014). "Unifying visual-semantic embeddings with multimodal neural language models". In: *arXiv preprint arXiv:1411.2539* (cit. on pp. 20, 23, 25).

Kiros, Ryan, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler (2015). "Skip-Thought Vectors". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on pp. 3, 14, 15, 22, 26, 30, 53, 64, 75, 88).

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: *International Journal of Computer Vision (IJCV)* 123.1, pp. 32–73 (cit. on pp. 52, 63, 67, 68, 70, 75, 98).

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on pp. 2, 8, 10, 11, 22, 25).

Kusmierczyk, Tomasz, Christoph Trattner, and Kjetil Nørvåg (2016). "Understanding and predicting online food recipe production patterns". In: *27th ACM Conference on Hypertext and Social Media (HT)* (cit. on pp. 24, 32).

Lai, Pei Ling and Colin Fyfe (2000). "Kernel and nonlinear canonical correlation analysis". In: *International Journal of Neural Systems* 10.05, pp. 365–377 (cit. on pp. 22, 24).

Lample, Guillaume, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer (2016). "Neural architectures for named entity recognition". In: *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)* (cit. on p. 3).

Law, Marc T, Nicolas Thome, and Matthieu Cord (2013). "Quadruplet-wise image similarity learning". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on p. 24).

Lazaridou, Angeliki, Nghia The Pham, and Marco Baroni (2015). "Combining Language and Vision with a Multimodal Skip-gram Model". In: *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)* (cit. on p. 20).

LeCun, Yann, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel (1989). "Backpropagation applied to handwritten zip code recognition". In: *Neural computation* 1.4, pp. 541–551 (cit. on p. 10).

LeCun, Yann, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324 (cit. on p. 9).

Li, Ruiyu and Jiaya Jia (2016). "Visual Question Answering with Question Representation Update (QRU)". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on pp. 52, 53).

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 (CVPR)* (cit. on p. 104).

Li, Yuanpeng, Yi Yang, Jianyu Wang, and Wei Xu (2018). "Zero-Shot Transfer VQA Dataset". In: *arXiv preprint arXiv:1811.00692* (cit. on p. 104).

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: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on pp. 3, 24, 63).

Lin, Tsung-Yu, Aruni RoyChowdhury, and Subhransu Maji (2015). "Bilinear CNN Models for Fine-grained Visual Recognition". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on p. 56).

Linsley, Drew, Junkyung Kim, Alekh Ashok, and Thomas Serre (2020). "The function of contextual illusions". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on p. 105).

Liu, Feng, Tao Xiang, Timothy M Hospedales, Wankou Yang, and Changyin Sun (2018). "iVQA: Inverse visual question answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 104).

Liu, Rosanne, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski (2018). "An intriguing failing of convolutional neural networks and the coordconv solution". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on p. 73).

Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg (2016). "Ssd: Single shot multibox detector". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on p. 13).

Long, Jonathan, Evan Shelhamer, and Trevor Darrell (2015). "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 10, 12).

Lowe, David G (2004). "Distinctive image features from scale-invariant keypoints". In: *International Journal of Computer Vision (IJCV)* 60.2, pp. 91–110 (cit. on p. 24).

Lu, Jiasen, Dhruv Batra, Devi Parikh, and Stefan Lee (2019). "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks". In: *Advances in Neural Information Processing Systems (NeurIPS)* (cit. on p. 103).

Lu, Jiasen, Xiao Lin, Dhruv Batra, and Devi Parikh (2015). *Deeper LSTM and normalized CNN visual question answering model*. URL: https://github.com/GT-Vision-Lab/VQA_LSTM_CNN (cit. on p. 68).

Lu, Jiasen, Caiming Xiong, Devi Parikh, and Richard Socher (2017). "Knowing when to look: Adaptive attention via a visual sentinel for image captioning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 3).

Lu, Jiasen, Jianwei Yang, Dhruv Batra, and Devi Parikh (2016a). "Hierarchical Question-Image Co-Attention for Visual Question Answering". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on p. 53).

Lu, Jiasen, Jianwei Yang, Dhruv Batra, and Devi Parikh (2016b). "Hierarchical question-image co-attention for visual question answering". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on p. 84).

Mahajan, Dhruv, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten (2018). "Exploring the limits of weakly supervised pretraining". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on p. 103).

Malinowski, Mateusz, Carl Doersch, Adam Santoro, and Peter Battaglia (2018). "Learning Visual Question Answering by Bootstrapping Hard Attention". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on p. 71).

Malinowski, Mateusz and Mario Fritz (2014a). "A Multi-World Approach to Question Answering about Real-World Scenes based on Uncertain Input". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on pp. 4, 48, 50).

Malinowski, Mateusz and Mario Fritz (2014b). "Towards a Visual Turing Challenge". In: *Learning Semantics* (cit. on pp. 5, 49).

Malinowski, Mateusz, Marcus Rohrbach, and Mario Fritz (2015). "Ask your neurons: A neural-based approach to answering questions about images". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on p. 52).

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

Marblestone, Adam H, Greg Wayne, and Konrad P Kording (2016). "Toward an integration of deep learning and neuroscience". In: *Frontiers in computational neuroscience* 10, p. 94 (cit. on p. 105).

Marin, Javier, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba (2019). "Recipe1M+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (cit. on p. 24).

Mascharka, David, Philip Tran, Ryan Soklaski, and Arjun Majumdar (2018). "Transparency by Design: Closing the Gap Between Performance and Interpretability in Visual Reasoning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 54).

McCulloch, Warren S and Walter Pitts (1943). "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5.4, pp. 115–133 (cit. on p. 7).

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013a). "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (cit. on p. 25).

Mikolov, Tomáš, Martin Karafiát, Lukáš Burget, Jan Černock, and Sanjeev Khudanpur (2010). "Recurrent neural network based language model". In: *Eleventh annual conference of the international speech communication association* (cit. on p. 13).

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013b). "Distributed representations of words and phrases and their compositionality". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 3111–3119 (cit. on pp. 3, 13, 22, 25, 26, 30, 53).

Misra, Ishan, C Lawrence Zitnick, Margaret Mitchell, and Ross Girshick (2016). "Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 82).

Nam, Hyeonseob, Jung-Woo Ha, and Jeonghee Kim (2017). "Dual attention networks for multimodal reasoning and matching". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 25).

Nesterov, Yurii (1964). "A method of solving a convex programming problem with convergence rate O(1/sqr(k))". In: *Soviet Mathematics Doklady*. Elsevier, 32:372–376 (cit. on p. 10).

Nielsen, Michael (2015). *Neural networks and deep learning*. Online publication. URL: http://neuralnetworksanddeeplearning.com (cit. on p. 5).

Nocedal, Jorge and Stephen Wright (2006). *Numerical optimization*. Springer Science & Business Media (cit. on p. 8).

Noh, Hyeonwoo and Bohyung Han (2016). "Training recurrent answering units with joint loss minimization for vqa". In: *arXiv preprint arXiv:1606.03647* (cit. on pp. 66, 70).

Norcliffe-Brown, Will, Efstathios Vafeias, and Sarah Parisot (2018). "Learning Conditioned Graph Structures for Interpretable Visual Question Answering". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on pp. 53, 62, 69).

Oquab, Maxime, Leon Bottou, Ivan Laptev, and Josef Sivic (2014). "Learning and transferring mid-level image representations using convolutional neural networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 10).

Oquab, Maxime, Léon Bottou, Ivan Laptev, and Josef Sivic (2015). "Is object localization for free?-weakly-supervised learning with convolutional neural networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 12).

Pennington, Jeffrey, Richard Socher, and Christopher Manning (2014). "Glove: Global vectors for word representation". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (cit. on pp. 13, 22, 53).

Perez, Ethan, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville (2018). "FiLM: Visual Reasoning with a General Conditioning Layer". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)* (cit. on pp. 54, 73).

Perronnin, Florent, Jorge Sanchez, and Thomas Mensink (2010). "Improving the Fisher Kernel for Large-Scale Image Classification". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on p. 56).

Polyak, Boris T (1964). "Some methods of speeding up the convergence of iteration methods". In: *USSR Computational Mathematics and Mathematical Physics* 4.5, pp. 1–17 (cit. on p. 9).

Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang (2016). "SQuAD: 100,000+ Questions for Machine Comprehension of Text". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (cit. on p. 3).

Ramakrishnan, Sainandan, Aishwarya Agrawal, and Stefan Lee (2018). "Overcoming language priors in visual question answering with adversarial regularization". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on pp. 80, 83, 89–92, 99).

Rasiwasia, Nikhil, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert RG Lanckriet, Roger Levy, and Nuno Vasconcelos (2010). "A new approach to cross-modal multimedia retrieval". In: *Proceedings of the ACM International Conference on Multimedia* (cit. on p. 4).

Redmon, Joseph and Ali Farhadi (2017). "YOLO9000: better, faster, stronger". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 13).

Ren, Mengye, Ryan Kiros, and Richard S. Zemel (2015). "Exploring Models and Data for Image Question Answering". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on pp. 49, 51, 53).

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun (2015). "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on pp. 2, 12, 60, 75, 98).

Rippel, Oren, Manohar Paluri, Piotr Dollar, and Lubomir Bourdev (2016). "Metric learning with adaptive density discrimination". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on p. 24).

Rohrbach, Anna, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko (2018). "Object Hallucination in Image Captioning". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (cit. on p. 82).

Rosenblatt, Frank (1958). "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6, p. 386 (cit. on p. 7).

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams (1986). "Learning representations by back-propagating errors". In: *Nature* 323.6088, pp. 533–536 (cit. on p. 9).

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. (2015a). "Imagenet large scale visual recognition challenge". In: *International Journal of Computer Vision (IJCV)* 115.3, pp. 211–252 (cit. on pp. 2, 4).

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei (2015b). "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3, pp. 211–252 (cit. on pp. 11, 29, 52).

Salvador, Amaia, Nicholas Hynes, Yusuf Aytar, Javier Marin, Ferda Ofli, Ingmar Weber, and Antonio Torralba (2017). "Learning Cross-modal Embeddings for Cooking Recipes and Food Images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 20, 21, 23, 24, 26–29, 31–34, 36, 39, 40, 45).

Sanjo, Satoshi and Marie Katsurai (2017). "Recipe Popularity Prediction with Deep Visual-Semantic Fusion". In: *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM)* (cit. on p. 20).

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 (NIPS)* (cit. on p. 54).

Schroff, Florian, Dmitry Kalenichenko, and James Philbin (2015). "Facenet: A unified embedding for face recognition and clustering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 21, 23, 32).

Sermanet, Pierre, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun (2014). "Overfeat: Integrated recognition, localization and detection using convolutional networks". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on p. 22).

Shah, Meet, Xinlei Chen, Marcus Rohrbach, and Devi Parikh (2019). "Cycle-consistency for robust visual question answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 104).

Shi, Jiaxin, Hanwang Zhang, and Juanzi Li (2019). "Explainable and explicit visual reasoning over scene graphs". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 104).

Shi, Yang, Tommaso Furlanello, Sheng Zha, and Animashree Anandkumar (2018). "Question Type Guided Attention in Visual Question Answering". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on p. 70).

Shih, Kevin J., Saurabh Singh, and Derek Hoiem (2016). "Where To Look: Focus Regions for Visual Question Answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 53).

Shrestha, Robik, Kushal Kafle, and Christopher Kanan (2019). "Answer Them All! Toward Universal Visual Question Answering Models". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 90).

Simonyan, Karen and Andrew Zisserman (2015). "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on pp. 11, 22, 52).

Socher, Richard and Li Fei-Fei (2010). "Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 24).

Socher, Richard, Milind Ganjoo, Christopher D Manning, and Andrew Ng (2013). "Zero-shot learning through cross-modal transfer". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on p. 24).

Socher, Richard, Andrej Karpathy, Quoc V. Le, Christopher D. Manning, and Andrew Y. Ng (2014). "Grounded Compositional Semantics for Finding and Describing Images with Sentences". In: *Transactions of the Association for Computational Linguistics (TACL)* 2, pp. 207–218 (cit. on p. 25).

Stock, Pierre and Moustapha Cisse (2018). "ConvNets and ImageNet Beyond Accuracy: Understanding Mistakes and Uncovering Biases". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on p. 82).

Sutskever, I, O Vinyals, and QV Le (2014). "Sequence to sequence learning with neural networks". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on p. 3).

Sutskever, Ilya, James Martens, George Dahl, and Geoffrey Hinton (2013). "On the importance of initialization and momentum in deep learning". In: *International Conference on Machine Learning (ICML)* (cit. on pp. 9, 10).

Tan, Hao and Mohit Bansal (2019). "Lxmert: Learning cross-modality encoder representations from transformers". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (cit. on p. 103).

Teney, Damien, Peter Anderson, Xiaodong He, and Anton van den Hengel (2018). "Tips and Tricks for Visual Question Answering: Learnings From the 2017 Challenge". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 67, 68).

Thomason, Jesse, Daniel Gordon, and Yonatan Bisk (2019). "Shifting the Baseline: Single Modality Performance on Visual Navigation & QA". In: *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)* (cit. on p. 82).

Tieleman, T. and Geoffrey E. Hinton (2012). *Divide the gradient by a running average of its recent magnitude*. URL: `https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf` (cit. on p. 10).

Torralba, Antonio and Alexei A. Efros (2011). "Unbiased look at dataset bias". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 82).

Tucker, Ledyard R. (1966). "Some mathematical notes on three-mode factor analysis". In: *Psychometrika* 31.3, pp. 279–311 (cit. on pp. 55, 58).

Vapnik, Vladimir N (1999). "An overview of statistical learning theory". In: *IEEE transactions on neural networks* 10.5, pp. 988–999 (cit. on p. 5).

Vapnik, Vladimir N and A Ya Chervonenkis (1972). "On the uniform convergence of relative frequencies of events to their probabilities". In: *Measures of complexity*. Springer, pp. 11–30 (cit. on p. 5).

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on pp. 3, 22, 103).

Wang, Hao, Doyen Sahoo, Chenghao Liu, Ee-peng Lim, and Steven CH Hoi (2019). "Learning Cross-Modal Embeddings with Adversarial Networks for Cooking Recipes and Food Images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 104).

Wang, Kaiye, Qiyue Yin, Wei Wang, Shu Wu, and Liang Wang (2016). "A comprehensive survey on cross-modal retrieval". In: *arXiv preprint arXiv:1607.06215* (cit. on p. 4).

Wang, Liwei, Yin Li, Jing Huang, and Svetlana Lazebnik (2018). "Learning two-branch neural networks for image-text matching tasks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 41.2, pp. 394–407 (cit. on p. 25).

Wang, Liwei, Yin Li, and Svetlana Lazebnik (2016). "Learning deep structure-preserving image-text embeddings". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 22).

Wang, Xin, D. Kumar, N. Thome, M. Cord, and F. Precioso (2015). "Recipe recognition with large multimodal food dataset". In: *IEEE International Conference on Multimedia  Expo Workshops* (cit. on p. 20).

Weinberger, Kilian Q. and Lawrence K. Saul (2009). "Distance Metric Learning for Large Margin Nearest Neighbor Classification". In: *J. Mach. Learn. Res.* 10, pp. 207–244 (cit. on p. 23).

Widrow, Bernard and Marcian E Hoff (1960). *Adaptive switching circuits*. Tech. rep. Stanford Univ Ca Stanford Electronics Labs (cit. on p. 7).

Xing, Eric P, Michael I Jordan, Stuart J Russell, and Andrew Y Ng (2003). "Distance metric learning with application to clustering with side-information". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on p. 23).

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 (ICML)* (cit. on pp. 3, 53).

Yan, Fei and Krystian Mikolajczyk (2015). "Deep Correlation for Matching Images and Text". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 22).

Yang, Zichao, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola (2016). "Stacked attention networks for image question answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 53, 81, 84, 90).

Young, Peter, Alice Lai, Micah Hodosh, and Julia Hockenmaier (2014). "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions". In: *Transactions of the Association for Computational Linguistics* 2, pp. 67–78 (cit. on p. 24).

Yu Jiang*, Vivek Natarajan*, Xinlei Chen*, Marcus Rohrbach, Dhruv Batra, and Devi Parikh (2018). "Pythia v0.1: the Winning Entry to the VQA Challenge 2018". In: *arXiv preprint arXiv:1807.09956* (cit. on pp. 52, 53, 64, 69, 70, 77).

Yu, Ruichi, Ang Li, Vlad I. Morariu, and Larry S. Davis (2017). "Visual Relationship Detection With Internal and External Linguistic Knowledge Distillation". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on p. 65).

Yu, Zhou, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian (2019). "Deep Modular Co-Attention Networks for Visual Question Answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 103).

Yu, Zhou, Jun Yu, Jianping Fan, and Dacheng Tao (2017). "Multi-modal Factorized Bilinear Pooling with Co-Attention Learning for Visual Question Answering". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on pp. 53, 65, 75, 103).

Yu, Zhou, Jun Yu, Chenchao Xiang, Jianping Fan, and Dacheng Tao (2018). "Beyond Bilinear: Generalized Multi-modal Factorized High-order Pooling for Visual Question Answering". In: *IEEE Transactions on Neural Networks and Learning Systems* (cit. on pp. 53, 65, 67, 84, 103).

Zador, Anthony M (2019). "A Critique of Pure Learning: What Artificial Neural Networks can Learn from Animal Brains". In: *Nature Communications*, p. 582643 (cit. on p. 105).

Zhang, Yan, Jonathon Hare, and Adam Prügel-Bennett (2018). "Learning to Count Objects in Natural Images for Visual Question Answering". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on pp. 53, 67–69).

Zhao, Jieyu, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang (2017). "Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-level Constraints". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (cit. on p. 82).

Zhu, Bin, Chong-Wah Ngo, Jingjing Chen, and Yanbin Hao (2019). "R2GAN: Cross-Modal Recipe Retrieval With Generative Adversarial Network". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 104).

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 (CVPR)* (cit. on p. 49).