Beyond Question-Based Biases: Assessing Multimodal Shortcut Learning in Visual Question Answering

Corentin Dancette[1 *] Rémi Cadène[1,2 *†] Damien Teney[3,4] Matthieu Cord[1,5]
[1]Sorbonne Université, CNRS, LIP6, 4 place Jussieu, Paris
[2]Carney Institute for Brain Science, Brown University, USA [3]Idiap Research Institute
[4]Australian Institute for Machine Learning, University of Adelaide [5]Valeo.ai
[1]{firstname.lastname}@sorbonne-universite.fr [3] damien.teney@idiap.ch

Abstract

We introduce an evaluation methodology for visual question answering (VQA) to better diagnose cases of shortcut learning. These cases happen when a model exploits spurious statistical regularities to produce correct answers but does not actually deploy the desired behavior. There is a need to identify possible shortcuts in a dataset and assess their use before deploying a model in the real world. The research community in VQA has focused exclusively on question-based shortcuts, where a model might, for example, answer "What is the color of the sky" with "blue" by relying mostly on the question-conditional training prior and give little weight to visual evidence. We go a step further and consider multimodal shortcuts that involve both questions and images. We first identify potential shortcuts in the popular VQA v2 training set by mining trivial predictive rules such as co-occurrences of words and visual elements. We then introduce VQA-CounterExamples (VQA-CE), an evaluation protocol based on our subset of CounterExamples i.e. image-question-answer triplets where our rules lead to incorrect answers. We use this new evaluation in a large-scale study of existing approaches for VQA. We demonstrate that even state-of-the-art models perform poorly and that existing techniques to reduce biases are largely ineffective in this context. Our findings suggest that past work on question-based biases in VQA has only addressed one facet of a complex issue. The code for our method is available at https://github.com/cdancette/detect-shortcuts

1. Introduction

Visual Question Answering (VQA) is a popular task that aims at developing models able to answer free-form questions about the contents of given images. The research community introduced several datasets [5, 23, 26, 27] to study various topics such as multimodal fusion [7] and visual reasoning [4, 22]. The popular VQA v2 dataset [21] is the largest dataset of photographs of real scenes and human-provided questions. Because of strong selection biases and annotation artifacts, these datasets have served as a test-bed for the study of dataset biases and shortcut learning [18] (we will use the term "shortcut" exclusively in the rest of the paper). These spurious correlations correspond to superficial statistical patterns in the training data that allow predicting correct answers without deploying the desirable behavior. Issues of shortcut learning have become an increasing concern for other tasks in vision and natural language processing [18, 14]. In extreme cases, shortcuts in VQA may allow guessing the answer without even looking at the image [1]. Some shortcuts can be more subtle and involve both textual and visual elements. For instance, training questions containing *What sport* are strongly associated with the answer *tennis* when they co-occur with a racket in the image (see Figure 1). However, some examples can be found in the validation set, such as *What sport field is in the background ?*, that lead to a different answer (*soccer*) despite a racquet being present in the image. Because of such exceptions, a model that strongly relies on simple co-occurrences will fail on unusual questions and scenes. Our work studies such multimodal patterns and their impact on VQA models.

The presence of dataset biases in VQA datasets is well known [1, 21, 23, 29], but **existing evaluation protocols are limited to text-based shortcuts**. Our work introduces VQA-*CounterExamples* (VQA-CE for short) which is an evaluation protocol for multimodal shortcuts. It is easy to reproduce and can be used on any model trained on VQA v2, without requiring retraining. We first start with a method to discover superficial statistical patterns in a given VQA dataset that could be the cause of shortcut learning. We discover a collection of co-occurrences of textual and visual elements that are strongly predictive of certain an-

*Equal contribution †Work done before April 2021 and joining Tesla

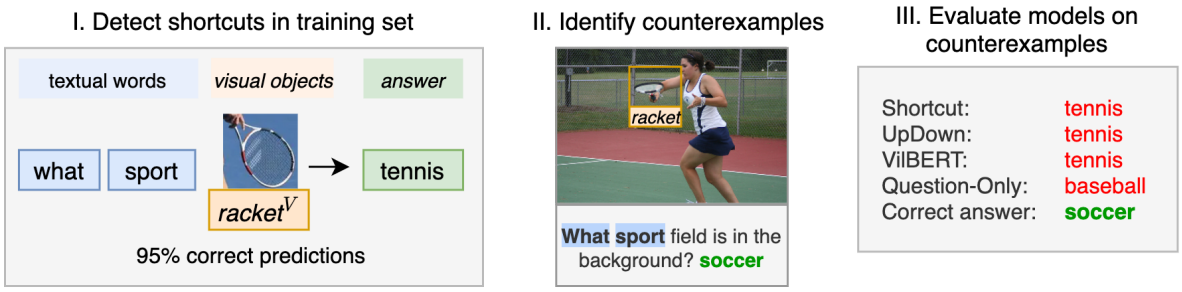

Figure 1. Overview of this work. We first mine simple predictive rules in the training data such as: what + sport + racket^V $\rightarrow$ **tennis**. We then search for counterexamples in the validation set that identify some rules as undesirable statistical shortcuts. Finally, we use the counterexamples as a new challenging test set and evaluate existing VQA models like UpDown [3] and VilBERT [31].

swers in the training data and often transfer to the validation set. For instance, we discover a rule that relies on the appearance of the words "what","they","playing" together with the object "controller" in the image to always predict the correct answer "wii". We consider this rule to be a shortcut since it could fail on arbitrary images with other controllers, as it happens in the real world. Thus, our method can be used to reflect biases of the datasets that can potentially be learned by VQA models.

We go one step further and identify counterexamples in the validation set where the shortcuts produce an incorrect answer. These counterexamples form a new challenging evaluation set for our VQA-CE evaluation protocol. We found that the accuracy of existing VQA models is significantly degraded on this data. More importantly, we found that most current approaches for reducing biases and shortcuts are ineffective in this context. They often reduce the average accuracy over the full evaluation set without significant improvement on our set of counterexamples. Finally, we identify shortcuts that VQA models may be exploiting. We find several shortcuts giving predictions highly correlated with existing models' predictions. When they lead to incorrect answers on some examples from the validation set, VQA models also provide incorrect answers. This tends to show that VQA models exploit these multimodal shortcuts. In summary, the contributions of this paper are as follows.

1. We propose **a method to discover shortcuts** which rely on the appearance of words in the question and visual elements in the image to predict the correct answer. By applying it to the widely-used VQA v2 training set, we found a high number of multimodal shortcuts that are predictive on the validation set.
2. We introduce **the VQA-CE evaluation protocol** to assess the VQA models' reliance on these shortcuts. By running a large-scale evaluation of recent VQA approaches, we found that state-of-the-art models exploit these shortcuts and that bias-reduction methods are ineffective in this context.

2. Related Work

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

Detecting cases of shortcut learning A first type of approaches consists in detecting the use of shortcuts by leveraging explainability methods [16, 36, 40, 32]. However, they require costly human interpretation, or additional annotations [15] to assess whether a particular explanation reveals the use of a shortcut. A second type consists in evaluating a model on artificial data or out-of-distribution data. For instance, [19] artificially modify the texture of natural images to show that convolutional networks trained on ImageNet exploit features related to textures instead of shapes. Also, [2] and [6] evaluate vision models on out-of-distribution data to show that they cannot identify known objects when their poses changed significantly. In this line of works, [24] focus on evaluating models on adversarial examples and show links with statistical regularities or "non-robust features" that models exploit. A third type of approaches use specific models with a known type of biases to assess the amount of biases of this type directly in the dataset. For instance, in computer vision, BagNet [9] obtained high accuracy on ImageNet by using occurrences of small local image features, without using the global spatial context. This suggests that state-of-the-art ImageNet models are biased towards local image features. Similarly, our approach leverages specific shallow models that are constructed to only exploit biases of a certain type.

This kind of approaches have been used in VQA. Previous works [1, 5, 21] used question-only and image-only models to quantify the amount of unimodal shortcuts in a dataset. Instead, our approach is not only able to quantify the amount of shortcuts but also identify these shortcuts. More importantly, our method can identify multimodal shortcuts that combine elements of the question and the image. The closest approach to ours [32] uses the Apriori algorithm to extract predictive rules that combine the appearance of words and visual contents. However, these

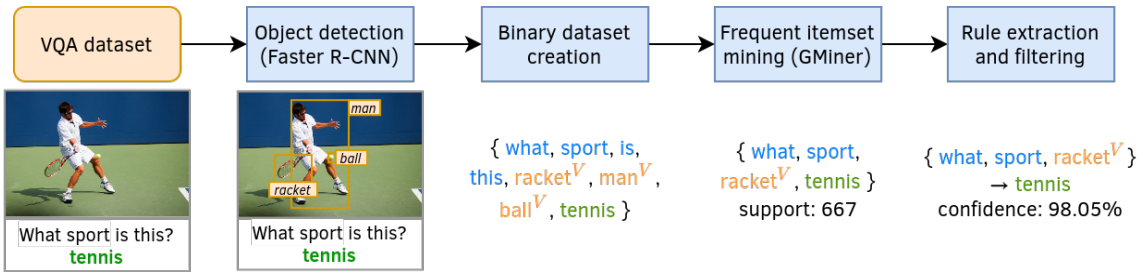

Figure 2. Pipeline of the proposed method to detect potential shortcuts in a VQA training set. We detect and label objects in images with a Faster R-CNN model. We then summarize each VQA example with binary indicators representing words in the question, answer, and labels of detected objects. Finally, a rule mining algorithm identifies frequent co-occurrences and extracts a set of simple predictive rules.

rules are specific to the attention maps and predictions of the VQA model from [28]. More problematically, they are extracted on the validation set and are mainly used for qualitative purposes. Our approach also relies on the Apriori algorithm but extracts rules directly on the training set, independently of any model, and the predictive capacity of the rules is evaluated on the validation set.

Evaluating VQA models' reliance on shortcuts Once a class of shortcuts has been identified, a first way to evaluate model's robustness is to build external out-of-distribution evaluation datasets on which using these shortcuts leads to a wrong prediction. In Visual Question Answering, the VQA-Rephrasing [37] dataset contains multiple rephrased but semantically-identical questions. The goal is to test model's sensitivity to small linguistic variations and will penalize usage of a certain class of question-related shortcuts. Similar datasets exist for natural language processing [25, 33].

Another type of evaluation methods artificially injects certain kind of shortcuts in the training set and evaluate models on examples that do not possess these shortcuts. The widely used **VQA-CP** [1] evaluation procedure consists in resplitting the original VQA datasets so that the distribution of answers per question type ("how many", "what color is", etc.) is different between the training and evaluation set. Models that rely on those artificial shortcuts are therefore penalized. VQA-CP was used to develop methods that aim at avoiding learning shortcuts from the question type on this modified training set [10, 13, 17, 35, 20, 11, 17, 39, 41, 42, 43]. Similar approaches for VQA exists [13]. The downside of these approaches is that they focus on artificially introduced shortcuts and only target text-related biases and shortcuts. More importantly, models that have been trained on original datasets, i.e. VQA v2, need to be retrained on their modified versions, i.e. VQA-CP v2. Other concerns have been raised in [43]. On the contrary, our proposed evaluation method does not require additional data collection or data generation, focuses on multimodal shortcuts, and does not require retraining. We follow guidelines from [14, 43] for a better evaluation of the use of shortcuts.

Finally, the GQA-OOD [29] dataset extracts from the GQA[23] validation and testing set example with rare answers, conditioned on the type of question. Thus, it targets question-related shortcuts. It enables the evaluation of models without retraining on a separate training set.

3. Detecting multimodal shortcuts for VQA

3.1. Our shortcut detection method

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

Mining predictive rules on a training set Our goal is to detect shortcuts that f might use to provide an answer without deploying the desired behavior. To this end, we limit ourselves to a class of shortcuts that we think is often leveraged by f. We display in Figure 2 our rule mining process. These shortcuts are short predictive association rules $A \rightarrow C$ that associate an **antecedent** A to a **consequent** C. Our antecedents are composed of words of the question and salient objects in the image (or image patch), while our consequents are just answers. For instance, the rule {what, color, plant} $\rightarrow$ {**green**} provides the answer "green" when the question contains the words "what", "color" and "plant". These shallow rules are by construction shortcuts. They are predictive on the validation set but do not reflect the complex behavior that needs to be learned to solve the VQA task. For instance, they do not rely on the order of words, neither the position and relationships of visual contents in the image. They lack the context that is required to properly answer the question. Moreover,

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

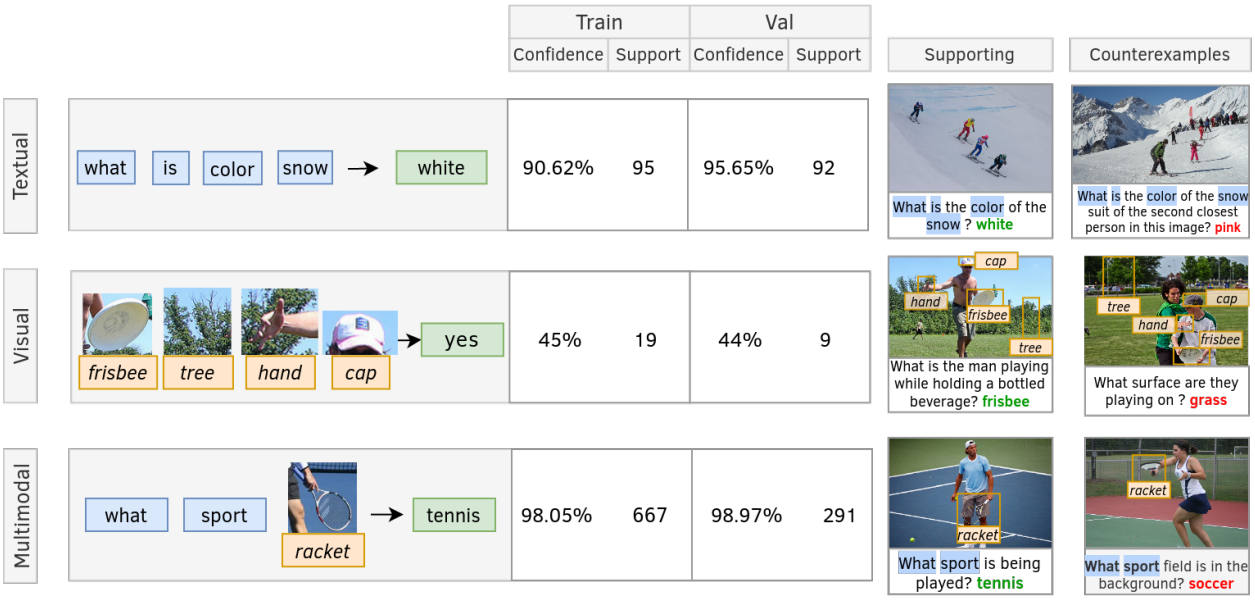

Figure 3. Examples of shortcuts found in the VQA v2 dataset. The confidence is the accuracy obtained by applying the shortcut on all examples matching by its *antecedent*. The support is the number of matching examples. More supporting examples and counterexamples are shown in the supplementary material.

even rules that seem correct often have a few counterexamples in the dataset, i.e. examples that are matched by the antecedent but the consequent provides the wrong answer. We later use these counterexamples in our evaluation procedure.

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

Frequent itemset mining On our binary dataset, we apply the GMiner algorithm [12] to efficiently find frequent *itemsets*. An itemset is a set of tokens $\mathcal{I} = \{i_1, .., i_n\}$ that appear very frequently together in the dataset. The **support** of the itemset is its number of occurrences. For example, the itemset {what, color, plant, **green**} might be very common in the dataset and have a high support. GMiner takes one parameter, the minimum support. We include an additional parameter, which is the maximum length for an itemset. We detail how we select parameters at the end of this section.

Rules extraction and filtering The next step is to extract rules from the frequent itemsets. First, we filter out the itemsets that do not contain an answer token, as they cannot be converted to rules. For the others that do contain an answer a, we remove it from the itemset to create the antecedent $\mathcal{X}$ ($\mathcal{X} = \mathcal{I} \setminus a$). The rule is then $\mathcal{X} \Rightarrow a$. The **support** s of the rule is the number of occurrences of $\mathcal{X}$ in the dataset. The **confidence** c of the rule is the frequency of correct answers among examples that have $\mathcal{X}$.
We then proceed to filter rules. We apply the following three steps: **(a)** we remove the rules with a confidence on the training set lower than 30% (remove when $c < 0.3$) **(b)** if some rules have the same antecedent but different answers, then we keep the rule with the highest confidence and remove the others. For instance, given the rules {is, there} $\Rightarrow$ **yes** and {is, there} $\Rightarrow$ **no** with a respective confidence of 70% and 30%, we only keep the first one with the answer **yes**. **(c)** if a rules r_1's antecedent is a superset of another rule r_2's antecedent, if both have the same answer, and r_1 has a lower confidence than r_2, then we remove r_1. For instance, given the rules {is, there} $\Rightarrow$ **yes** and {is, there, cat} $\Rightarrow$ **yes** with a respective confidence of 70% and 60%, we only keep the first one without the word *cat*. We consider the remaining rules as shortcuts. Note that rules with a confidence of 100% could be considered *correct* and not shortcuts, but these rules will not influence our evaluation protocol, detailed in Section 4.

3.2. Analysis of shortcuts on natural data

We analyze the shortcuts that our approach can detect on the widely used VQA v2 dataset [21] made of 1.1M image-question-answer examples and based on 200K images from the MS-COCO dataset [30]. We extract shortcuts with dif-

ferent combinations of minimum support and confidence. Each time, we aggregate them into a classifier that we evaluate on the validation set. We detail how to build this kind of classifier in Section 4.2. We select the support and confidence leading to the best overall accuracy. It corresponds to a minimum support of $2.1 \cdot 10^{-5}$ (about $\sim$8 examples in training set), and a minimum confidence of 0.3. Once these shortcuts have been detected, we assess their number and type(purely textual, purely visual, or multimodal). We also verify that they can be used to find counterexamples that cannot be accurately answered using shortcuts. Finally, we evaluate their confidence on the validation set. In the next section, we leverage these counterexamples with our VQA-CE evaluation protocol to assess model's reliance on shortcuts.

Words-only and objects-only shortcuts First, we show that our approach is able to detect shortcuts that are purely textual or visual. In the first row of Figure 3, we display a shortcut detected on VQA v2 that only accounts for the appearance of words in the question. It predicts the answer "white" when the words "what", "color", "is", "snow" appear at any position in the question. In the training set, these words appear in 95 examples and 90.62% of them have the "white" answer. This shortcut is highly predictive on the validation set and gets 95.65% of correct answers over 92 examples. We also display an example on which exploiting the shortcut leads to the correct answer, and a counterexample on which the shortcut fails because the question was about "the color of the snow suit" which is "pink". In the second row, we show a shortcut that only accounts for the appearance of visual objects. It predicts "yes" when a "frisbee", a "tree", a "hand" and a "cap" appear in the image. However, this kind of shortcuts is usually less predictive since they cannot exploit the question-type information which is highly correlated with certain answers, i.e. "what color" is usually answered by a color.

Multimodal shortcuts Then, we show that our approach is able to detect multimodal shortcuts. They account for the appearance of both words **and** visual objects^V. In the third row of Figure 3, we display a multimodal shortcut that predicts "tennis" when the words what, sport and a racket^V appear. It is a common shortcut with a confidence of 98.05% based on a support of 667 examples in the training set. It is also highly predictive on the validation set with 98.97% confidence and 291 support. At first sight, it is counter-intuitive that this simple rule is a shortcut but answering complex questions is not about detecting frequent words and objects in images that correlate with an answer. In fact, this shortcut is associated to counterexamples where it fails to answer accurately. Here, the sport that can be played in the background is not tennis but soccer.

Number of shortcuts and statistics per type Here we show that our approach can be used to detect a high number of multimodal shortcuts. Overall, it detects ~1.12M shortcuts on the VQA v2 training set. As illustrated in Figure 4, since there are ~413K examples, it is often the case that several shortcuts can be applied to the same example. This is the main reason behind the high number of shortcuts For instance, the antecedent {animals, what, giraffe^V} overlaps with {animals, these, what, giraffe^V}. Among all the shortcuts that our method can detect, only ~50k are textual, ~77k are visual and ~1M are multimodal. In other words, ~90% are multimodal. In addition to being more numerous, they are also more predictive. For instance, the most confident shortcut that matches an example, highlighted in green in Figure 4, is multimodal 91.80% of the time. Finally, ~3K examples are not matched by any shortcut antecedents. They have unusual question words or visual content. We later do not take them into account in our VQA-CE evaluation protocol. We display some representative examples in the supplementary materials.

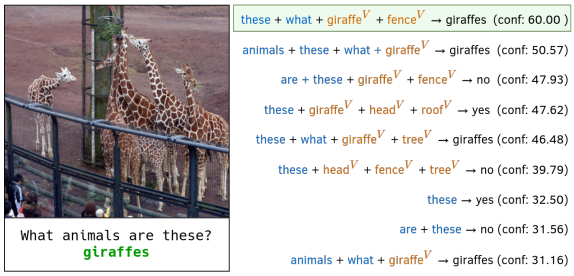

Figure 4. Multiple shortcuts can often be exploited to find the correct answer in any given example. The confidence is the percentage of accurate answers among examples that are matched by the shortcut *antecedent*. The shortcut of highest confidence (in green) is multimodal for ~92% of examples.

Confidence distribution on training and unseen data In the supplementary materials, we display the confidence distribution of these shortcuts. We observe that our shortcuts are predictive on unseen data that follows the training set distribution. The number of shortcuts that reach a confidence between 0.9 and 1.0 is as high on the validation set as it is than on the training set. This means that shortcuts detected on the VQA v2 training set transfer to the validation set. Additionally, most shortcuts obtain a confidence lower than 1.0, which allows finding examples that contradict them by leading to the wrong answers. These counterexamples are the core of our approach to assess the VQA model's reliance on shortcuts which is described next.

4. Assessing models' reliance on shortcuts

The classic evaluation protocol in VQA consists in calculating the average accuracy over all the examples. Instead, we introduce the VQA-*CounterExamples* evaluation

protocol (VQA-CE) that additionally calculates the average accuracy over a specific subset of the validation set. This subset is made of counterexamples that cannot be answered by exploiting shortcuts. Models that do exploit shortcuts are expected to get a lower accuracy. It is how we assess the use of shortcuts. Importantly, our protocol does not require retraining as it was the case with the previous VQA-CP [1] protocol. We first detail the subsets creation procedure at the core of our VQA-CE protocol. Then we run extensive experiments to assess the use of shortcuts on many VQA models and bias-reduction methods. Finally, we identify shortcuts that are often exploited by VQA models.

4.1. Our VQA-CE evaluation protocol

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

Distribution of examples Here, we show how the split between our two subsets Counterexamples and Easy affects the distribution of examples. In Figure 5, we show that the original distribution of answers is similar to the Easy distribution but dissimilar to the Counterexamples distribution. Highlighted in blue, we display the five most common answers from the Easy distribution. They can be found at the same positions in the original distribution, the two major answers being "yes" and "no". It is not the case in the Counterexamples subset where these answers appear less frequently. Nonetheless, they are still in the top 30 answers which shows that our subsets creation is not a trivial splitting between frequent and rare answers. Similarly, the five most common answers from the Counterexamples subset, highlighted in orange, can be found in the Easy and All subset. We report similar observations for the questions and answer-type distributions in the supplementary materials.

4.2. Main results

In Table 1, we report results of some baselines, common VQA models, and latest bias-reduction methods following our VQA-CE evaluation protocol. Models that exploit shortcuts are expected to get a lower accuracy on the Counterexamples compared to their overall accuracy. All models have been trained on the VQA v2 training set, and evaluated on the VQA v2 validation set. We detail them and discuss our findings in the next paragraphs. We report additional results on VQA v1 and v2 in the supplementary materials.

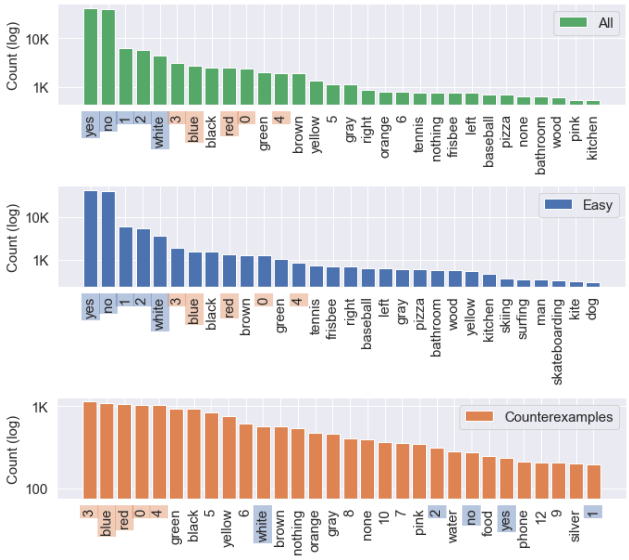

Figure 5. Number of examples per answer (30 most frequent ones) in the complete validation set, our Counterexamples subset, and our Easy subset. Answers highlighted in blue and orange are the top 5 answers for the Easy and Counterexamples subsets respectively.

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

Aggregating shortcuts to create a classifier In order to evaluate our shortcuts as a whole, we aggregate them to build a VQA classifier. As shown in the preceding section, each training example is associated with shortcuts that can be used to find the correct answer. Among these correct shortcuts, we select the highest-confidence one for each example. This leaves us with 115,718 unique shortcuts. In order to predict an answer for an unseen example, we take the most predicted answer for all its matching shortcuts weighted by the confidence of the shortcuts. For the examples that are not matched by any shortcut, we output "yes", the most common answer. Our shortcut-based clas-

	Approaches	Overall	**Counterexamples (ours)**	**Easy (ours)**	VQA-CP v2 [1]
	Number of examples	*214,354*	*63,298*	*147,681*	
Baselines	Shortcuts	42.26	0.00	61.13	22.64
	Image-Only	23.70	1.58	33.58	19.31
	Question-Only	44.12	11.59	58.61	15.95
VQA models	SAN [44] – *grid features*	55.61	26.64	68.45	24.96
	UpDown [3]	63.52 (+0.00)	33.91 (+0.00)	76.69 (+0.00)	39.74
	BLOCK [8]	63.89	32.91	77.65	38.69
	VilBERT [31] – *pretrained*†	67.77	39.24	80.50	—
Bias-reduction methods	*UpDown [3] is used as a base architecture for bias-reduction methods*				
	RUBi [10]	61.88 (-1,64)	32.25 (-1,66)	75.03 (-1.66)	44.23
	LMH + RMFE [17]	60.96 (-2.56)	33.14 (-0.77)	73.32 (-3.37)	54.55
	ESR [39]	62.96 (-0.56)	33.26 (-0.65)	76.18 (-0.51)	48.50
	LMH [13]	61.15 (-2.37)	34.26 (+0.35)	73.12 (-3.57)	52.05
	LfF [34]	63.57 (+0.05)	34.27 (+0.36)	76.60 (-0.09)	39.49
	LMH+CSS [11]	53.55 (-9.97)	34.36 (+0.45)	62.08 (-14.61)	58.95
	RandImg [43]	63.34 (-0.18)	34.41 (+0.50)	76.21 (-0.48)	55.37

Table 1. Results of our VQA-CE evaluation protocol. We report accuracies on VQA v2 full validation set and on our two subsets: **Counterexamples** and **Easy** examples. We re-implemented all models and bias-reduction methods. †VilBERT is pretrained on Conceptual Caption and fine-tuned on VQA v2 training set. Scores in (green) and (red) are relative to UpDown [3]. We also report accuracies on VQA-CP v2 [1] which focus on question biases, and comes with a different training set and testing set. VilBERT was not evaluated for VQA-CP as it was pretrained on balanced datasets.

sifier reaches an overall accuracy of 42.26%, close to the 44.12% of the deep question-only baseline. Interestingly, both use a different class of shortcuts. Ours is mostly based on shallow multimodal shortcuts, not just shortcuts from the question. Since we use the same shortcuts to create our subsets, the shortcut-based classifier reaches a score of 0% on the Counterexamples. On VQA-CP testing set, our classifier reaches 22.44% accuracy. It highlights the difference with our counterexamples subset: VQA-CP does penalize some shortcuts, but there are still some that can be exploited.

VQA models learn shortcuts We compare different types of VQA models: SAN [44] represents the image as a grid of smaller patches and uses a stacked attention mechanism over these patches, instead UpDown [3] represents the image as a set of objects detected with Faster-RCNN and uses a simpler attention mechanism over them, BLOCK [8] also relies on the object representations but uses a more complex attention mechanism based on a bilinear fusion, VilBERT [31] also relies on the object representations but uses a transformer-based model that has been pretrained on the Conceptual Caption dataset [38]. First, they suffer from a loss of ~29 accuracy points on the counterexamples compared to their overall accuracy. This suggests that, despite their differences in modeling, they all exploit shortcuts. Note that comparable losses are reported on VQA-CP v2 [1] which especially focuses on shortcuts based on question-types. Second, our evaluation protocol can be used to compare two models that get similar overall accuracies: UpDown and BLOCK which gets +0.37 points over UpDown. We can explain that this gain is due to a superior accuracy on the Easy subset with +0.96 and report a loss of -1.00 points on the Counterexamples. These results suggest that the bilinear fusion of BLOCK better captures shortcuts. On the contrary, VilBERT gets a better accuracy on our both subsets. It might be explained by the advantages of pretraining on external data.

Bias-reduction methods do not work well on natural multimodal shortcuts Our evaluation protocol can also be used to assess the efficiency of common bias-reduction methods. We use publicly available codebases when available, or our own implementation. All methods have been developed on the VQA-CP v2 dataset. It introduces new training and evaluation splits of VQA v2 that follow different distributions conditioned on the question-type. All the studied methods have been applied to UpDown and reached gains ranging from +5 to +20 accuracy points on the VQA-CP testing set. We evaluate them in the more realistic context of the original VQA v2 dataset. We show that their effect on our Counterexamples subset is very small. More specifically, some methods such as RUBi [10], LMH+RMFE [17], and ESR [39] have a negative effect on all subsets. Some methods such as LMH [13] and LMH+CSS [11] slightly improve the accuracy on counterexamples but strongly decrease the accuracy on the Easy subset, and consequently decrease the overall accuracy. As reported in [43], most of those methods rely on knowledge about the VQA-CP testing distribution (inversion of the an-

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

Table 2. Instances of shortcuts that are highly correlated with VQA models' predictions. We display their antecedent made of words from the question and objects^V from the image, and their **answer**. Their support, i.e. number of examples matched by the antecedent, and confidence, i.e. percentage of correct answers among them, have been calculated on the VQA v2 training and validation sets. We report the correlation coefficients of their predictions with those of three VQA models: UpDown [3] that uses an object detector, VilBERT [31] that has been pretrained on a large dataset, and Q-only [21] that only uses the question. We show some counterexamples in the supplementary material.

swer distribution conditioned on the question), which no longer applies in our VQA v2 evaluation setting. Finally, we found two methods, LfF [34] and RandImg [43] that slightly improve the accuracy on the Counterexamples subset with gains of +0.36 and +0.50, while having a very small impact on the overall accuracy, even reaching small gains in the best case of LfF. It should be noted that LfF is more general than others since it was not designed for the VQA-CP context. Overall, all effects are much smaller compared to their effectiveness on VQA-CP. This suggests that those bias-reduction methods might exploit the distribution shift between VQA-CP training and evaluation splits. They are efficient in this setting but do not work as well to reduce *naturally-occurring* shortcuts in VQA.

4.3. Identifying most exploited shortcuts

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

In Table 2, we report shortcuts that obtain the highest correlation coefficient with UpDown [3] and VilBERT [31]. Overall, these shortcuts have a high confidence and support, which means that they are common in the dataset and predictive. Most importantly, they are multimodal. As a consequence, these shortcuts obtain low correlations with Question-Only [21]. On the contrary, they obtain a 100% correlation coefficient with VilBERT and UpDown. For instance, the second shortcut provides the answer **skateboarding** for the appearance of sport, this, what in the question and a skateboard^V in the image. It is a common shortcut with a support of 31 examples in the validation set. It gets a correlation of 0% because Question-Only mostly answer baseball for these examples. Its confidence of 87.1% indicates that 4 counterexamples can be found where the shortcut provides the wrong answer. To be correctly answered, they require more than a simple prediction based on the appearance of words and salient visual contents. These results once again confirm that VQA models tend to exploit multimodal shortcuts. It shows the importance of taking them into account in an evaluation protocol for VQA.

5. Conclusion

We introduced a method that discovers multimodal shortcuts in VQA datasets. It gives novel insights on the nature of shortcuts in VQA: they are not only related to the question but are also multimodal. We introduced an evaluation protocol to assess whether a given models exploits multimodal shortcuts. We found that most state-of-the-art VQA models suffer from a significant loss of accuracy in this setting. We also evaluated existing bias-reduction methods. Even the most general-purpose of these methods do not significantly reduce the use of multimodal shortcuts. We hope this new evaluation protocol will stimulate the design of better techniques to learn robust VQA models.

6. Acknowledgements

The effort from Sorbonne Université was partly supported by ANR grant VISADEEP (ANR-20-CHIA-0022). This work was granted access to HPC resources of IDRIS under the allocation 2020-AD011011588 by GENCI.

References

[1] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 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)*, 2018. 1, 2, 3, 6, 7

[2] Michael A Alcorn, Qi Li, Zhitao Gong, Chengfei Wang, Long Mai, Wei-Shinn Ku, and Anh Nguyen. Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[3] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 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)*, 2018. 2, 7, 8

[4] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[5] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. 1, 2

[6] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2

[7] Hedi Ben-Younes, Remi Cadene, Nicolas Thome, and Matthieu Cord. Mutan: Multimodal tucker fusion for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 1

[8] Hedi Ben-Younes, Remi Cadene, Nicolas Thome, and Matthieu Cord. Block: Bilinear superdiagonal fusion for visual question answering and visual relationship detection. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2019. 7

[9] W. Brendel and M. Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 2

[10] Remi Cadene, Corentin Dancette, Hedi Ben-Younes, Matthieu Cord, and Devi Parikh. RUBi: Reducing Unimodal Biases for Visual Question Answering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 3, 7

[11] Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. Counterfactual samples synthesizing for robust visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3, 7

[12] Kang-Wook Chon, Sang-Hyun Hwang, and Min-Soo Kim. Gminer: A fast gpu-based frequent itemset mining method for large-scale data. *Information Sciences*, 439:19–38, 2018. 4

[13] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4060–4073, 2019. 3, 7

[14] Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint arXiv:2011.03395*, 2020. 1, 3

[15] Abhishek Das, Harsh Agrawal, Larry Zitnick, Devi Parikh, and Dhruv Batra. Human attention in visual question answering: Do humans and deep networks look at the same regions? *Computer Vision and Image Understanding*, 163:90–100, 2017. 2

[16] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[17] Itai Gat, Idan Schwartz, Alexander Schwing, and Tamir Hazan. Removing bias in multi-modal classifiers: Regularization by maximizing functional entropies. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3, 7

[18] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *arXiv preprint arXiv:2004.07780*, 2020. 1

[19] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 2

[20] Tejas Gokhale, Pratyay Banerjee, Chitta Baral, and Yezhou Yang. Mutant: A training paradigm for out-of-distribution generalization in visual question answering. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. 3

[21] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 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)*, 2017. 1, 2, 4, 8

[22] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[23] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional

question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3

[24] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2

[25] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017. 3

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

[27] Kushal Kafle and Christopher Kanan. An analysis of visual question answering algorithms. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 1

[28] Vahid Kazemi and Ali Elqursh. Show, ask, attend, and answer: A strong baseline for visual question answering. *arXiv preprint arXiv:1704.03162*, 2017. 3

[29] Corentin Kervadec, Grigory Antipov, Moez Baccouche, and Christian Wolf. Roses are red, violets are blue... but should vqa expect them to? *arXiv preprint arXiv:2006.05121*, 2020. 1, 3

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014. 4

[31] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2, 7, 8

[32] Varun Manjunatha, Nirat Saini, and Larry S. Davis. Explicit Bias Discovery in Visual Question Answering Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[33] Tom McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics. 3

[34] Junhyun Nam, Hyuntak Cha, Sung-Soo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems*, 33, 2020. 7, 8

[35] Sainandan Ramakrishnan, Aishwarya Agrawal, and Stefan Lee. Overcoming language priors in visual question answering with adversarial regularization. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 3

[36] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016. 2

[37] Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. Cycle-consistency for robust visual question answering. In *2019 Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019. 3

[38] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL Short Papers)*, 2018. 7

[39] Robik Shrestha, Kushal Kafle, and Christopher Kanan. A negative case analysis of visual grounding methods for vqa. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8172–8181, 2020. 3, 7

[40] Pierre Stock and Moustapha Cisse. Convnets and imagenet beyond accuracy: Understanding mistakes and uncovering biases. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, 2018. 2

[41] Damien Teney, Ehsan Abbasnejad, and Anton van den Hengel. Unshuffling data for improved generalization. *arXiv preprint arXiv:2002.11894*, 2020. 3

[42] Damien Teney and Anton van den Hengel. Actively seeking and learning from live data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1940–1949, 2019. 3

[43] Damien Teney, Kushal Kafle, Robik Shrestha, Ehsan Abbasnejad, Christopher Kanan, and Anton van den Hengel. On the value of out-of-distribution testing: An example of goodhart's law. *arXiv preprint arXiv:2005.09241*, 2020. 3, 7, 8

[44] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 7