VisQA: X-raying Vision and Language Reasoning in Transformers

Theo Jaunet, Corentin Kervadec, Romain Vuillemot, Grigory Antipov, Moez Baccouche, Christian Wolf

VisQA: X-raying Vision and Language Reasoning in Transformers

Théo Jaunet*, Corentin Kervadec*, Romain Vuillemot, Grigory Antipov, Moez Baccouche, and Christian Wolf

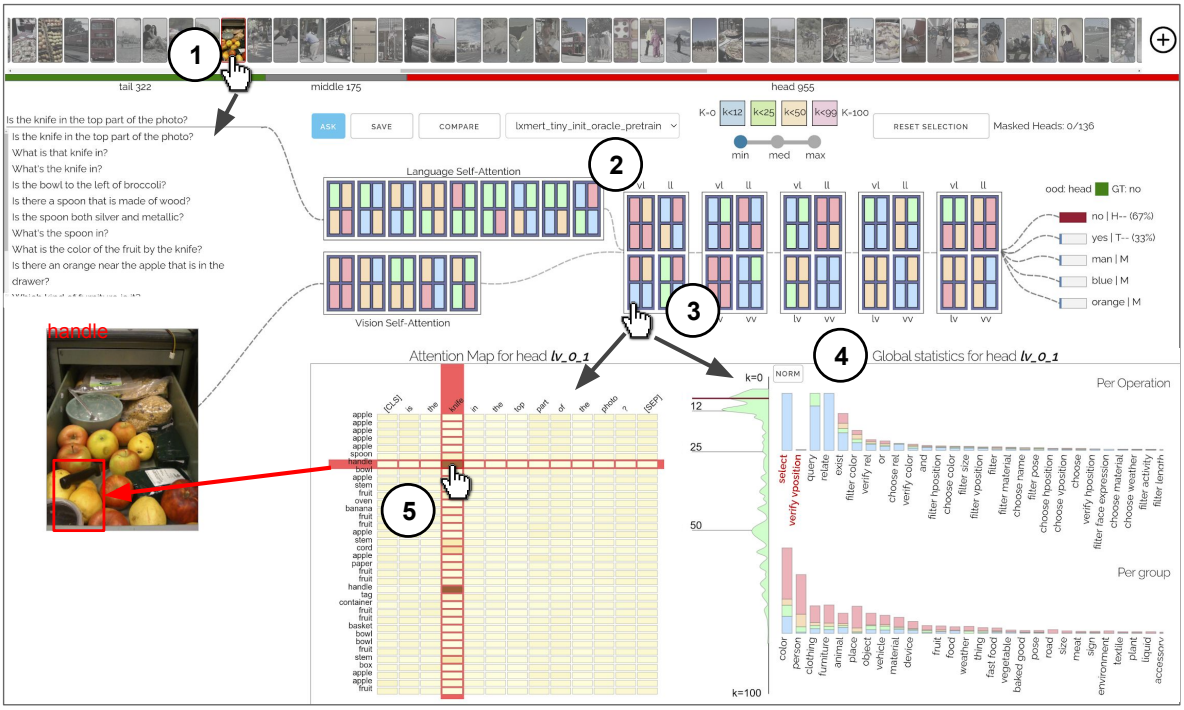

Fig. 1. Opening the black box of neural models for vision and language reasoning: given an open-ended question and an image ①, VISQA enables to investigate whether a trained model resorts to reasoning or to bias exploitation to provide its answer. This can be achieved by exploring the behavior of a set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

Abstract— Visual Question Answering systems target answering open-ended textual questions given input images. They are a testbed for learning high-level reasoning with a primary use in HCI, for instance assistance for the visually impaired. Recent research has shown that state-of-the-art models tend to produce answers exploiting biases and shortcuts in the training data, and sometimes do not even look at the input image, instead of performing the required reasoning steps. We present VisQA, a visual analytics tool that explores this question of reasoning vs. bias exploitation. It exposes the key element of state-of-the-art neural models — attention maps in transformers. Our working hypothesis is that reasoning steps leading to model predictions are observable from attention distributions, which are particularly useful for visualization. The design process of VisQA was motivated by well-known bias examples from the fields of deep learning and vision-language reasoning and evaluated in two ways. First, as a result of a collaboration of three fields, machine learning, vision and language reasoning, and data analytics, the work lead to a better understanding of bias exploitation of neural models for VQA, which eventually resulted in an impact on its design and training through the proposition of a method for the transfer of reasoning patterns from an oracle model. Second, we also report on the design of VisQA, and a goal-oriented evaluation of VisQA targeting the analysis of a model decision process from multiple experts, providing evidence that it makes the inner workings of models accessible to users.

Index Terms—Transformers, Visual Question Answering, Visual analytics

1 INTRODUCTION

- *(*) indicates equal contribution*
- *T. Jaunet and C. Wolf are with INSA-Lyon and LIRIS.*
- *C. Kervadec is with INSA-Lyon, LIRIS, and Orange Innovation.*
- *R. Vuillemot is with École Centrale de Lyon and LIRIS.*
- *G. Antipov and M. Baccouche are with Orange Innovation.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

Visual Question Answering (VQA) systems [5] attempt to answer questions provided as input in a textual form together with a corresponding image. As an example, asking the question *"Is the knife in the top part of the photo?"* associated with the input image shown in Figure 1 should yield the answer *"No"*. Direct applications of such systems are support for the visually impaired, semi-autonomous robot navigation through language instructions, and, more generally, AI tools covering a broad spectrum of tasks guided through language input. In particular, VQA serves as a testbed for learning high-level reasoning from data, as the performance of targeted models and methods relies on advances in Computer Vision (CV), Natural Language Processing (NLP), and

Machine Learning (ML). The task deals with large varieties, and solving an instance can involve visual recognition, logic, arithmetic, spatial reasoning, intuitive physics, causality, and multi-hop reasoning. It also requires combining two modalities of different nature: images and language.

Recent VQA models are based on a powerful type of deep neural network called transformers [56]. Originally developed for NLP tasks, these models have been extensively applied to VQA [42, 54, 62] and recently even on pixel-level in pure image-based problems [17, 18, 22, 59, 65]. Transformers are conceptually simple models, which, however, can learn very complex relationships between the items of unordered sets, each of which is represented as a (learned) embedding in a vector space. Making sense of a learned neural model and verifying its inner workings is a difficult problem, which we address in this work.

In this paper, we focus on a typical and important problem arising with trained neural models, and in particular models for vision and language reasoning: as they are trained with supervision to provide correct answers, they often tend to find shortcuts in learning and learn to exploit spurious biases in training data instead of the desired reasoning a human would apply in a similar situation [2, 24, 32, 44]. To provide an example, if a model is asked *"What is the color of the banana"?* with an image of a *"banana"*, it might learn to answer *"yellow"* whatever the real color of the image is, as this is probably the correct answer for a large majority of input images — solving the correct reasoning problem is a much harder task. An exact definition of the term "correct reasoning" is difficult, we refer to [7, 32] and define it as *algebraically manipulating words and visual objects to answer a new question*. In particular, we interpret reasoning as the opposite of exploiting spurious biases in training data.

Existing work on bias reduction and the evaluation of bias origins tends to focus on statistical techniques, whose power lies in quantitative evaluation and visualizations on dataset-level, showing full or marginal distributions of inputs, features, and outputs, and resort to dimensionality reduction. While these techniques are very useful, their power is limited when we search for insights into detailed inner workings of neural models, for which an investigation per sample is more helpful. Only for a single instance, it is possible to observe the origins for lack of reasoning, which can include, aside from errors in the trained reasoning module itself, also problems in the input pipeline (the object detection module) and wrong annotations of ground truth data.

We introduce VISQA, an instance-based visual analytics tool designed to help domain experts, referred to as model builders [27], investigate how information flows in a neural model and how the model relates different items of interest to each other in vision and language reasoning. Attention maps are at the heart of transformer-based deep networks, and as such are the primary object studied by VISQA. It allows an expert to browse through image and question pairs sorted by an automatic estimate of the amount of reasoning that went into answering each sample. Once a pair is selected, users can explore the different attention maps represented as heatmaps. The exploration is guided by their position in the model, but also by color codes that convey the intensity of each head, i.e. whether they focus attention narrowly on specific items, or broadly over the full input set. Complementary dataset-wide statistics are provided for each selected attention head, either globally, or with respect to specific reasoning modes of language functions, e.g. *"What is", "Where is", "What color"* etc. While the tool is post-hoc, it is also interactive and allows certain modifications to the internal structure of the model. At any time, attention maps can be pruned to observe their impact on the output answer.

VISQA is the result of a collaboration between experts in visual analytics, and experts in Visual Question Answering systems and Machine Learning. As will be detailed in section 4, this collaboration, and data gathered using VISQA, led to improvements of the reasoning capabilities of transformer-based models by introducing new methodological contributions in machine learning and computer vision, which we also recently reported in a different associated publication [33]. The usability of VISQA has been evaluated by different experts in deep learning, who were not involved in the project nor its design. We report experiments with qualitative interviews and results in section 8.

In this work, we contribute to a better understanding of bias in VQA models as follows:

- **VISQA an interactive visual analytics tool** which helps experts to explore the inner workings of transformers models for VQA by displaying models' attention heads in an instance-based fashion.
- **A set of visualizations to address bias in VQA systems** designed to explore models' performances in real-time with altered attention, and/or by asking free-text questions.
- **Insights on the emergence bias in transformers for VQA** gained by experts through an in-depth analysis using VISQA, along with an evaluation of its usability to estimate models' predictions and eventually bias exploitation.

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

2 BACKGROUND

We first introduce some background on understanding neural networks in vision and language reasoning, the context of this paper, and we will provide a short and concise introduction into transformers, the type of neural networks which currently dominates academic and industrial research in language reasoning, and in vision-based language problems.

2.1 Transformers and Attention

Following the introduction and success of transformers applied to natural language processing tasks [16, 56], transformer-based models were also proposed for VQA [21, 62]. Their key strength is the ability to contextualize input representations, i.e. to take input items like words and objects, each one represented in a vectorial form called "*embedding*", and to enrich them, adding information on relationships. This is achieved by series of transformations of the input vectors, which effectively encodes the reasoning process, and the underlying key mechanism is attention (self-attention). We start with a brief overview of how a typical language transformer works by applying it to encode the question *"What is the name of the clothing item that is white?"*.

Step 1: Preparation: Sentence Tokenization. We split the question into elementary language items (called "tokens") with the WordPiece tokenizer [60]. Two special tokens are added at the beginning and at the end of the sentence (respectively '`CLS`' and '`SEP`'). While the latter encodes the end of the sentence, the former is of the uttermost importance; it is transformed by the model as are the other tokens, with the difference that the '`CLS`' token is transformed to encode the task-specific information, and the answer. In the given example, at the end of the transformation, it is expected to contain the information required to predict the name of the white clothing. Each token (including special tokens) is then projected into a high-dimensional vector space through a learned dictionary, resulting in a sequence of N token embeddings: $\boldsymbol{L} = [\boldsymbol{l}_{CLS}, \boldsymbol{l}_1, \ldots, \boldsymbol{l}_i, \ldots, \boldsymbol{l}_{N-2}, \boldsymbol{l}_{SEP}], \boldsymbol{l}_i \in \mathbb{R}^n$, n being the (chosen) embedding dimension.

Step 2: Attention Maps. Transformers progressively contextualize each of the N input embeddings by a sequence of self-attention operations (layers), with the objective of making each token embedding "aware" of the neighboring embeddings. In our example, it might be helpful to combine the information from the embeddings of tokens *'item'*, *'clothing'* and *'white'* into one "enriched" embedding describing the referred object, as the three words are semantically related. More generally, the so-called "self-attention operations" (layers) are the key elements of this type of model. They are implemented via the calculation of *attention maps* $\boldsymbol{A} = \{\boldsymbol{\alpha}_{ij}\}$, which reflect the N^2 pairwise interactions between the tokens, each $\boldsymbol{\alpha}_{ij}$ being the similarity between token embeddings i and j. The similarity function, here, the scaled dot-product, is calculated between a trainable projection of the embedding i, called *query*, and a trainable projection of the embedding j, called *key*. The per-token attention energy is normalized into a probability distribution using a row-wise softmax. In our example, the row vector

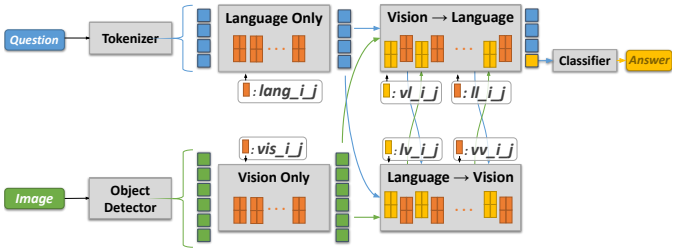

Fig. 2. An Illustration of the VL-Transformer architecture used in the paper. Question and image are first tokenized and then encoded using vision (in green) and language (in blue) only transformers [56], followed by (bi-directional) inter-modality transformers [54]. The answer is predicted from the *"CLS"* token. Yellow and orange rectangles represent, respectively, inter- and intra-modality attention heads. i and j are the layers and head indices used for naming attention heads through the paper.

$A_7 = \{\boldsymbol{\alpha}_{7j}\}_{j\in\{0,\ldots,N-1\}}$ encodes the N interactions between *'clothing'* and the other words.

Step 3: Token Updates. Each token embedding is updated as a linear combination of a trained function of the input embeddings, called *values*, weighted by the attention map $\boldsymbol{A}$. Hence, the model transforms each token by learning a strategy for looking at specific other words.

Multi-headed Attention. As more classical neural networks, Transformers are organized into a sequence of layers. Each of these layers bundles together *multiple attention heads* working in parallel, which allows the model to learn different cooperating strategies. Different heads might learn different syntactic or semantic language functions, as shown in [58] for language models, and as we will show in the experimental section for vision and language reasoning. The outputs of the attention heads are combined with standard neural network blocks.

2.2 Vision-Language (VL)-Transformers

Transformers have been extended to reasoning on multiple modalities, in particular vision and language, through different types of layers: Language-only and vision-only layers, referred to as *intra*-modal layers, and language-vision layers, referred to as the *inter*-modal ones. Fig. 2 depicts the transformer architecture designed for VQA which we call "*VL-Transformer*". Each layer is named as *X_i_j*, where *X* denotes the layer type (*e.g.* vision-only intra-modal layer, vision-language inter-modal layer, etc.) and i and j index, respectively, layer and head.

Intra-modality. Both modalities are first processed in two independent streams (cf. Fig. 2): heads *lang_i_j* encode question words, and *vis_i_j* heads encode visual objects detected in the image by an off-the-shelf object detector, a mainstream approach in VQA [4, 54]. Visual embeddings are the concatenation of 2048-dimensional object appearance embeddings and 4-dimensional bounding box coordinates.

Inter-modality. Subsequent layers combine information between both modalities, c.f. Fig. 2, in a bidirectional way: from question words to visual objects in *lv_i_j*, and vice-versa in *vl_i_j* (*lv* means 'language to vision' while the opposite *vl* means 'vision to language'). This requires a minor, but essential, modification of the attention mechanism. Intuitively, and a bit simplified, in vision-to-language heads, the language (word) embeddings are transformed by taking each word and checking its similarity to the full set of visual input objects, and vice-versa for language-to-vision heads. Details are given in supplementary materials. As shown in Fig. 2, each *lv* or *vl* attention head is immediately followed by an intra-modal attention head called, respectively, *vv* or *ll*.

Predicting the answer. The answer is produced by decoding the final representation of the "*CLS*" token using a 2-layered neural network. It predicts a probability vector over the most frequent answers found in the training set, the answer with the highest score is then chosen.

Training details. For the experiments in this paper, we set the embedding size to d=128 and the number of heads per-layers to h=4. Following [54], our model is composed of 9 language only and 5 vision only intra-modality transformers layers, and 5 language $\rightarrow$ vision and vision $\rightarrow$ language layers. In addition to the VQA objective, we train the model parameters also on MS-COCO [39] and Visual-Genome [35] images following the semi-supervised BERT [16]-like strategy introduced in [54]. In particular, the model is trained to perform simple tasks such as recognizing masked words and visual objects, or predicting if a given sentence matches the question. After pre-training on these auxiliary tasks, the model is fine-tuned on the GQA [29] dataset with the VQA objective. Our VL-Transformer is a variant of the LXMERT model [54], in line with the many works adapting BERT [16]-like pre-training to vision and language tasks [14, 20, 38, 42, 53].

Discussion. In this paper, we focus on the interpretation of the attention maps, as they contain crucial cues on the internal reasoning in transformers. These maps highlight to what extent a given token has been contextualized by which neighbors, high attention $\boldsymbol{\alpha}_{ij}$ indicating strong interaction between tokens i and j. We argue, that attention maps provide strong insights on how our the model handles interactions between the question the image.

3 Related Work

Our work is related to building visual analytics tools for interpretability of Deep Learning. Our design targets in particular the study of attention maps from transformers models to grasp insights on their potential exploitation of bias. This section reviews previous work on the visual analysis of deep learning models, and a review of previous work from machine learning communities to tackle bias in VQA systems.

3.1 Visual Analytics for Interpretability

In recent years, the visual analytics community has joined forces with machine learning experts and provided contributions improving the interpretability of deep neural networks [27], by providing insights on their inner workings. Those models are often considered black-boxes due to the large amount of parameters and data they manipulate to reach a decision. Prior work focused on the analysis of image processing models, known as convolutional neuronal networks, by exposing their gradients over the input images [63]. This approach, enhanced with visual analytics [41], and provided glimpses on how the neurons of those models are sensible to different patterns in the input. More recently, CNNs have been analyzed through the prism of attribution maps in works such as Activation-atlas [10] and attribution graphs [26].

On the other side, natural language processing (NLP) with recurrent neural networks, have also been explored through static visualization [30] which provided insights, among others, on how those models can learn to encode patterns in sentences beyond their architectures in capacities. Interactive visual analytics works such as LSMTViz [52], and RetainVis [36] have also addressed the interpretability of those models through visual encoding of their inner parameters, which can then be filtered and completed with additional information. Those parameters are collected during forward pass on models, as opposed to RNNbow [12], which has the particularity to focus on visualizing gradients of those models through back-propagation during training.

3.2 Interpretability of Attention

More recently, models with attention [56] increasingly gained popularity due to their improvement of state-of-the-art performance, and their attention mechanisms which may be more interpretable than CNNs and RNNs. The interpretability of attention models similar to the transformer models used in this work, initially designed for NLP, has also been addressed by visual analytics contributions. Commonly, in works such as [45, 51, 57], the attention of those models is presented, in instance-based [27] interfaces as graphs with bipartite connections that can be inspected to grasp how input words are associated with each other. Attention Flows [15] addresses the influence of BERT pre-training on model predictions by comparing two transformers models applied to NLP. Similar to our work, such a tool displays an overview of each attention head with a color encoding their activity. Those methods are specific to NLP tasks. In this work, we address the challenges provided by the bi-modality of vision and language reasoning, and expand the interpretability of VQA systems which can rely on visual cues or dataset biases. Current practices of VQA visualization include attention heatmaps of selected VL heads based on their activation [37] to highlight word/key-object associations, global overview heatmaps of

attention heatmaps towards a specific token [9], and guided backpropagation [25] to highlight the most relevant words in questions. Following those works, VISQA provides a visualization of every head's attention heatmaps and word/object associations, along with an overview of their activations.

Our focus is on post-hoc interpretability [40], i.e., the analysis of a trained model's decision policy after-the-fact. Our approach relies on instance-based analysis, which displays inner model parameters with respect to current input. Such analysis is often combined with direct manipulation mechanisms designed to let users experiment with desired input conditions, like drawing the input [11]. Our working hypothesis is that a transformer-based model's mode of operation, i.e. whether it is reasoning or exploiting dataset biases, is observable from its trained parameters, and in particular from attention maps, an intermediate representation dependent on parameters.

3.3 Bias Reduction in VQA

Bias reduction has been addressed on the data side through cleaning and balancing. In particular, GQA [29] focuses on semantics with the help of human-annotated scene graphs and automatically generated questions. On the contrary, VQAv2 [24] dataset is crowdsourced, which leads to more natural questions, but includes cognitive and/or social biases [19] as well as annotation mistakes. In addition, evaluation benchmarks have been specifically designed to identify the presence of bias dependencies. VQA-CP [3] proposes to evaluate models bias dependency by introducing distribution shifts between training and testing sets. However, this approach has limitations to evaluate the decisions and biases exploitation of models as it emphasizes the diversity in answers rather than the reasoning itself. As result, random predictions can also contribute to improving a model's score on this benchmark [50,55]. On the other side, GQA-OOD [32] undertakes a similar strategy while keeping the training set distribution untouched. Instead, the authors propose to evaluate the VQA performances on question-answer pairs regarding their frequency in the dataset. This makes it possible to evaluate both in- and out-of-distribution performances and, at the same time, estimating the reasoning ability of the system. Indeed, if a model's prediction is correct while the question-answer pair is rare, there are chances that the model has not used statistical biases. These datasets and evaluation benchmarks have lead to the conception of diverse bias-reduction methods. While an exhaustive survey of these methods is out of the scope of this paper, one can mention families of approaches such as training regularization [50], or counterfactual learning [1,23].

In this work, we define bias as the way how a model learns regularities or shortcuts from its training dataset, which may push it to provide answers which it frequently encountered as correct during training, without even considering information from the input image. Technically, we evaluate bias as introduced in [32]. Hence, every question from the dataset is grouped using their topic (e.g., "furniture") and function (i.e., task extracted from the semantic of the question). Both the semantic and topic metadata are provided by the GQA dataset. Then for each kind of question, the frequency of their answers is computed. We estimate that the model exploits bias when it incorrectly predicts an answer which is among the 20% most frequent answers for the given question, while its ground-truth answer is among the 20% least frequent. While our proposed tool allows to partially open the black-box of transformer-based neural vision and language models in a very general sense, making it possible to inspect how they handle relationships between data items, we nevertheless specifically focus on the important problem of bias reduction.

4 Motivating Case Study

This work was primarily motivated by growing concerns in the field over bias exploitation of models trained in large-scale settings, in particular when trained on very broad problems like vision and language reasoning [2,24,44]. Our own contributions in this area include new benchmarks [32] and additional supervision and regularization for neural models [31]. Here we extend our previous efforts by providing a tool for instance-level visualizations and performing visual analytics on a single sample. This choice was ultimately taken when comparisons

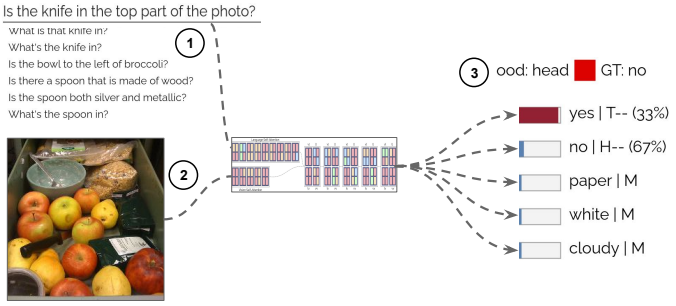

Fig. 3. When asked *"Is the knife in the top part of the photo"* ① the tiny-LXMERT model, with the image of a knife at the bottom ②, incorrectly outputs *"yes"* ③ with more than 95% confidence. While an exploitation of bias can be considered, we can observe that the answer *"yes"* represents only 33%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

of different trained models through statistics failed to provide concrete answers on the sources of confusion and errors. Statistical models enabled us to drill down examinations to a certain minimal level of aggregation, for instance, linguistic function groups, but this kind of analysis was not fine-grained enough.

We illustrate the advantages and the power of instance-level visualizations with our contribution, VISQA on the following case study, which is also available in video form in the supplementary material. It is based on the exploration of a tiny version of the state-of-the-art neural model LXMERT [54] as described in sec. 2.2. We provide it with the following input instance, i.e., the image given in Fig. 3(②), and associated question *"Is the knife in the top part of this photo?"* ①. The correct ground truth answer is of course *"No"*, but the baseline tiny-LXMERT model incorrectly answers *"Yes"* ③. We see the frequency of the different possible answers provided in the interface, and observe that the wrong answer *"Yes"* is not the most frequent one for this kind of question as *"No"* is the correct answer 67% of the time, which does not provide evidence for bias exploitation. The objective is to use VISQA to dive deeper into the inner workings of the model.

A first step is to analyze whether the model is provided with all necessary information. While the input image itself does contain all clear pictures of the answer, the neural transformer model reasons over a list of objects detected by a first object detection and recognition module (Faster R-CNN [48]) which may output errors. In the rest of this paper, we will refer to this tiny-LXMERT as the *noisy model*.

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

Can attention maps provide cues for reasoning modes? VISQA focuses on attention maps, which are a key feature of transformer-based neural models, as they fully determine relationships between input items. Users can select different heads and explore the corresponding attention maps. For the example case, we are interested in checking the correspondence between the question word *"knife"* and the set of bounding boxes, which should provide us with evidence whether the model was capable of associating the concept with the visual object in the scene, which is, of course, not sufficient for correctly answering, but a necessary step. This verification is non-trivial, however, since the model is free to perform this operation in any of the inter-modality layers and heads. VISQA allows to select the different heads, and we could observe that none of the heads provides a correct association. As an example, we can see the behavior of a head in Fig. 4 ①, which associates the word *"knife"* to various objects, mostly fruits. No other

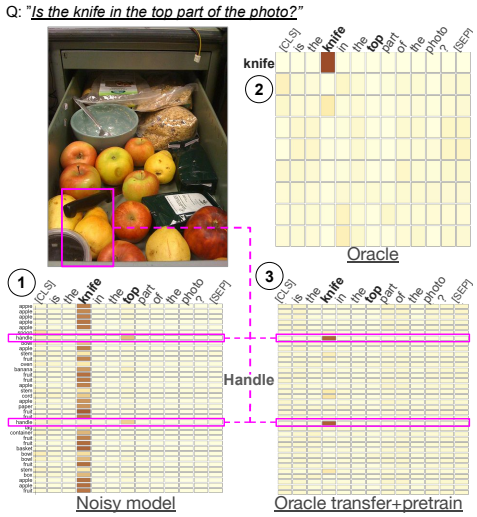

Fig. 4. Visualization of a selected vision-to-language head and attention map for two different models. ① the *noisy model* associates the *"knife"* word with a large number of different objects, including fruit. ② the oracle model learns a perfect association between the word *"knife"* and the *"knife"* object; ③ the oracle transfer model associates the word *"knife"* with two different bounding boxes of type *knife handle*, whose embeddings are sufficiently close for correct reasoning. Head selections are not comparable between models, we therefore checked for permutations.

head is found indicating a more promising relationship.

Is computer vision the bottleneck? From the example above, as well as similar observations in other instances, we conjecture that the computer vision input pipeline (notably, the imperfect object detector) is one of the main bottlenecks in preventing correct reasoning. To validate this hypothesis, we explored training an *Oracle* model with perfect sight, which thus takes as input the ground truth objects provided by human annotation instead of the noisy object detections by a trained neural model. This improves the performance of the model considerably, reaching ∼80% accuracy on the difficult questions with rare ground-truth answers, compared to ∼20% for the standard model reasoning on noisy input. This particularly high difference in performance for questions with rare answers suggests a higher performance in correct reasoning of the oracle model. By loading this model into VISQA, we observe in Fig. 4 ②, that there exists an attention map which associates the word *"knife"* to a visual object *"knife"*, which, we recall, is an object indicated through human annotation. This correct association is reassuring, but by itself does not yet guarantee correct reasoning — further exploration is possible, but we will now concentrate on this problem of finding correspondences between words and visual objects and explore this question further.

Transferring reasoning modes. Given these observations, we conjecture that a transformer model with perfect sight can learn modes of reasoning which are less biased than models trained on real but noisy input data, which is an insight we gathered from using VISQA as a tool for visual analytics. Oracle models, on the downside, are not deployable to real-life problems, as they work on human annotations only, per definition. We explore a solution to transfer reasoning modes from oracle models to noisy input data, which can be done using knowledge transfer by parameter initialization [61]. In more detail, we pretain the Oracle model on ground-truth data, once it reaches convergence, use its weights as initialization for the model using noisy inputs.

We load this model, which we call *Oracle Transfer*, into VISQA, with the objective of exploring whether reasoning modes have been transferred into a deployable model capable of providing answers given real input. Fig. 4 shows the vision-to-language attention maps from the same layer as the ones explored above (*Note: Each layer contains several attention heads, and these heads are not ordered. Reasoning associated with a given head in one model can correspond to the reasoning mode of a different head in another model. We cope with these possible permutations in our experiments by searching over all possible heads of a layer*). We can observe that the model's attention is drawn towards *"handle"* visual objects, which are parts of the only knife in the picture. While the object classes *"knife"* and *"handle"* are logically different, we can conjecture that their vectorial feature embeddings are different but sufficiently similar to allow reasoning. More importantly, we can deduce that the model adapted to the absence of the knife object by relying on what is available, i.e. knife handles. This leads the model to correctly answering *"No"*, as did the Oracle model on ground truth data, and in contrast to the baseline *noisy model* without transfer — see also the illustration in Fig. 1. We further describe this Oracle knowledge transfer, and improvements it provides to transformer-based VQA in a different associated publication [33]. This model will be used in evaluations performed by deep learning experts in section 8. All models, noisy, oracle, and oracle transfer, can be tested and explored online in a prototype.

5 Design Goals

Prior to the design of VISQA, when working on an associated publication [33], we conducted discussions with experts (co-authors of this paper), about their workflow to analyze bias in VQA systems. From those discussions, and literature review introduced in sec. 3, we distill their needs in the following four main themes of design goals for instance-based analysis.

G1 Examine the performances of each instance for a given model. To investigate bias in VQA systems as introduced in sec. 3.3, it is first important to examine the model predictions, along with its confidence score with respect to its inputs. In order to be useful, those predictions need to be combined with the ground-truth, to estimate if the model is wrong, and how frequent is the ground-truth answer and predictions. Inputs need to be inspected as well as they may convey ambiguities that may be at the beginning of an explanation for a mistake. Finally, due to a large amount of data available for inspection, experts may prioritize inputs the more likely to be biased, i.e., those with infrequent ground-truth answers and frequent predictions.

G2 Browse the attention of the model for an instance. Analyzing the attention maps of LXMERT models is crucial for understanding what factors influenced its decision, and eventually whether or not the model attends to both language and vision. While visualizing individually each attention map is feasible, we aim at improving such an exploration, by contextualizing each attention head with their neighborhood (i.e., other attention heads directly connected to it), and position within the model. This is relevant as attention heads get closer to the output, they both encapsulate previous attention, and may be more influential on models' decisions. In addition, experts may need to prioritize heads conveying salient attention, thus those heads need to be summarized and/or emphasized. Finally, for in-depth analysis, experts need to visualize the complete attention map and link each of their elements to human-understandable information, i.e., words of the question and bounding boxes within the input image.

G3 Link attention to language tasks. Once a relevant attention head is observed by an expert, the user should be able to contextualize it with the rest of the dataset. In particular, experts are interested in evaluating whether or not this head is responsive to certain tasks provided by the semantics of questions (e.g., find a color), or rather if the head is responsive to certain topics (e.g., clothing). Such information can be provided in the VQA training dataset, considered as categories.

G4 Explore alternative scenarios. Once cues on how the model uses its attention to output a decision are gathered, the next step, is to test this knowledge by querying the model on altered input or parameters. Ultimately the experts desire to answers questions such as: "would the model have a similar attention if the question were on another object of the image?", or "is this head or group of heads relevant for the final decision?". This can be regrouped into two categories first, the possibility to ask free-from questions, and second the possibility to modify the model's attention. In order to be usable, and due to the number of queries an expert may need to execute, both those manipulations require to interact with the model in a reasonable amount of time, e.g., less than a couple of seconds.

6 DESIGN OF VISQA

We designed VISQA, an visual analytics tool designed to facilitate in-depth analysis of the internal structure of transformers models applied to visual question answering. VISQA implements attention heads and interactive heatmaps visualizations, and other interactions such as free-form question and pruning mechanisms, to assess whether a model resorts to reasoning or bias exploitation when answering questions.

6.1 Workflow

Through iterative design resulting from frequent meetings with VQA experts co-authors of this work, we extracted the following workflow of use of VISQA, based on their experience analyzing VQA systems, and the mantra *overview first, zoom and filter, then details on demand* [49]:

1. the user picks and loads into the model an instance, informed by the likeliness of the question to be answered with bias;
2. the result of the model is presented both as attention heads (internal structure of the model), as well as the top-5 predicted answers;
3. the user then may interact with the model internal structure (e.g., heads intensity, attention maps, etc.) which triggers updates to the statistical views;
4. bounding boxes are displayed on the input image to reflect based on attention heat-map selections, showing how the model associates words and visual objects.

VISQA can also be used beyond this typical workflow for in real-time query of the model by asking user-defined questions, or pruning attention heads, as further detailed in Sec. 6.4.

6.2 Visualization of Instances

At its core, VISQA is organized following an end-to-end approach, from model input to output. This is done to contextualize attention heads with their neighborhood and position with the model (**G2**). We added input selection to guide users in their exploration, and a visual summary of the attention of the model to keep it visually compact.

Image ranking-by-feature (Fig.1 ①). In order to ease user exploration over the complete dataset, VISQA displays images in the top bar from left to right based on the likelihood of their questions to be answered using statistical biases. To do so, we classify questions using ground truth answers, as proposed in [32]: top 20% of the most frequent answer will be classified as *Head*, as opposed to *Tail* which describes questions with the least frequent answers. We attribute a score to each image based on their *Head*-questions/*Tail*-questions ratio. The more an image has *Tail*-questions over *Head*-questions, the higher its score is. The underlying hypothesis is that frequent answers will be chosen more likely when a model tends to exploit biases (e.g. *"Yellow bananas"*). Also, frequent questions are harder to analyze since if any bias is exploited by the model, it will answer correctly. Answers and their frequencies are displayed in (Fig.1 ② right) following design goal (**G1**).

Instance view (Fig.1 ②). This view, inspired by VL-Transformer representations illustrated in Fig. 2, is the root of any analysis of attention maps using VISQA (**G2**). It matches the internal data flow through the internal structure of the model, from left to right: the input image/question pair, layers and heads with intra-modality layers first, and finally the answer output distribution (encoded as horizontal bars). A particular design decision was to display all attention maps at once, using a single-colored rectangle encoding the attention intensity as k-number [47] (see next paragraph for details). In the *Instance view*, attention heads can be selected with a mouse-over interaction as illustrated in Fig.1 ③ in order to provide details on demand, by displaying the corresponding attention heat-map Fig.1 ④ and head statistics.

Visual summary. Our model uses 136 attention maps with dimensions varying with respect to the number of words in the question and bounding boxes provided. As displaying all of those matrices would prevent experts to analyze them in a reasonable amount of time, we rely on summarizing each of them to a single scalar. Such a scalar, referred to as *k-number* [47], represents the normalized amount of tokens per row summed up to reach a threshold of 90% of energy. A *k-number* close to 0 indicates that the corresponding row has peaky attention focusing on only one column (as seen in Fig. 4 ①), and a high *k-number* encodes a uniform attention (as in Fig. 4 ②). Then we combine each of those *k-number* together using either *min*, *median*, or *max* functions. Such functions can be selected in VISQA by users, depending on the attention maps intensity they want to investigate. VISQA provides this interaction because for a head to have a low *k-number*, the majority of its rows needs to be highly activated. This can shadow attention maps with less than half of their rows with peaky attention. In VISQA, the *k-number* is discretized and color encoded in 4 categories as it follows: k<12 k<25 k<50 k<99. The decision to use a logarithmic discretization and color encoding, as introduced in [47], was done to emphasize peaky attention maps ($k < 12$) that need to be prioritized for analysis. In addition, this discretization, used throughout VISQA, is particularly useful to regroup instances in head-stat view Fig. 1 ④ and select them by clicking on the corresponding square (legend on the right of ② in Fig. 1) for *Head pruning* further detailed in section 6.4. Finally, this reduces the learning curve of VISQA, as experts are familiar with this discretization of *k-numbers*.

6.3 Visualization of Selected Heads

VISQA provides details for a selected head in the Instance view using the attention map and head statistics.

Attention Maps (Fig.1 ⑤). are represented as heat-maps, with cell colors encoding the attention intensity over a sequential, single hue scale from *no attention* (i.e., 0) in beige, to *full attention* (i.e., 1) in brown. This matrix-based approach contrasts with bipartite connections representations found in Seq2Seq-Vis [51] and BertViz [57]. We use matrices as they provide a clutter-free representation of attention and they can display multiple heads at once by aggregating connections. However, it may suffer from visual clutter issues as the number of words grows. Seq2Seq-Vis tackles this issue by using interactions to hide graph lowest connections (i.e., lowest attention). In our case, we argue that visualizing the attention of one head at a time can lead to a better understanding of its function in the model. Such an exploration of head functions is further detailed in Sec.8.

Head Statistics (Fig.1 ④). are represented using three charts. A vertical area chart (leftmost chart) represents the distribution of *k-numbers* of the selected head over the complete validation dataset (around 1500 image/question pairs). The vertical axis encodes the values of *k-numbers*, while the horizontal axis encodes the density of the corresponding *k-number*. The current *k-number*, for the selected head, which corresponds to the image/question pair loaded in VISQA, is represented as a horizontal red bar positioned on the vertical axis. This area-chart provides insights such as the detection of useless heads with constant high *k-number* which can be reduced to calculation on average overall items instead of selecting specific items. In contrast, heads with constant low *k-number* can be interpreted as conveying key information. More specialized heads, with bi-modal *k-number* distributions, can also be observed. Two stacked bar-charts represent the *k-numbers* of the selected head grouped by question operations (**G3**). Question operations are ground truth information provided by the GQA dataset [29] describing the semantic reasoning operation of asked questions (e.g., select, query..). Each bar is associated with an operation, and its length encodes how many questions in the dataset are using the corresponding operation.

6.4 Interacting with Models

VISQA also includes features, complementary to the usual workflow introduced in Sec.6.4, designed to investigate hypotheses on reasoning.

Free-form questions. By default, VISQA loads the GQA dataset [29] to provide images and questions. But at any time, users can type and ask free-form open-ended questions (**G4**). Such an interaction allows investigating the model's bias exploitation. For instance, when asked the following question from the GQA dataset *"Is this a mirror or a sofa"*, the model correctly outputs "*mirror*". However, when asked the following user-inputted question *"Is there a mirror in this image?"*, the model fails and outputs *"no"*. This suggests that the model might have exploited biases when it answered the first question, which is supported by the fact that in the GQA dataset, *"mirror"* is the correct answer to the question *"Is this a mirror or a sofa"* in 85% of all cases.

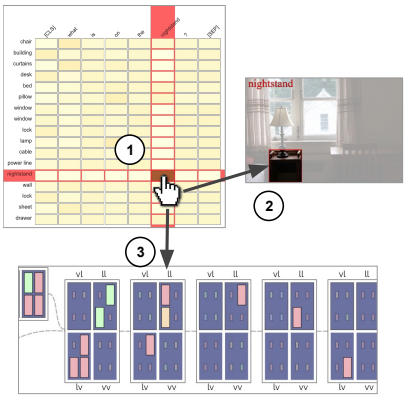

Fig. 5. Hovering the mouse over a cell of the attention maps ① filters the corresponding object bounding box in the input image ②. While clicking on this cell filters attention heads in *instance-view* to display those within which the selected cell is highly activated.

Head filtering (Fig.5). As shown in Fig.5 ①, attention heat-maps feature two interactions. First, hovering with the cursor a row, a column, a cell, which corresponds to an object in the image, automatically displays its corresponding bounding box along with its label over the input (Fig.5 ②). The user can also, by clicking on a row, column, or cell, filter attention heads to only keep the ones in which the corresponding clicked element has attention above a threshold. For rows and columns which contain multiple attention values, such a filtering process will merge those values by using one of three functions *min*, *median*, and *max* depending on a user selection. The result of such a filtering, as displayed in Fig.5 ③, occurs in the *instance view* in which the size and opacity heads that do not match with the user's query are reduced while the others are preserved. Such interactions facilitate seeking for heads in which a specific association is expected e.g., a word in the question with an object of the image required to answer.

Head pruning. Users can select attention heads by clicking on them in the *instance view*, or by their *k-number* category. Such a selection can then be used to prune the corresponding head for the next forward of the model. Pruning here means that the attention head does not perform any focused attention, but uniformly distributes attention over the full set of items (objects or words). Each row of a pruned attention map is thus the equivalent of an average calculation. At any time, users can request a new forward pass of the model by clicking on the top left button "ask" (**G4**), which allows to see the effect of the configured pruning on the model's predictions. This can be used in order to test hypotheses on attention head interpretations as explored in Sec. 8.2.

7 Implementation

The GQA [29] standard dataset provides question/image pairs along with their answers, the ground truth of bounding boxes, and semantic descriptions of questions. By default, VisQA provides around 1500 question/image pairs, but as images are loaded progressively when users request it, such a quantity can be increased without affecting performances. The models have been trained on a significantly larger amount of training data (about 9M image/sentence pairs), which is different from the validation data on which the performance is evaluated. The user interface of VisQA is implemented using D3 [6], and directly interacts with transformer models implemented in Pytorch [46], using JSON files through a python Flask server. As the possibility to ask free-form questions offers an infinity of possible combinations, each question/image pair is forwarded through the model in a plug-in fashion, i.e., without altering the model and its performances.

8 Evaluation with Domain Experts

We conducted a user study with 6 experts with experience in building deep neural networks, who were not involved in the project or the design process of VisQA. We report on their feedback using VisQA to evaluate the decision process of the *Oracle transfer model*, with 57.8% accuracy on GQA, introduced in Sec.4 on several provided problem instances, as well as insights they received from this experience. Hypotheses drawn from single instances cannot be confirmed or denied, but as illustrated in the following sections, such a fine-grained analysis aims to provide cues (often unexpected) that can later be explored through statistical evidence outside of VisQA.

8.1 Evaluation Protocol

For each expert, we conducted an interview session lasting on average two hours. Sessions were organized remotely and began with a training on VisQA, showing step-by-step how to analyze attention maps. During this presentation, experts were able to ask questions. The study then began with questions on 6 problem instances, i.e., image/question pairs loaded into VisQA in a browser window on participants' workstations. Those instances were balanced between the prediction failures and successes, head or tail distributions of question rarity as described in sec. 3.3, as well as our estimation on whether the model resorts to bias for this instance grasped using VisQA. VisQA, configured as conditioned during evaluations is accessible online at: (https://theojaunet.github.io/visqEval/). The model outputs were hidden and the experts were asked to use VisQA to provide an estimate for two different questions: (1) will the model predict a correct answer, (2) what will it be?, and (3) does it exploit biases for its prediction, or does it reason correctly? During this part of the interview, experts were asked to explain out loud what lead them to each decision. Once those questions were completed, post-study questions were asked on the usability of VisQA, such as *"Which part of* VisQA *is the least useful?"*, and *"What was the hardest part to understand?"*.

Results. The ability of users to predict failures and specific answers of VQA systems has already been addressed through evaluation [13] under different conditions. The experiment closest to ours is question+image attention [43] with instant feedback — similarly to ours, users were asked to estimate whether a model will predict a correct answer when provided with attention visualizations of the model, and reaching a similar score of ∼75% accuracy. The difference is that in [43] attention is overlaid over the visual input, whereas our attention maps allow to inspect reasoning in a more detailed and fine-grained manner, and not necessarily tied to the visual aspects. The similarity in results changes when users are asked to provide the specific answer predicted by the model: this accuracy drops to 61% in our case, and to 51% in [13]. While our results are promising, they cannot be directly compared to their results due to the different pool and amount of users. Future work will address studies on a larger number of human experts.

More importantly, our work focuses on qualitative results of bias estimation in which experts obtained a precision of 75% on whether the model exploited any bias. We extracted the ground truth estimate by comparing the rarity of the question, following [32]. Supplementary material provides more details on the evaluation protocol and experts' performances. These results are encouraging, as they provide a first indication that the reasoning behavior of VL models can be examined and estimated by human users with VisQA. While 75% of performance reasoning vs. bias is not a perfect score, it is also far away from the random performance of 50%, which is important given the large capacity of these models, which contain millions of trainable parameters.

8.2 Object Detection and Attention

To provide an answer, a model must first grasp which objects from the image are requested and thus are essential to focus on. Such an association needs to occur early in the model as those objects are needed for further reasoning. The experts widely observed high intensity in the first language-to-vision (LV) layer. As illustrated in Fig. 6, when asked *"Is the person wearing shorts?"*, the attention map *LV_0_1* has peaky activations in the columns *"person"* and *"shorts"*. This can be interpreted as the model correctly identifying with its self-attention for language that those two words are essential to answer the given question. In Fig. 6, the word *"person"* is associated with the bounding boxes labeled as *"woman"*,*"shirt"*,*"shoe"*,*"leg"*, while the word *"shorts"* is associated with the *"shorts"* bounding box. Based on this observation, all experts concluded that the model correctly sees the required objects, and more broadly over the evaluation instances, that the first LV layer

Fig. 6. When asked *"Is the person wearing shorts?"*, the *oracle transfer* model successfully answers *"yes"*. It can be observed in its first Language-to-Vision attention maps, that the word *"shorts"* (column) is strongly associated with the object *"shorts"* (row). The same phenomenon is also observed for the word *"person'*, strongly associated with objects labeled as *"woman"* among others.

might be responsible for the recognition of objects with respect to the question. One of the experts mentioned that therefore, *"if we don't see a good word/bounding-box association here, the model can hardly cope with such a mistake and might exploit dataset biases"*. In order to verify such a statement, we pruned the four heads in this LV layer, to observe how the model would behave with no association in them. From such pruning, we observe that the following vision-to-language (VL) layers have lower attention distributions, close uniform in some cases. In addition, after pruning, the model's prediction wrongly switched from *"Yes"*, a rare answer (in Tail), to *"No"*, the most frequent one.

8.3 Questions with Logical Operators

During the evaluation, experts were shown two instances with questions containing the word *"and"*. Such instances are interesting because, as one of the experts mentioned, *"this word has a lot of importance is this question"*. To answer correctly, the model needs to grasp that it must analyze the image over two different aspects. With the image, illustrated in Fig. 7, and asked *"Are there both knives and pizzas in this image?"*, the model fails and answer *"yes"*, the most frequent answer despite having no knife in the picture nor provided bounding-boxes. However, when asked *"Are there knives in this image?"* the model correctly answers *"no"*. This suggests that the model failed to grasp the meaning of the keyword *"and"*, and thus that the self-attention language heads might associate wrong words. Also, swapping the terms *"knives"* and *"pizzas"* in the question, yields the correct answer, i.e., *"no"*. This indicates that the model ignores the first term when questions contain the operator *"and"*. Using the head-filtering interaction, we can observe that in self-attention heads, the word *"and"* has little to no attention. Instead, the word *"both"* has peaky attention scattered across most of self-language layers, and some language-to-language heads. Pruning those 19 heads makes the model correctly yield *"no"*, regardless of the order the words *"knives"* and *"pizzas"* are in the question. Such a behavior can be observed over our evaluation dataset, in which 34 questions have the keyword *"and"*. On those questions the model, without pruning, can provide a correct answer 62% of cases, up to 64% with the two words around *"and"* are swapped. In opposition, while having the 19 attention-heads with peaky attention for the word *"both"* pruned, the model reached an accuracy of 76%, down to 74% with words around *"and"* swapped. Thus, in the worst case, this pruning of the 19 attention heads illustrated in Fig. 7 is responsible for an improvement of 10% on question contain the operator *"and"*.

8.4 Vision to Vision Contextualization

When asked *"What is the woman holding?"*, with the image in Fig. 8, the *Oracle transfer* model fails and outputs *"remote-control"*, a frequent answer, instead of *"hair dryer"*. This can be interpreted as the model exploiting a statistical bias from its training dataset. However, in such a dataset, *"remote-control"* is not among the 10 most common answers to this question. This raises the question of what leads the model to output such an answer. During evaluation on this instance, experts noticed that the object detector failed to provide a *"hair dryer"* object.

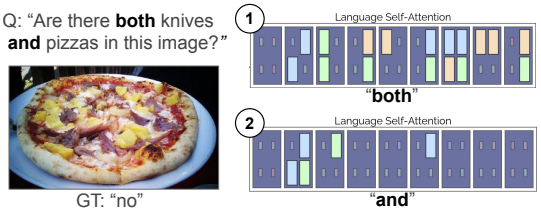

Fig. 7. When asked *"Are there both knives and pizzas in this image?"*, the *oracle transfer* model fails and answers *"yes"*. By filtering heads associated with a selected word, we can observ that language self-attention heads are more responsive to the word *"both"* ①, as opposed to the word *"and"* ②.

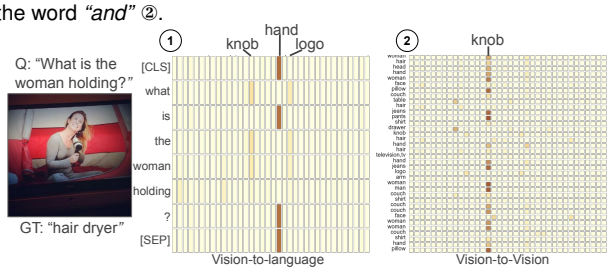

Fig. 8. Without any *"hair dryer"* provided by the object detector, the *oracle transfer* associates in its vision-to-language ① the object *"hand"* with the words {*"[CLS]","is","?","[SEP]"*}. While vision-to-vision focuses on a *"knob"* object ②.

Similar to the use case given in Sec. 4, such a mistake forces the *Oracle transfer* to draw its attention towards other bounding boxes related to the missing object. In this case, as observed by experts, a majority of the vision-to-language reached their highest association between the word *"holding"* and bounding boxes labeled as *"hands"*. Such an association is expected as held objects are directly related to hands, and no *"hair dryer"* bounding box is provided. Among those bounding boxes, we can observe the presence of one labeled as *"television"*, and another as *"knob"* which are associated to *"holding"* and *"woman"* in both vision-to-vision_2_2 and early vision-to-language layers. This suggests that those heads might have influenced the model's predictions towards *"remote-control"* instead of the most common dataset bias. This can be confirmed by pruning those heads which yields a more frequent answer: *"cell phone"*. In addition, one of the experts also highlighted that those attention heads had a high association with the tokens *"[CLS]"*, *"is"*, *"?"*, and *"[SEP]"*. Which the expert interpreted as *"the model correctly transferred the context of the question"*.

9 Discussions, Limitations and Future Work

Usability of VisQA. Overall, VisQA was positively received by experts, during discussions at the end of interviews, they expressed that VisQA is *"well designed"*, *"complete"*, and *"particularly useful as VQA transformers are hard to interpret"*. Two out of the six experts confessed that they felt *"overwhelmed at first"*, but gradually *"grasped where to look"*, and getting *"used to interactions"*. Four out of six experts stated that the *head-statistics* 1 ④ is the least useful feature implemented in VisQA, as it felt *"hard to understand"*. One expert mentioned that such a view is *"useful in theory, but less usable in practice"*. However, the rest of the experts mentioned that the *head-statistics* view helped them while analyzing attention to *"see if heads acted out of their distribution"*. Experts were unanimous, the attention heat-map, and in particular, its interactions, is the most useful feature of VisQA, as it *"provides a lot of information'*, and *head filtering "speeds up the analysis"*. One of the expert used this interaction on the first language to vision layer as an entry-point to grasp if the model had every information required to correctly answer the given question.

Expert Suggestions. As expressed by experts during evaluation, the main limitation of VisQA is how instances can be selected by users. Currently, such selections are handled by a bar at the top of the tool (Fig.1 ①), displaying images ordered from left to right based on the likelihood of their questions to be answered using statistical

biases. However, during interviews, experts mentioned their desire to quickly switch between similar instances in order to grasp if a behavior can be seen across different cases. To do so, experts suggested that such similarity could be measured at three levels: switch to an instance with the exact same question, switch to a similar image, and switch to an instance in which a selected head has similar attention distribution. In addition, currently in VISQA, it is difficult to evaluate how influential a head is over the model's output. To tackle such an issue, experts proposed to encode this information in the representation size of the *instance-view* attention heads. The impact of each head over the model's input could be retrieved through back-propagation, in particular by calculating the gradient of the model outputs with respect to a statistic of the attention head, for instance, its k-number. In addition to addressing those limits and experts' suggestions, we plan as future works to adapt the usage of VISQA on other problems involving transformers, such as machine translation, and to further investigate the role of each attention heads as done in [58].

Scalability. To date, the largest model loaded in VISQA is LXMERT with 12 attention heads per block and 768 hidden dimensions. Using VISQA, we noted that this model has a similar behavior as the tiny-LXMERT (see supplementary material), and can outperform LXMERT with auxiliary losses [34]. In addition, LXMERT is less accurate than the tiny-oracle model we used for evaluation, on questions with rare answers–i.e., those that may be the most susceptible to convey biases. While inspecting this model with VISQA, we noted that the more the number of heads increases, the more summary and filtering of heads became relevant for faster analysis. However, our visual encoding of heads may become tedious to use as the number of heads increases, and the amount of pixels allocated per head decreases. A workaround can be to increase the designated space of the model in instance-view, but ultimately, VISQA will be limited by screen real-estate. Thus, larger models (e.g., UNITER-large [14]), may need their heads to be filtered (e.g., by k-numbers) to avoid being overwhelming, before being displayed in VISQA. Similarly, heatmaps of attention may suffer from the same limitation. However, we decided to use them for attention as the maximum visual object (36 in most VQA systems using a Faster R-CNN), and the average of words per question is known beforehand. For larger sets of inputs, alternative designs such as bipartite graphs combined with aggregation methods to hide elements under a threshold of attention may be more space-efficient. However, we argue that this is a trade-off as displaying every bounding boxes and their attention may be relevant to evaluate a prediction (as depicted in sec. 4).

Generalization. This work focuses on detector-based VL transformers such as LXMERT, however other VQA systems may include single stream models such as [14, 53]. While VISQA is applicable to both single and 2-stream models, we decided to focus on the latest during evaluation. Those models are considered by experts as more interpretable because self-attention layers for language and vision can be observed separately, and there are no significant differences in performance between both architecture [8]. VQA systems' state-of-the-art often alternate between detector-based approaches (e.g., VinVL [64]) and pixel models [28]. However, at its current state, VISQA is not applicable to the latter type, as it would only require adapting to the higher number of visual tokens (depending on the granularity of the method) which raises the challenge of scalability of heatmaps discussed above. Future works can address this through aggregation methods to reduce the number of tokens displayed (e.g., by dividing input images in regions). Finally, our work mainly focuses on insights on reasoning skills grasped from models on the GQA dataset. This decision was taken because it has been argued [29] that it involves a larger variety of reasoning skills (spatial, logical, relational, and comparative) than in a dataset with where questions were annotated by humans directly (e.g., VQA2 [24]) which contains questions targeting difficult external knowledge (e.g., a name of a baseball team). Despite not being presented in this work, VISQA can directly be used on the VQA2 dataset.

Comparison of instances (Fig. 9). Comparison between models, and instances can also yield interesting insights on how a model behaves [15]. Currently, to this end, VISQA memorizes inputs and all intermediate and final results including *k-numbers* and attention maps.

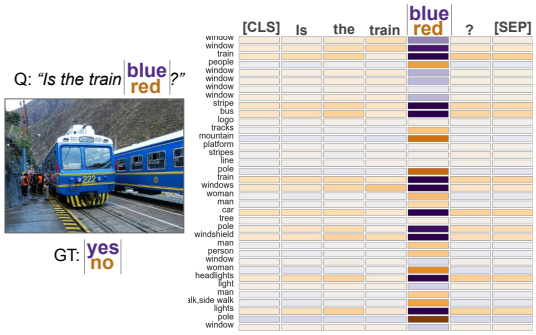

Fig. 9. Difference between the attentions of Head LV_1_0, when asked "is the the train blue?", and "is the the train red?". We can observe that in this head, the attention focuses on different objects (row) depending on the color asked (column).

This state can be saved, and then used at any time and compared to a new current instance through the *compare* button at the top of the *instance-view*. The comparison itself is obtained by computing the difference of *k-numbers*, and complete attention maps. As a result, head representations in *instance-view* now encode, with a single hue color scale, the difference between the two attention maps. Besides, as shown in Fig. 9, attention heat-maps also encodes such a difference using a diverging color scale from dark blue for values close to -1, i.e., a cell with high attention in the previous instance but not in the current one, to brown for values close to 1, the opposite. Comparisons are particularly useful when combined with the possibility to ask free-form questions. As it can be observed in fig 9, the manually asked questions *"Is the train blue?"* and *"Is the train red?"* are responsible for different attention modes between the word representing the color and bounding boxes of the image. By browsing those bounding boxes, it can be observed that the selected head *LV_1_00* associates them, in this case, to the word color only if their color matches. As an example,, in Fig 9 (right), the third row, which corresponds to the bounding box of a blue train, is associated through attention with the word *"blue"* but not the word *"red"*. In order for such a comparison to be relevant, shifts between two instances must be on a few words of the question, as otherwise, attention maps rows and columns would not align between the instances, and thus any difference would occur for wrong reasons. We plan for future work to enhance this functionality of VISQA through better visual encoding, more intuitive interactions, and an evaluation.

10 Conclusion

We introduced VISQA, an interactive visual analytics tool designed to perform an instance-based in-depth analysis of the reasoning behavior transformer neural networks for vision and language reasoning, in particular visual question answering. VISQA allows users to select display VQA instances based on the likelihood of bias exploitation; to display attention head intensities; to inspect attention distributions; to prune attention heads; and to directly interact with the model by asking free-form questions. Our quantitative evaluations are encouraging, providing first evidence that human users can obtain indications on the reasoning behavior of a neural network using VISQA, i.e. estimates on whether it correctly predicts an answer, and whether it exploits biases. VISQA received positive feedback from these experts, who also provided additional qualitative feedback on the nature of the information they extracted on the behavior of different neural models.

Acknowledgments

We would like to thank reviewers from the previous EuroVis'21 submission, as well as coworkers for their valuable discussions, and proofreading: Nicolas Jacquelin, Aurélien Tabard, and Antoine Coutrot. We finally thank our experts for their time and enthusiasm during the evaluation of VISQA. We also acknowledge grant ANR-20-CHIA-0018 "*Remember*" of call "ANR AI chairs *hors centres*".

REFERENCES

[1] E. Abbasnejad, D. Teney, A. Parvaneh, J. Shi, and A. v. d. Hengel. Counterfactual vision and language learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10044–10054, 2020.

[2] A. Agrawal, D. Batra, and D. Parikh. Analyzing the behavior of visual question answering models. *arXiv preprint arXiv:1606.07356*, 2016.

[3] A. Agrawal, D. Batra, D. Parikh, and A. 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*, pp. 4971–4980, 2018.

[4] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. 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*, pp. 6077–6086, 2018.

[5] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, 2015.

[6] M. Bostock, V. Ogievetsky, and J. Heer. D3: Data-Driven Documents. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2011.

[7] L. Bottou. From machine learning to machine reasoning. *Machine learning*, 94(2):133–149, 2014.

[8] E. Bugliarello, R. Cotterell, N. Okazaki, and D. Elliott. Multimodal pretraining unmasked: A meta-analysis and a unified framework of vision-and-language BERTs. *Transactions of the Association for Computational Linguistics*, 2021.

[9] J. Cao, Z. Gan, Y. Cheng, L. Yu, Y.-C. Chen, and J. Liu. Behind the scene: Revealing the secrets of pre-trained vision-and-language models. In *European Conference on Computer Vision*, pp. 565–580. Springer, 2020.

[10] S. Carter, Z. Armstrong, L. Schubert, I. Johnson, and C. Olah. Activation atlas. *Distill*, 2019. https://distill.pub/2019/activation-atlas. doi: 10.23915/distill.00015

[11] S. Carter, D. Ha, I. Johnson, and C. Olah. Experiments in Handwriting with a Neural Network. *Distill*, 2016. doi: 10.23915/distill.00004

[12] D. Cashman, G. Patterson, A. Mosca, N. Watts, S. Robinson, and R. Chang. Rnnbow: Visualizing learning via backpropagation gradients in rnns. *IEEE Computer Graphics and Applications*, 38(6):39–50, 2018.

[13] A. Chandrasekaran, V. Prabhu, D. Yadav, P. Chattopadhyay, and D. Parikh. Do explanations make VQA models more predictable to a human? In *EMNLP*, 2018.

[14] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu. Uniter: Universal image-text representation learning. In *European Conference on Computer Vision*, pp. 104–120. Springer, 2020.

[15] J. F. DeRose, J. Wang, and M. Berger. Attention flows: Analyzing and comparing attention mechanisms in language models. *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

[17] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[18] B. Duke, A. Ahmed, C. Wolf, P. Aarabi, and G. Taylor. SSTVOS: Sparse Spatiotemporal Transformers for Video Object Segmentation. In *CVPR*, 2021.

[19] C. Eickhoff. Cognitive biases in crowdsourcing. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pp. 162–170, 2018.

[20] Z. Gan, Y.-C. Chen, L. Li, C. Zhu, Y. Cheng, and J. Liu. Large-scale adversarial training for vision-and-language representation learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, eds., *Advances in Neural Information Processing Systems*, vol. 33, pp. 6616–6628. Curran Associates, Inc., 2020.

[21] P. Gao, Z. Jiang, H. You, P. Lu, S. C. Hoi, X. Wang, and H. Li. 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*, pp. 6639–6648, 2019.

[22] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman. Video Action Transformer Network. In *CVPR*, 2020.

[23] T. Gokhale, P. Banerjee, C. Baral, and Y. Yang. Mutant: A training paradigm for out-of-distribution generalization in visual question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 878–892, 2020.

[24] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. 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*, pp. 6904–6913, 2017.

[25] Y. Goyal, A. Mohapatra, D. Parikh, and D. Batra. Towards transparent ai systems: Interpreting visual question answering models. *arXiv preprint arXiv:1608.08974*, 2016.

[26] F. Hohman, H. Park, C. Robinson, and D. H. Chau. Summit: Scaling deep learning interpretability by visualizing activation and attribution summarizations. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2020.

[27] F. M. Hohman, M. Kahng, R. Pienta, and D. H. Chau. Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers. *IEEE Transactions on Visualization and Computer Graphics*, 2019. doi: 10.1109/TVCG.2018.2843369

[28] Z. Huang, Z. Zeng, B. Liu, D. Fu, and J. Fu. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849*, 2020.

[29] D. A. Hudson and C. 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*, pp. 6700–6709, 2019.

[30] A. Karpathy, J. Johnson, and L. Fei-Fei. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*, 2015.

[31] C. Kervadec, G. Antipov, M. Baccouche, and C. Wolf. Weak supervision helps emergence of word-object alignment and improves vision-language tasks. In *European Conference on Artificial Intelligence*, 2019.

[32] C. Kervadec, G. Antipov, M. Baccouche, and C. Wolf. Roses are red, violets are blue... but should vqa expect them to? *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[33] C. Kervadec, T. Jaunet, G. Antipov, M. Baccouche, R. Vuillemot, and C. Wolf. How Transferable are Reasoning Patterns in VQA? In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[34] C. Kervadec, C. Wolf, G. Antipov, M. Baccouche, and M. Nadri. Supervising the transfer of reasoning patterns in vqa. *arXiv preprint arXiv:2106.05597*, 2021.

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

[36] B. C. Kwon, M.-J. Choi, J. T. Kim, E. Choi, Y. B. Kim, S. Kwon, J. Sun, and J. Choo. Retainvis: Visual analytics with interpretable and interactive recurrent neural networks on electronic medical records. *IEEE transactions on visualization and computer graphics*, 25(1):299–309, 2018.

[37] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang. What does bert with vision look at? In *ACL (short)*, 2020.

[38] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pp. 121–137. Springer, 2020.

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

[40] Z. C. Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

[41] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu. Towards better analysis of deep convolutional neural networks. *IEEE transactions on visualization and computer graphics*, 23(1):91–100, 2016.

[42] J. Lu, D. Batra, D. Parikh, and S. Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pp. 13–23, 2019.

[43] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 289–297, 2016.

[44] V. Manjunatha, N. Saini, and L. S. Davis. Explicit bias discovery in visual question answering models. In *Proceedings of the IEEE Conference on*

Computer Vision and Pattern Recognition, pp. 9562–9571, 2019.

[45] C. Olah and S. Carter. Attention and augmented recurrent neural networks. *Distill*, 2016. doi: 10.23915/distill.00001

[46] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

[47] H. Ramsauer, B. Schäfl, J. Lehner, P. Seidl, M. Widrich, L. Gruber, M. Holzleitner, M. Pavlović, G. K. Sandve, V. Greiff, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020.

[48] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds., *Advances in Neural Information Processing Systems*, pp. 91–99. 2015.

[49] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *The craft of information visualization*, pp. 364–371. Elsevier, 2003.

[50] R. Shrestha, K. Kafle, and C. Kanan. A negative case analysis of visual grounding methods for vqa. *arXiv preprint arXiv:2004.05704*, 2020.

[51] H. Strobelt, S. Gehrmann, M. Behrisch, A. Perer, H. Pfister, and A. M. Rush. Seq2seq-vis: A visual debugging tool for sequence-to-sequence models. *IEEE transactions on visualization and computer graphics*, 25(1):353–363, 2018.

[52] H. Strobelt, S. Gehrmann, H. Pfister, and A. M. Rush. Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. *IEEE transactions on visualization and computer graphics*, 24(1):667–676, 2017.

[53] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019.

[54] H. H. Tan and M. Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Empirical Methods in Natural Language Processing*, 2019.

[55] D. Teney, K. Kafle, R. Shrestha, E. Abbasnejad, C. Kanan, and A. v. d. Hengel. On the value of out-of-distribution testing: An example of goodhart's law. *arXiv preprint arXiv:2005.09241*, 2020.

[56] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[57] J. Vig. A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714*, 2019.

[58] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5797–5808, 2019.

[59] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local Neural Networks. In *CVPR*, 2018.

[60] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[61] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *NIPS*, 2014.

[62] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian. Deep modular co-attention networks for visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[63] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

[64] P. Zhang, X. Li, X. Hu, J. Yang, L. Zhang, L. Wang, Y. Choi, and J. Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5579–5588, 2021.

[65] H. Zhao, J. Jia, and V. Koltun. Exploring Self-attention for Image Recognition . In *CVPR*, 2020.