How Transferable are Reasoning Patterns in VQA?
Corentin Kervadec, Theo Jaunet, Grigory Antipov, Moez Baccouche, Romain Vuillemot, Christian Wolf

To cite this version:
Corentin Kervadec, Theo Jaunet, Grigory Antipov, Moez Baccouche, Romain Vuillemot, et al.. How Transferable are Reasoning Patterns in VQA?. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2021, Nashville, Tennessee, United States. 10.1109/CVPR46437.2021.00419. hal-03192949

HAL Id: hal-03192949
https://hal.science/hal-03192949
Submitted on 8 Apr 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
How Transferable are Reasoning Patterns in VQA?

Corentin Kervadec\textsuperscript{1,2*}, Théo Jaunet\textsuperscript{2*}, Grigory Antipov\textsuperscript{1}, Moez Baccouche\textsuperscript{1}, Romain Vuillemot\textsuperscript{2}, Christian Wolf\textsuperscript{2}

\textsuperscript{1}Orange, Cesson-Sévigné, France \textsuperscript{2}LIRIS, INSA - École Centrale, Lyon, UMR CNRS 5205, France

corentinkervadec.github.io theo-jaunet.github.io christian.wolf@insa-lyon.fr {grigory.antipov, moez.baccouche}@orange.com romain.vuillemot@gmail.com

Abstract

Since its inception, Visual Question Answering (VQA) is notoriously known as a task, where models are prone to exploit biases in datasets to find shortcuts instead of performing high-level reasoning. Classical methods address this by removing biases from training data, or adding branches to models to detect and remove biases. In this paper, we argue that uncertainty in vision is a dominating factor preventing the successful learning of reasoning in vision and language problems. We train a visual oracle and in a large scale study provide experimental evidence that it is much less prone to exploiting spurious dataset biases compared to standard models. We propose to study the attention mechanisms at work in the visual oracle and compare them with a SOTA Transformer-based model. We provide an in-depth analysis and visualizations of reasoning patterns obtained with an online visualization tool which we make publicly available\textsuperscript{1}. We exploit these insights by transferring reasoning patterns from the oracle to a SOTA Transformer-based VQA model taking standard noisy visual inputs via fine-tuning. In experiments we report higher overall accuracy, as well as accuracy on infrequent answers for each question type, which provides evidence for improved generalization and a decrease of the dependency on dataset biases.

1. Introduction

The high prediction performance obtained by high-capacity deep networks trained on large-scale data has led to questions concerning the nature of these improvements. Visual Question Answering (VQA) in particular has become a testbed for the evaluation of the reasoning and generalization capabilities of trained models, as it combines multiple modalities of heterogeneous nature (images and language) with open questions and large varieties. It has been shown, that current models are prone to exploiting harmful biases in the data, which can provide unwanted shortcuts to learning in the form of “Clever Hans” effects [37, 22].

In this work we study the capabilities of VQA models to “reason”. An exact definition of this term is difficult, we refer to [7, 22] 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. We argue, and in Section 3 will provide evidence for this, that learning to algebraically manipulate words and objects is difficult when visual input is noisy and uncertain compared to learning from perfect information about a scene. When objects are frequently missing, detected multiple times or recognized with ambiguous visual embeddings wrongly overlapping with different categories, relying on statistical shortcuts may be an easy short-
cut for the optimizer. We show that a perfect-sighted oracle model learns to predict answers while significantly less relying on biases in training data. We claim that once any noise has been removed from visual input, replacing object detection output by Ground Truth (GT) object annotations, a deep neural network can more easily learn the reasoning patterns required for prediction and for generalization.

In the line of recent work in AI explainability [29, 33], and data visualization [18, 39, 11], we propose an in-depth analysis of attention mechanisms in Transformer-based models and provide indications of the patterns of reasoning employed by models of different strengths. We visualize different operating modes of attention and link them to different sub-tasks (“functions”) required for solving VQA. In particular, we use this analysis for a comparison between oracle models and standard models processing noisy and uncertain visual input, highlighting the presence of reasoning patterns in the former and less so in the latter.

Drawing conclusions from this analysis, we propose to fine-tune the perfectly-sighted oracle model on the real noisy visual input (see Fig. 1). Using the same analysis and visualization techniques, we show that attention modes absent from noisy models are transferred successfully from oracle models to deployable models, and we report improvements in overall accuracy and generalization.

Contributions — (i) An in-depth analysis of reasoning patterns at work in Transformer-based models, comparing Oracles vs. deployable models, including visualization of attention modes; an analysis of the relationships between attention modes and reasoning, and the impact of attention pruning on reasoning. (ii) We propose to transfer reasoning capabilities, learned by the oracle, to SOTA VQA methods with noisy input and improve overall performance and generalization on the GQA [19] dataset (iii) We show that this transfer is complementary with self-supervised large-scale pre-training (LXMERT [36]/BERT-like).

2. Related work

Visual Question Answering (VQA) — as a task was introduced in various datasets, such as VQA-v1 [5] and VQA-v2 [17] (built from human annotators), or CLEVR [20] and GQA [19] (automatically-generated from fully-synthetic and real-world images, respectively). Additional splits were proposed to evaluate specific reasoning capabilities. For instance, VQA-CP [1] explicitly inverts the answer distribution between train and test splits. Following recent critics and controversies about these evaluations [37, 34], the GQA-OOD dataset [22] introduced a new split of GQA focusing on rare (Out-Of-Distribution / OOD) question-answer pairs, and showed that many VQA models strongly rely on dataset biases. This growing amount of diverse datasets has been accompanied by the development of more sophisticated VQA models. While an exhaustive survey of methods is out of the scope of this paper, one can mention families based on object-level attention [3], bilinear fusion [23], tensor decomposition [6], neural-symbolic reasoning [42], neural [3] and meta [9] module networks.

Transformers and Vision-Language reasoning — In this work, we focus on Transformers [38] due to their wide adoption and their powerful attention mechanism. MCAN [43] and DFAF [15] introduced the use of object-level self-attention and co-attention mechanisms to model intra- and inter-modality interactions in VQA. More recent work [21, 10, 30, 36] suggests that the combination of Transformers with a large-scale BERT [12]-like pretraining can be beneficial for VQA. Self-attention on pixel-level [2, 13] is, but up to our knowledge, not used done for VQA.

Attention and reasoning patterns in Transformers — Analysis of self-attention mechanisms has received considerable attention recently. In [40], a visualization tool for analysing BERT attention layers is introduced. [11] studies how training strategies and fine-tuning impact attention in BERT-like models, Voita et al. [41] classifies BERT’s attention heads according to their functionality, reporting a significant simplification of the model’s complexity via pruning. Finally, [31, Appendix A.5] measures energy distributions and classifies them based on their meta-stable states.

Following this work in NLP, similar studies have appeared in vision and language. [21] explores the emergence of word-object alignment in the attention maps when adding a weakly supervised objective. [8, 27] study to what extent the attention maps in BERT-like pretrained VQA Transformers encode various vision-language information. While these methods provide a better understanding of the amount of information captured by VQA models, they do not shed light on how this information is used. In our work, we analyze how various VQA tasks are encoded in different attention heads. To this end, we apply an energy-based analysis inspired by [31]. In addition, we study attention in perfect-sighted oracle Transformers in order to identify which patterns lead to better reasoning. Our findings lead to an Oracle Transfer strategy, which allows to improve performance and generalization in standard transformer models. Finally, our work is related to [14], which found evidence for relationships between questions and the modulation of (non transformer) model parameters on the synthetic CLEVR [20] dataset.

3. Analysis of Reasoning Patterns

In this section we will analyze reasoning behavior in Transformer-based VQA models and make the case for the impact of training on visual GT data. First, to make the paper self-contained, we provide a short introduction into
Vision-Language (VL)-Transformers as proposed in standard literature [43, 15, 36, 30, 26, 10, 35].

Given two different modalities, Vision (V) and Language (L), VL-Transformers are composed of the succession of intra-modality $T^V_L(\cdot), T^L_L(\cdot)$ and inter-modality $T^V_{\L \\L}^V(\cdot), T^L_{\L \\V}(\cdot, \cdot)$ multi-head attention layers [38]. As defined in the seminal paper [38], multi-head attention layers $T(\cdot)$ (both intra- and inter-modality ones) can be expressed as a set of self-attention layers $t(\cdot)$ which are performed in parallel on several “heads”. For example, given an input sequence $v$ of the visual embeddings, a visual intra-modality $n$-head attention layer $T^V_L$ performs as a set of $h$ visual intra-modality self-attention layers \( \{ t^V_{(1)}(\cdot), \ldots, t^V_{(h)}(\cdot) \} \), the outputs of which are concatenated and then combined:

\[
T^V_L(v) = \left[ t^V_{(1)}(v), \ldots, t^V_{(h)}(v) \right] W^O
\]

where $W^O$ is a trainable matrix which is particular for each multi-head attention layer. Each layer $t(\cdot)$ is defined on a set of input (vision or language) embeddings $x$ of the same dimension $d$ as

\[
t_{\cdot}(x) = \sum_j \alpha_{ij} x^v_j,
\]

where the query $x^q$, key $x^k$ and value $x^v$ matrices are given as follows: $x^q=W^q x$, $x^k=W^k x$ and $x^v=W^v x$. All $W$ are trainable parameters. In particular, $x^q$ and $x^k$ are used to calculate the self-attention weights $\alpha_{ij}$ as follows:

\[
\alpha_{ij}=(\alpha_{ij}, \ldots, \alpha_{ij}, \ldots, \alpha_{nj})=\sigma(\frac{x^q_i x^k_j}{\sqrt{d}}, \ldots),
\]

with $\sigma$ being the softmax operator. In this paper, we mainly focus on the attention maps \( \{ \alpha_{ij} \} \) which are composed of these self-attention weights.

Finally, inter-modality self-attention layers $t_{\L \L}(\cdot, \cdot)$ are defined in the same way, as the intra-modality ones, but unlike the latter they calculate queries, keys and values on two sets of input embeddings of different modalities. More precisely, for the self-attention layer $t^V_{\L \L}(v, l)$, the query matrix $v^q = W^q v$ is calculated on vision embeddings, while the key $l^k = W^k l$ and the value matrices $l^v = W^v l$ are calculated on the language outputs. For the $t^L_{\L \V}(l, v)$ self-attention layer, the matrices are calculated symmetrically ($l^q = W^q l$, $l^k = W^k l$, and $l^v = W^v l$, respectively). In addition, each $t^V_{\L \L}$ (resp. $t^L_{\L \V}$) is followed by a self-attention $t^V$ (resp. $t^L$). A general view of the architecture is available in the supp. mat..

Experimental setup — All analyses in this section have been performed with a hidden embedding size $d = 128$ and a number of per-layer heads $h = 4$. This corresponds to a tiny version of the architecture used in LXMERT [36] where $d = 768$ and $h = 12$. Therefore, “tiny-LXMERT” corresponds to the VL-Transformer architecture plus BERT-like (LXMERT) pre-training. Unless specified otherwise, objects have been detected with Faster R-CNN [32]. Visualizations are done on QQA [19] (validation set) as it is particularly well suited for evaluating a large variety of reasoning skills. However, as QQA contains synthetic questions constructed from pre-defined templates, the dataset only offers a constrained VQA environment. Additional experiments might be required to extend our conclusions to more natural setups.

### 3.1. Visual noise vs. models with perfect-sight

We conjecture that difficulties in the computer vision pipeline are the main cause preventing VQA models in learning to reason well, and which leads them to exploit spurious biases in training data. Most of these methods use pre-trained off-the-shelf object detectors during training and evaluation steps. But in a significant number of cases, the visual objects necessary for reasoning are misclassified, or even not detected at all, as indicated by detection rates of SOTA detectors on the Visual Genome dataset [25], for instance. Under these circumstances, even a perfect VQA model is unable to predict correct answers without relying on statistical shortcuts.

To further explore this working hypothesis, we trained an oracle model with perfect sight, i.e a model which receives perfect visual input, and compare it with tiny-LXMERT. Based on the same VL-Transformer, it receives the GT objects from the QQA annotations, encoded as GT bounding boxes and 1-in-K encoded object classes replacing the visual embeddings of the classical model. All GT objects are fed to the model, not only objects required for reasoning. We study the capabilities of both models, the oracle model and the classical one, to “reason”. Following [22] we measure the reasoning capabilities of a VQA model as the capability to correctly answer questions, where the GT answer is rare w.r.t. the question group, i.e the type of questions being asked. We evaluate the models on the QQA-OOD benchmark [22] designed for OOD evaluation.

Fig. 2 illustrates the model behavior in different situations. At the extreme case (left side of the plot), the model is evaluated on the rarest samples only, while on the right side all samples are considered. We observe that the performance of the classical model taking noisy visual (tiny-LXMERT) drops sharply for (image, question) pairs with rare GT answers, which is an indication for a strong dependency on dataset biases. We would like to insist that in this benchmark the rarity of a GT answer is determined w.r.t. the question type, which allows to measure biases taking into account language. The oracle model, on the other hand, obtains performances which are far less dependent on answer rarity, providing evidence for its ability to overcome...
In this experiment, we study the relationships between attention modes and question types, which correspond to different functionalities in the VL-Transformer, where we observed a high diversity in attention modes. We also observed that some layers’ heads, especially those processing the visual modality ($t^V_{L-}$ or $t^V_{L+}$) are mainly working with close-to-average attention distributions (very large meta-stable states [31]). On the other hand, we observed smaller meta-stable states in the language layers ($t^L_{L-}$ or $t^L_{L+}$). This indicates that the reasoning process in the oracle VL-Transformer is in large part executed by the model as a transformation of the language features, which are successively contextualized (i.e. influenced) by the visual features (and not the opposite).

In contrast to the attention modes reported in [31], we also observed bi-modal $k$-number distributions, shown in Fig. 3-a, which are a combination of a dirac (Fig. 3-b) and uniform (cf Fig. 3-c) attention modes. We call these modes “bimorph” attention, since they reveal the existence of two different shapes of attention distribution: for some samples, a dirac activation is generated, while other samples lead to uniform attention (averaging over tokens). Besides, in Fig. 4, we compare attention mode diversity between the noisy visual model and the oracle $t^L_{L+}$ heads, where we observe higher diversity for the oracle. In particular, “bimorph” attention are mostly performed by the oracle.

### 3.3. Attention modes and task functions

In this experiment, we study the relationships between attention modes and question types, which correspond to different functionalities in the VL-Transformer, where we observed a high diversity in attention modes. We also observed that some layers’ heads, especially those processing the visual modality ($t^V_{L-}$ or $t^V_{L+}$) are mainly working with close-to-average attention distributions (very large meta-stable states [31]). On the other hand, we observed smaller meta-stable states in the language layers ($t^L_{L-}$ or $t^L_{L+}$). This indicates that the reasoning process in the oracle VL-Transformer is in large part executed by the model as a transformation of the language features, which are successively contextualized (i.e. influenced) by the visual features (and not the opposite).

In contrast to the attention modes reported in [31], we also observed bi-modal $k$-number distributions, shown in Fig. 3-a, which are a combination of a dirac (Fig. 3-b) and uniform (cf Fig. 3-c) attention modes. We call these modes “bimorph” attention, since they reveal the existence of two different shapes of attention distribution: for some samples, a dirac activation is generated, while other samples lead to uniform attention (averaging over tokens). Besides, in Fig. 4, we compare attention mode diversity between the noisy visual model and the oracle $t^L_{L+}$ heads, where we observe higher diversity for the oracle. In particular, “bimorph” attention are mostly performed by the oracle.

### 3.3. Attention modes and task functions

In this experiment, we study the relationships between attention modes and question types, which correspond to different functionalities in the VL-Transformer, where we observed a high diversity in attention modes. We also observed that some layers’ heads, especially those processing the visual modality ($t^V_{L-}$ or $t^V_{L+}$) are mainly working with close-to-average attention distributions (very large meta-stable states [31]). On the other hand, we observed smaller meta-stable states in the language layers ($t^L_{L-}$ or $t^L_{L+}$). This indicates that the reasoning process in the oracle VL-Transformer is in large part executed by the model as a transformation of the language features, which are successively contextualized (i.e. influenced) by the visual features (and not the opposite).

In contrast to the attention modes reported in [31], we also observed bi-modal $k$-number distributions, shown in Fig. 3-a, which are a combination of a dirac (Fig. 3-b) and uniform (cf Fig. 3-c) attention modes. We call these modes “bimorph” attention, since they reveal the existence of two different shapes of attention distribution: for some samples, a dirac activation is generated, while other samples lead to uniform attention (averaging over tokens). Besides, in Fig. 4, we compare attention mode diversity between the noisy visual model and the oracle $t^L_{L+}$ heads, where we observe higher diversity for the oracle. In particular, “bimorph” attention are mostly performed by the oracle.
different functions of reasoning required to solve the problem instance. In other words, we explore to what extent the neural model adapts its attention distribution to the question at hand. We group the set of questions according to functions using the GQA [19] annotation, using 54 different functions such as e.g. “filter color”, “verify size”, etc.\(^6\).

We link functions to the attention modes introduced in Section 3.2. In Fig. 5 we show functions in columns and a selection of attention heads in rows, while the color encodes the median \(k\)-number for the oracle model. We observe a strong dependency. Certain functions, e.g. the majority of the “choose X” functions, tend to cause the emergence of small meta-stable states. In these modes, the attention mechanism is fundamental as it allows the model to attend to specific token combinations by detecting specific patterns. On the other hand, some functions requiring to attend to very general image properties, such as “choose location” or “verify weather”, seem to be connected to very large meta-stable states. We conjecture, that to find general scene properties, a large context is needed. In this mode, the attention mechanism is less important, and replacing it with a simple averaging operation is likely to keep performance — an experiment we explore in Section 3.4. Similarly, when focusing on heads instead of functions, we observe that a majority of heads typed as \(t_{X-L}^{L}()\) or \(t_{X}^{L-L}()\) tends to behave independently of the question functions and they generally show close-to-uniform attention. On the other hand, the \(t_{L}^{L}()\) and \(t_{X-L}^{L-V}()\) heads are highly dependent on the question functions. As shown in Figs. 5 and 6, these heads do not behave in the same way and are not “activated” (i.e. have a smaller metastable-state) for the same combination of functions. This provides some evidence for modularity of the oracle VL-Transformer, each attention head learning to specialize to one or more functions. In addition, in Fig. 6, we visualize the difference in oracle attention modes between two different function configurations: Fig. 6-a is the distribution of median \(k\)-numbers over all samples, i.e. involving all functions, whereas Fig. 6-b shows the distribution over samples involving the “choose color” function. We show the 3rd \(T_{X}^{L-V}L\) Transformer layer heads. Over all functions, these heads show “bimorph” behavior, whereas on questions requiring to choose a color, these same heads show either dirac or uniform behavior.

Oracle vs. Noisy Input — In the next experiment, we explore the difference in behavior between the perfect-sighted oracle and the classical model taking noisy visual input. For each input sample, we create a 80-dimensional representation describing the attention behavior of the model by collecting the \(k\)-numbers of the 80 cross-attention heads into a flat vector, taking the median over the tokens for a given head. Fig. 7 shows two different t-SNE projections of these attention behavior space, one for the oracle model and one for the noisy model. While the former produces clusters regrouping functions according to their general type, the function representation of the noisy model is significantly more entangled. We conjecture, that the attention-function relationship provides insights into the reasoning strategies of the model. VQA requires to handle a large variety of reasoning skills and different operations on the input objects and words. Question-specific manipulation of words and objects is essential for correct reasoning. In contrast to the oracle one, the t-SNE plot for the noisy visual model paints a muddier picture, and does not show clear relationships between attention modes and functions.

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

\(^6\)There is limited overlap between functions, e.g. “filter” contains, among others, the “filter color” and “filter size”.

![Figure 5. Attention modes for selected attention heads (rows) related to functions required to be solved to answer a question (columns). The head’s notation \(x, i, j\) refers to the head \(j\) of the \(i\)th Transformer layer of type \(x\): ‘lang’/‘ll’=\(t_{X-L}^{L}()\), ‘vis’/‘vv’=\(t_{X}^{L-V}()\), ‘vl’=\(t_{L}^{L-V}()\), ‘v’=\(t_{X-V}^{L}()\). The VL-Transformer’s architecture is presented in supp. mat.. The color encodes the attention mode, i.e. median of the \(k\)-number [31]. We observe (1) attention heads behave differently depending on the function; (2) a given function causes different attention modes for different heads.](image1)

![Figure 6. Influence of the question on oracle’s “bimorph” attention heads. We compare attention modes of the third layer of \(T_{X}^{L-V}L\) heads as a distribution of the \(k\)-numbers [31] over (a) samples of all functions, and (b) samples with questions involving the “choose color” function, and observe a clear difference. The function “choose color” seems to cause the activation (i.e. emergence of a small meta-stable state) of the \(1^{st}\), \(2^{nd}\) and \(4^{th}\) head, and the desactivation of the \(3^{rd}\) one, further indicating task dependence of attention head behavior.](image2)


## 3.4. Attention pruning

We further analyze the role of attention heads by evaluating the effect of pruning heads on model performance. As reported by [41, 31], specific attention heads may be useful during training, but less useful after training. In the same lines, for specific heads we replace the query-key attention map by a uniform one, “pruned” heads will therefore simply contextualize each token by an averaged representation of all other tokens, as a head with large meta-stable state would have done. In Table 1 we report the effect of pruning on GQA validation accuracy according to different attention categories and observe that the oracle model is resilient to pruning of the $t^v_v$ (·) and $t^v_{v←L}$ (·) heads, but that pruning of $t^L_v$ (·) and $t^L_{v←V}$ (·) heads results in sharp drops in performance. This indicates that the bulk of reasoning occurs over the language tokens and embeddings, which are contextualized from the visual information through $t^L_{v←V}$ (·) cross-attention. We can only conjecture why this solution emerges after training — we think that among reasons are the deep structure of language and the fact that in current models the answer is predicted from the CLS language token.

### Impact on functions

We study the impact of pruning on the different task functions by randomly pruning $n$ cross-attention heads and measuring accuracy for different function groups, $n$ being varied between 0% (no pruning) to 100% (all heads are pruned), as shown in Fig. 8 for the oracle and noisy vision-based models. For the sake of clarity only 4 different function are shown, additional results are provided in supplementary material. For the perfect-sighted oracle (Fig. 8-a), we first observe that the pruning has a different impact depending on the function. Thereby, while filter and choose are dominated by negative curvature where performance drops only when a large number of heads are pruned, verify and and, are characterized by a sharp inflection point and an early steep drop in performance. This indicates that the model has learned to handle functions specifically, resulting in various degrees of reasoning distribution over attention heads. For the noisy vision-based model, on the other hand, the effect of head pruning seems to be unrelated to the function type (Fig. 8-b).

### 3.5. Interactive visualization

The analysis described above was based on integrating information of various kinds over a full dataset, GQA-validation. Additional insights can be gained by exploring the behavior of individual problem instances and relating them to statistics extracted from the population, in particular attention modes and groups of functions. We have performed this analysis on a large number of samples and we provide a tool, which allows the reader to perform similar experiments online, making it possible to load different oracle or noisy vision based models. This tool is available at [1] (online experience + source code). Fig. 9 gives a simple visualization (see Section 4 for a discussion), a video of its usage is provided in the supplementary material.

**Discussion** — The experiments of this section have shown a pronounced difference in attention modes between the perfectly-sighted oracle and a noisy vision based model. More importantly, the oracle model shows a strong relationship between attention mode and task function, which we interpret as the capability of adapting reasoning to the task.

---

Table 1. Impact of pruning different types of attention heads of the trained oracle model.

<table>
<thead>
<tr>
<th>Pruned attentions</th>
<th>n/a</th>
<th>L</th>
<th>V</th>
<th>L→V</th>
<th>V→L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>91.5</td>
<td>37.9</td>
<td>91.4</td>
<td>52.8</td>
<td>68.1</td>
</tr>
</tbody>
</table>

Figure 8. Impact of random pruning of varying numbers of attention heads in cross-modal layers on GQA-validation accuracy. (a) For the oracle, the impact is related to the nature of the function, highlighting its modular property. (b) For the noisy-vision-based model, pruning seems to be unrelated to function types.

Figure 7. t-SNE projection of the attention mode space, i.e. the 80-dim representation median k-numbers, one per head of the model. Colors are functions, also provided as overlaid text. We compare projections of (a) the oracle, and (b) the noisy visual model, and observe a clustering of functions in the attention space for the oracle, but significantly less for the noisy input model.
at hand. The classical model significantly lacks this abilities, suggesting a strategy of transferring patterns of reasoning from an oracle model pre-trained on visual GT to a model taking noisy visual input.

4. Transferring Reasoning Patterns

We propose Oracle Transfer, transferring reasoning patterns from a perfectly-sighted model to a deployable model taking noisy visual inputs. We argue, that the first optimization steps are crucial for the emergence of specific attention modes. Training proceeds as follows (see Fig. 1):

1. Training of a perfectly-sighted oracle model on GT visual inputs from the GQA [19] annotations, in particular a symbolic representation concatenating the 1-in-K encoded object class and attributes of each object.

2. Initialize a new model with the oracle parameters. This new model is taking noisy visual input in a form of the dense representation (2048-dim feature vector extracted by Faster-RCNN [32] fused with bounding-boxes). The first visual layers ($T^V_x$) are initialized randomly due to the difference in nature between dense and symbolic representations.

3. Optionally and complementary, continue training with large-scale self-supervised objectives (LXMERT [36]/BERT-like) on combined data from Visual Genome [25], MS COCO [28], VQAv2 [17].

4. Fine-tune with the standard VQA classification objective on the target dataset (GQA [19] or VQAv2 [17]).

Experimental setup — We use the same VL-Transformer architecture defined in Section 3 (more details in supp. mat.), with $d=128$ and $h=4$, which corresponds to a tiny version of LXMERT [36] architecture. Following [36], we use 36 objects per image. We evaluate on the GQA [19], GQA-OOD [22] and VQAv2 [17] datasets.

Evaluating transfer — We evaluate the impact of Oracle Transfer on three different benchmarks in Table 2, observing that transferring knowledge from the oracle significantly boosts accuracy. We also evaluate the effect of Oracle Transfer on bias reduction and benchmark on GQA-OOD [22], reporting gains in Out-Of-Distribution settings — rare samples, “acc-tail” — by a large margin, which suggests improved generalization ability. Our experiments show that Oracle Transfer is complementary to large-scale vision-language self-supervised objectives of type LXMERT/BERT-like pretraining as introduced in [36]. An overall gain of about +1 accuracy points is observed from models (c) to (d) in Table 2, attributed to Oracle Transfer. As a comparison, LXMERT/BERT pretraining alone does not improve “acc-tail” on GQA-OOD.

Cross-dataset training — We explore whether the effects of oracle knowledge generalize beyond the GQA dataset, and evaluate training the oracle on GQA GT annotations, performing LXMERT/BERT pretraining, and transferring to a model trained on VQAv2 dataset [17]. We improve VQAv2 accuracy by a significant margin, suggesting positive transfer beyond GQA (Table 2).

Transfer ablation studies — We evaluate different variants of knowledge transfer, shown in Table 3, on the GQA validation set only. We explore a direct transfer from the
oracle to a deployable model without retraining, by making visual input representations comparable. To this end, the deployable model receives 1-in-K encoded class information, albeit not from GT classes but taking classes from the Faster R-CNN detector (Table 3-b). While inferior to the baseline, its performance is surprisingly high, suggesting that the oracle learns knowledge which is applicable in real/noisy settings. Performance gains are, however, only obtained by finetuning the model to the uncertainties in dense visual embeddings. Retraining only the visual block (Table 3-c), performances are on par with the baseline, retraining the full model (Table 3-d) gains +4.6p.

Comparison with SOTA — Oracle Transfer allows to improve performance of the tiny-LXMERT model both in and out of distribution [22] (Table 4, bottom part). Transfer is parameter efficient and achieves on-par overall accuracy with MCAN-6 [23] while halving capacity.

Qualitative analysis & interpretability — Finally, we qualitatively study the effects of Oracle Transfer and interpretability of attention heads. As shown in Fig. 10, after transfer, the VL-Transformer preserves the “bimorph” property of its attention heads, which was present in the original oracle model (Fig. 4-a), but absent in the baseline (Fig. 4-b). In addition, Fig. 9 shows the attention maps of the $T^X$ and $T^V$ heads in the second cross-modal layer for an instance. This head, referenced as $T^X \rightarrow V$, is observed to be triggered to questions such as “verify attr” and “verify color” provided as example. We observe that the oracle model draws attention towards the object “fork” in the image, and also, to a lesser extend, in the transferred model, but not in the baseline model. Similar attention patterns were observed on multiple heads in the corresponding cross-modal layer — this analysis took into account possible permutations of heads between models. Interestingly the miss-classification as a “knife” prevents the baseline from drawing attention to it, but not the transferred model.

5. Conclusion

We have provided a deep analysis and visualizations of several aspects of deep VQA models linked to reasoning on the GQA dataset. We have shown, that oracle models produce significantly better results on questions with rare GT answers than models on noisy data, that their attention modes are more diverse and that they are significantly more dependent on questions. We have also performed instance level analysis and we propose a tool available online1, which allows to visualize attention distributions and modes, and their links to task functions and dataset wide statistics.

Drawing conclusions from this analysis, we have shown that reasoning patterns can be partially transferred from oracle models to SOTA VQA models based on Transformers and BERT-like pre-training. The accuracy gained from the transfer is particularly high on questions with rare GT-answers, suggesting that the knowledge transferred is related to reasoning, as opposed to bias exploitation.

Acknowledgements — C. Wolf acknowledges support from ANR through grant “Remember” (ANR-20-CHIA-0018).

---

Table 2. Quantitative evaluation of the proposed knowledge transfer from oracle models. All listed models are deployable, no GT input is used for testing. Models: (c)+(d) are pre-trained with LXMERT [36]/BERT-like objectives after Oracle Transfer. All scores GQA-OOD-testdev [22]; GQA [19]-testdev; VQAv2-test-std [17]. Training hyperparameters selected on respective validation sets.

<table>
<thead>
<tr>
<th>Method</th>
<th>Input train</th>
<th>Input test</th>
<th>Acc.</th>
<th>GQA-OOD</th>
<th>GQA</th>
<th>VQAv2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Baseline</td>
<td>Dense</td>
<td>Dense</td>
<td>61.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(b) Transf. w/o retrain</td>
<td>1-in-K GT</td>
<td>1-in-K pred.</td>
<td>58.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(c) Transf. w/ $T^V$ retrain</td>
<td>1-in-K GT</td>
<td>Dense</td>
<td>61.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(d) Transf. w/ retrain</td>
<td>1-in-K GT</td>
<td>Dense</td>
<td>66.3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Impact of different types of transfer, GQA [19] val. accuracy. All models are deployable (no GT used for testing).

<table>
<thead>
<tr>
<th>Method</th>
<th>Input train</th>
<th>Input test</th>
<th>Acc.</th>
<th>$\Theta$</th>
<th>O</th>
<th>L</th>
<th>OOD</th>
<th>GQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUTD [3]</td>
<td>22</td>
<td>42.1</td>
<td>51.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAN-4 [23]</td>
<td>50</td>
<td>47.2</td>
<td>54.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCAN-6 [43]</td>
<td>52</td>
<td>46.5</td>
<td>56.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>26 ✓</td>
<td>48.5</td>
<td>56.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LXMERT-tiny</td>
<td>26 ✓</td>
<td>47.5</td>
<td>56.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LXMERT-tiny + Ours</td>
<td>26 ✓ ✓</td>
<td>48.3</td>
<td>57.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LXMERT [36]</td>
<td>212 ✓</td>
<td>49.8</td>
<td>59.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 = number of parameters (M); OOD = GQA-OOD [22] Acc-tail. O = Oracle Transfer, L = LXMERT/BERT pretraining.

Table 4. Comparison with SOTA on GQA and GQA-OOD on test-dev. Hyperparameters were optimized on GQA-validation.
References


[22] Corentin Kervadec, Grigory Antipov, Moez Baccouche, and Christian Wolf. Roses are red, violets are blue... but should vqa expect them to? In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 1, 2, 3, 4, 7, 8


A. Interactive visualization of reasoning patterns

In the video added to this supplementary material, we provide a detailed analysis of the differences in attention modes for a given instance between the oracle model, the noisy baseline, and the oracle transfer model, providing indications for computer vision being the bottleneck in learning, and showing how patterns of attention are adapted by the transfer.

This video has been shot as a commented screencast of an interactive application we made available online at https://reasoningpatterns.github.io. The tool has been designed to explore the behavior of individual problem instances. We hope it will help scientists to better understand attention mechanisms at work in VL-Transformers of VQA models.

In addition, we provide another example of the differences in attention in Fig 11.

B. Is there a confounding factor in GQA-OOD

Could there be a confounding factor, in which the rare answers involve objects that are simply more difficult to recognize? Rare objects certainly had fewer visual examples for training the visual recognition models, and/or could be generally smaller or more occluded, for example. We evaluated this by comparing the performance of the object detector for two different sets (in Table 5): (1) Objects required to answer questions w/ rare GT answers (tail); and (2) objects required to answer questions w/ frequent GT answers (head). We report similar performance, and hence no evidence supporting the hypothesis of a confounder.

C. Additional visualizations

Attention pruning — We provide additional plots of the impact of attention pruning on performances, structured according to task functions, in Fig 13. Functions are gathered according to their general type, e.g. all questions involving to “filter something” (filter color, filter material, etc) are grouped as filter function. We still observe a significant difference between oracle and noisy visual models.

Attention distribution — To give a better insight of the differences in attention modes between oracle and noisy

D. Technical details

VL-Transformer architecture — More information about the VL-Transformer architecture can be found in the schematic illustration drawn in Fig 12. All notations and abbreviations are summarized in Table 6. The chosen architecture is similar to LXMERT [36].

Training details — All models were trained with the Adam optimizer [24], a learning rate of $10^{-4}$ with warm starting and learning rate decay. Training was done on one P100 GPU. Two P100 GPUs were used for BERT/LXMERT [36] pre-training. For the oracle, the batch size was equal to 256. We train during 40 epochs and select the best epoch using accuracy on validation. The oracle transfer follows exactly the same procedure, except when using LXMERT pretraining.

In that case, BERT/LXMERT [36] pretraining is performed during 20 epochs max with a batch size of 512. All pretraining losses are added from the beginning, including the VQA one. Note that LXMERT [36] is originally pre-trained on a corpus gathering images and sentences from MSCOCO [28] and VisualGenome [25]. As the GQA dataset is built upon VisualGenome, the original LXMERT pre-training dataset contains samples from the GQA validation split. Therefore, we removed these validation samples from the pre-training corpus, in order to be able to validate on the GQA validation split.

After pre-training, we finetune either on GQA [19] or VQAv2 [17]. For GQA, we finetune during 4 epochs, with a batch size of 32 and a learning rate equal to $10^{-5}$. For

### Table 5. Are there confounding factors? We report R-CNN recall (R) on objects required for reasoning w/ various IoU thresholds.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Notation</th>
<th>Abbrev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head</td>
<td>$T^L_\times$ ($\cdot$)</td>
<td>lang, i, j</td>
</tr>
<tr>
<td>Head</td>
<td>$T^V_\times$ ($\cdot$)</td>
<td>vis, i, j</td>
</tr>
<tr>
<td>Cross-modal</td>
<td>$V \leftarrow V$</td>
<td>$T^V_{\times}L$ ($\cdot$)</td>
</tr>
<tr>
<td>Cross-modal</td>
<td>$V \leftarrow V$</td>
<td>$T^V_\times$ ($\cdot$)</td>
</tr>
<tr>
<td>Cross-modal</td>
<td>$V \leftarrow V$</td>
<td>$T^V_{\times}L$ ($\cdot$)</td>
</tr>
<tr>
<td>Cross-modal</td>
<td>$V \leftarrow V$</td>
<td>$T^V_\times$ ($\cdot$)</td>
</tr>
</tbody>
</table>

### Table 6. VL-Transformer architecture and notation. A schematic view of the transformer is given in Fig 12.
“Does the bench look brown and wooden?”

Figure 11. Example for the difference in attention in the first vision to language layer. The oracle drives attention towards a specific object, “bench”, also seen after transfer but not in the baseline (we checked for permutations). This analysis was performed with our interactive visualization tool, which also allows to visualize attention models, not shown here (https://reasoningpatterns.github.io)

Figure 12. Schematic illustration of the VL-Transformer architecture used in the paper. It is composed of: (a) 9 × language only self-attention layers (b) 5 × vision only self-attention layers; (c) 5 × bi-directional vision-language cross-attention layers. Crossed arrows symbolize the cross modal information flow. Small rectangles illustrate the individual attention heads. Notations and abbreviations are summarized in Table 6.

VQAv2, we finetune during 8 epochs, with a batch size of 32 and a learning rate equal to $10^{-5}$. 
Figure 13. Impact of random pruning of varying numbers of attention heads in cross-modal layers on GQA-validation accuracy. (a) For the oracle, the impact is related to the nature of the function, highlighting its modular property. We plot the mean and standard deviation for each function.
Figure 14. Comparison of k-distribution of VL-attention heads for two different models for the function choose color: (a) oracle (4 first rows); (b) noisy visual input (4 last rows). Heads are colored according to their $k$-number median. As a recall, for each head we plot the distribution of the number $k$ of tokens required to reach 90% of the attention energy (GQA-val). The x-axis represents in % the number of tokens $k$ relatively to the total number of token, it goes from 0% to 100%.