

How Transferable are Reasoning Patterns in VQA?

Supplementary Material

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

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

Table 1. Are there confounding factors? We report R-CNN recall (R) on objects required for reasoning w/ various IoU thresholds.

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 which 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 1.

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 1): (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.

Layer		Notation	Abbrev.
9×Language only	$L \leftarrow L$	$T_-^L(\cdot)$	$lang, i, j$
5×Vision only	$V \leftarrow V$	$T_-^V(\cdot)$	vis, i, j
5×Cross-modal	$L \leftarrow V$	$\begin{cases} T_{\times}^{L \leftarrow V}(\cdot) \\ T_-^L(\cdot) \end{cases}$	$\begin{cases} vl, i, j \\ ll, i, j \end{cases}$
		$\begin{cases} T_{\times}^{V \leftarrow L}(\cdot) \\ T_-^V(\cdot) \end{cases}$	$\begin{cases} lv, i, j \\ vv, i, j \end{cases}$

Table 2. VL-Transformer architecture and notation. A schematic view of the transformer is given in Fig 2.

C. Additional visualizations

Attention pruning — We provide additional plots of the impact of attention pruning on performances, structured according to task functions, in Fig 3. 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. In particular, for the oracle, the *common* function is highly impacted by pruning. We later found that the $t_-^L(\cdot)$ heads at works in cross modal layers were essential for this function.

Attention distribution — To give a better insight of the differences in attention modes between oracle and noisy visual models, we provide more visualizations in Fig 4. These plots are measured on the $t_{\times}^{L \leftarrow V}(\cdot)$ attention heads for questions involving to choose a color. We recommend the reader to compare Fig 4 with Fig 4 in the main paper, to better understand the influence of the *choose color* function. We observe that the oracle’s attention heads are dependant on the functions involved in the question. In particular, the bi-morph heads become either *dirac* or *uniform* depending on the function. In contrast, the attention heads of the noisy visual model remain identical regardless of the function.

D. Technical details

VL-Transformer architecture — More information about the VL-Transformer architecture can be found in the

“Does the bench look brown and wooden?”

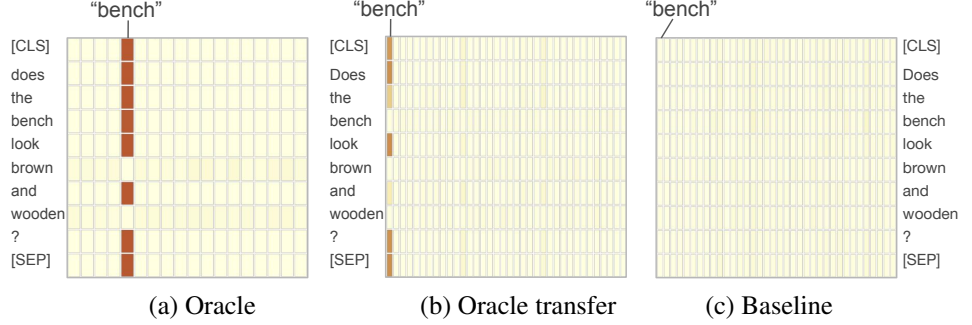


Figure 1. Example for the difference in attention in the first vision to language layer. The oracle drives attention towards a specific object, “bench”, also seen after transfer but not in the baseline (we checked for permutations). This analysis was performed with our interactive visualization tool, which also allows to visualize attention models, not shown here (<https://reasoningpatterns.github.io>)

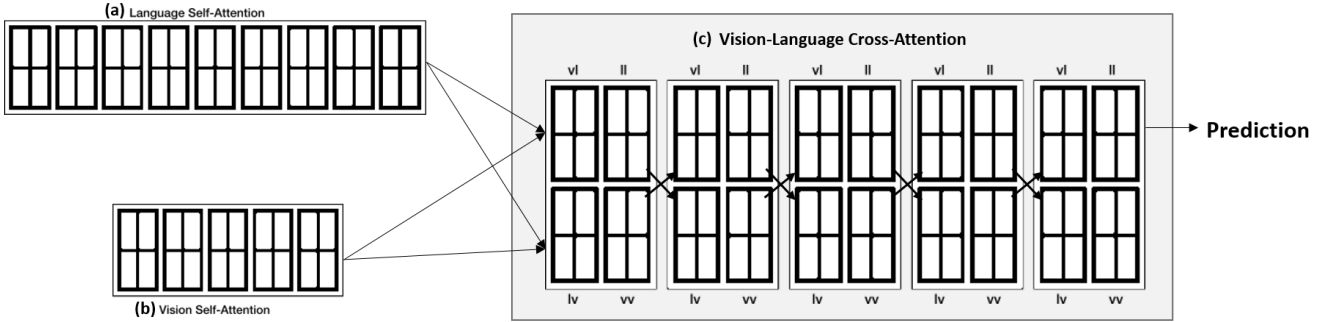


Figure 2. Schematic illustration of the VL-Transformer architecture used in the paper. It is composed of: (a) $9 \times$ language only self-attention layers (b) $5 \times$ vision only self-attention layers; (c) $5 \times$ bi-directional vision-language cross-attention layers. Crossed arrows symbolize the cross modal information flow. Small rectangles illustrate the individual attention heads. Notations and abbreviations are summarized in Table 2.

schematic illustration drawn in Fig 2. All notations and abbreviations are summarized in Table 2. The chosen architecture is similar to LXMERT [6].

Training details — All models were trained with the Adam optimizer [3], 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 [6] 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 [6] 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 [6] is originally pre-trained on a corpus gathering images and sentences from MSCOCO [5] and VisualGenome [4]. 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 [2] or VQAv2 [1]. For GQA, we finetune during 4 epochs, with a batch size of 32 and a learning rate equal to 10^{-5} . For VQAv2, we finetune during 8 epochs, with a batch size of 32 and a learning rate equal to 10^{-5} .

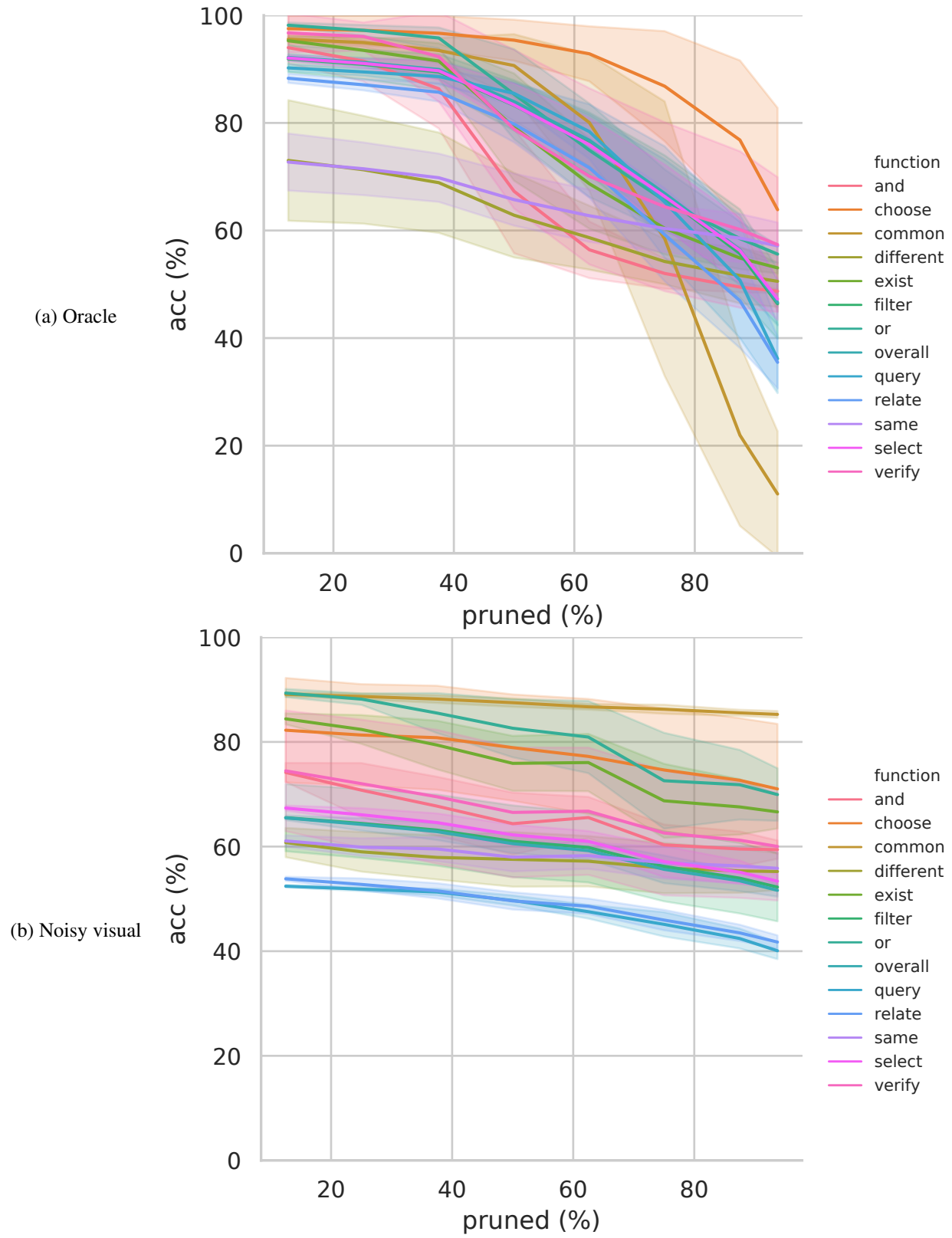


Figure 3. Impact of random pruning of varying numbers of attention heads in cross-modal layers on GQA-validation accuracy. (a) For the oracle, the impact is related to the nature of the function, highlighting its modular property. We plot the mean and standard deviation for each function.

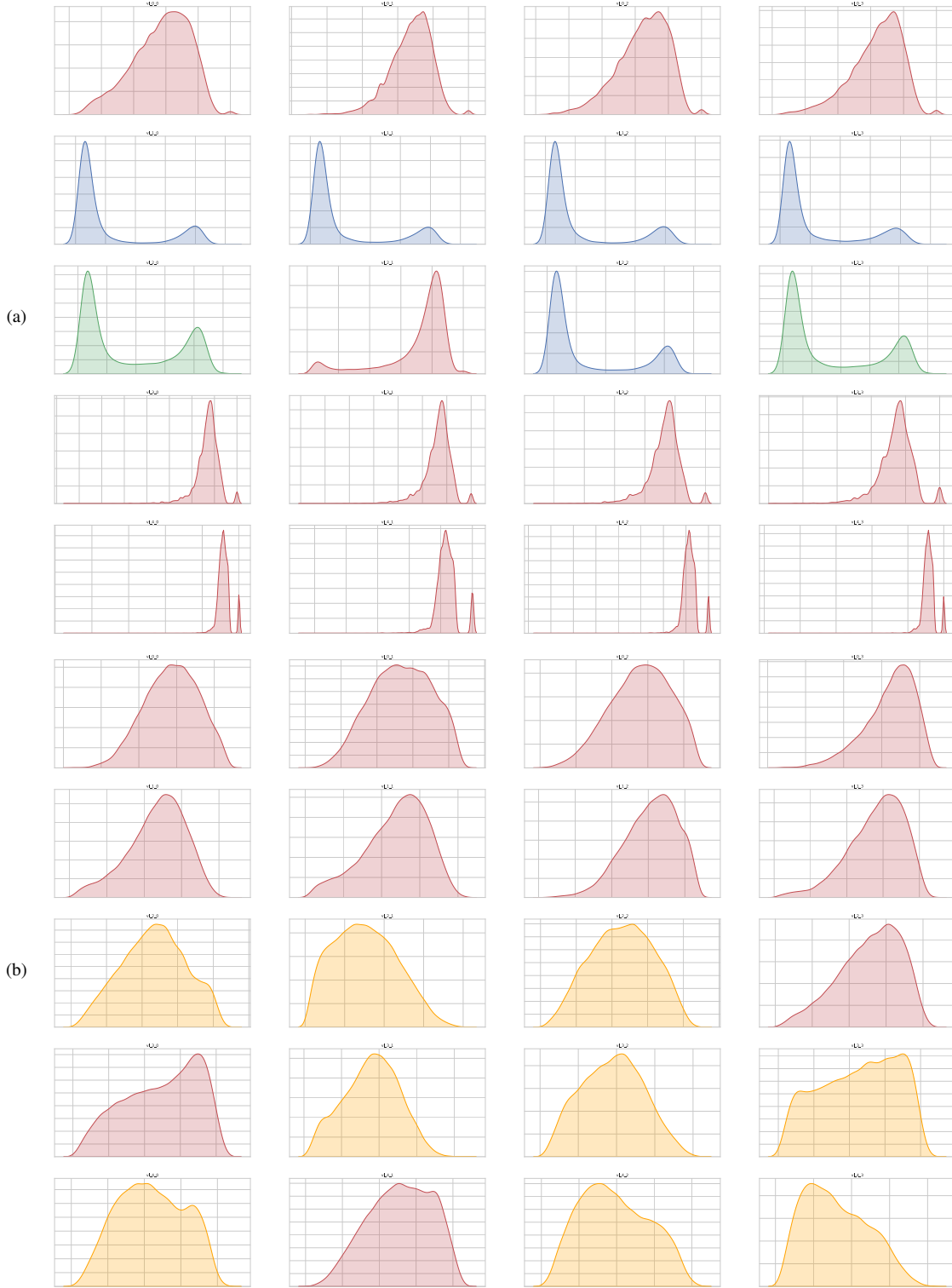


Figure 4. Comparison of k-distribution of VL-attention heads for two different models for the function *choose_color*: (a) oracle (4 first rows); (b) noisy visual input (4 last rows). Heads are colored according to their k -number median. As a recall, for each head we plot the distribution of the number 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%.

References

- [1] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913, 2017. 2
- [2] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6700–6709, 2019. 2
- [3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2014. 2
- [4] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David 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. 2
- [5] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2
- [6] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5103–5114, 2019. 2