

Detection of parametric or contextual knowledge: are the schemas similar across modalities?

Binesh Arakkal Remesh^{1,2}, Thomas Gerald¹, Camille Guinaudeau¹, Sahar Ghannay¹,
Eva Feillet¹,

¹ CNRS, Université Paris-Saclay, LISN ² Université de Lorraine,

Correspondence: firstname.lastname@lisn.upsaclay.fr

Abstract

In LLMs, the source of information used to produce an answer is necessary to explain the generation process. This work considers the case where the answer can be extracted from the question’s context (*contextual knowledge*) or from the model weights (*parametric knowledge*). While methods have been developed to uncover the grounding sources of LLMs in question answering, these approaches mainly rely on the text modality, whereas most tasks involve multiple modalities. This study proposes to uncover LLMs’ internal signals to determine whether they use parametric or contextual knowledge in a bimodal setting. To this end, we create a counterfactual dataset to evaluate whether the model relies on parametric or contextual knowledge on text and/or image-based questions. We leverage probing methods based on the internal attention and the different latent representations to determine the model’s source of information. We evaluate the performances of the probing methods, focusing on the different modalities. We show that models share common signals across the corpus and modalities. This study opens the way to develop controllable methods by uncovering common signals in LLMs to determine the knowledge used in generation.

1 Introduction

The rapid progress of large language models (LLMs) in question answering has enabled their widespread adoption as intelligent teaching assistants in education and other application domains (Kweon et al., 2025). The information used to answer the question is crucial for the interpretation and reliability of the generation. In this work, we consider two main sources of information that the model can use to answer a question. *Parametric knowledge*, when the information used to answer is extracted from the model weights, i.e., acquired during the training phase, and *contextual informa-*

tion, i.e., when the model answer is derived from the information given along with the question.

Understanding the source of information for the generation in question-answering tasks can help to improve the reliability and truthfulness of the generated answers. It will be helpful in detecting hallucination (where the model response is not rooted in the given contextual information) (Radevski et al., 2025), or provide clues to prioritize the generation according to the task or the user preferences. This paper proposes uncovering LLMs’ hidden signals to determine whether models rely on parametric or contextual knowledge. We design a probing method based on internal attention and on latent representations across layers to determine the model’s source of information. To train our probes, we developed a corpus of closed-ended counterfactual contexts with counterfactual answers, based on the existing corpora: SciQ (Welbl et al., 2017) and TriviaQA (Joshi et al., 2017) for textual question answering; the VQA dataset proposed by (Gerald et al., 2026), and TextbookQA (Kembhavi et al., 2017) for visual question answering.

We also evaluate the relevance of the trained probes on the semi-open-ended QA dataset (Xie et al., 2024).

This study addresses the following research questions: Q1) To what extent can a probe trained on one modality be applied to another modality? Q2) What is the best probing method in this case? Q3) Where is the information useful to determine the source of information of the model located?

This work is organized as follows: previous work closely related to our study is described in Section 2. In Section 3, we present the methodology for creating the proposed counterfactual dataset and how we deduce annotations about the knowledge used by leveraging the model output. Next, we introduce the probing architecture and types in Section 4, and the experimental protocol in Section 5. We finally report and discuss the results in Section

6, and conclude in Section 7.

2 Related Work

Identifying the source of knowledge. A widely used approach for assessing the source of knowledge used for generation by the model is to provide counterfactual information and check whether the generation produces an answer extracted from the context (contextual knowledge) or not (parametric knowledge). These methods are seamlessly applied to benchmark the model behavior when provided information conflicts with their internal state. For instance, (Longpre et al., 2021) use this mechanism to examine whether a model’s answers are derived from its parametric knowledge or from the provided external context based on entity swaps and examining the generated answer. Closer to our work, (Zhao et al., 2024) and (Tighidet et al., 2024) use probing methods based on a counterfactual dataset. The former (Zhao et al., 2024) examines the residual stream of LLM under knowledge conflicts and uses probing methods to assess how the model’s representation changes when the source of the answer changes. The latter (Tighidet et al., 2024) proposes a similar methodology that focuses on the internal model representation across layers and components. Both works highlight that we can find in models’ hidden representations a signal that can be used to determine the source of information leveraged for generation, especially one that becomes stronger from mid-layer representations onward.

Our work extends these works by studying the methodology for Visual QA alongside textual QA and verifying the transferability of the trained classifier to another modality or unseen data type.

QA Datasets. In recent years, many QA datasets have been developed to benchmark methods and models. Usually, most of them were developed on textual modality in closed QA settings such as (Rajpurkar et al., 2016), Sciq (Welbl et al., 2017), Trivia QA (Joshi et al., 2017). Especially, these corpora are useful for checking knowledge source attribution, since the answer can be directly extracted from the context or limited to specific choices (MCQ). This led to a proposed counterfactual corpus, for instance, by replacing the truth entity/answer from the provided context in the NewsQASwap dataset (Longpre et al., 2021). Open QA datasets have also been shared by the community (Nguyen et al., 2016; Jin et al., 2019); however,

these datasets are more difficult to evaluate, due to the freedom of accepted answers. In addition, it is hard to locate specific parts used to answer the descriptive question from context. While these corpora focus only on textual data, most state-of-the-art models are built on at least the image and text modalities. Consequently, recent benchmarks consider both modalities for QA, for instance, Textbook QA (Kembhavi et al., 2017) and ScienceQA (Lu et al., 2022), which contain images in their context. Also, recent work (Gerald et al., 2026) has highlighted the importance of images and text (an excerpt from a course) for the educational domain.

3 Dataset Creation

Evaluating whether a model uses information from context or trained data requires a probing method that uses an annotated dataset constructed during model generation. To this end, we draw inspiration from previous methods that first construct a corpus of counterfactual knowledge and then observe the model’s predictions to determine whether they conflict with the provided information. Later, a classifier (a probe) will be trained on the latent representation or attention signal extracted at the generation of the answer.

As we are interested in the different modalities and the answer format (open/closed QA), we propose building counterfactual datasets for the different properties we are interested in. We propose two pipelines for counterfactual dataset construction, the first for text-only QA and the second for visual-based QA.

Text only dataset construction. For the textual counterfactual dataset, we start from existing corpora, namely Sciq and Trivia QA datasets. This dataset consists of high school text-based questions. The proposed pipeline is based on a few-shot approach that prompts the Qwen3-8B model, guided by predefined rules for counterfactual context creation. We are mainly considering five rules¹ in the prompting setting given the original context, the question and the answer: the new context should be fluent; the answer to the question should be different for the new context; the new context can be formed by replacing the entities; the new context should not contain the correct answer; the entities shared in the question and the context should remain untouched; if the entities cannot be changed

¹The prompt is reported in the appendix G

in the context, a negation can be added to the sentence containing the answer in the context. In this setting, the original question remains exactly the same. We collected 4555 examples following this process. Notice that we also developed a pipeline (see appendix B) leveraging an NER approach and a BERT model; nonetheless, the first results were not promising.

Visual-based dataset construction. Creating a counterfactual image QA corpus, was needed for the research. The pipeline to construct the corpus is shown in Figure 1. TextbookQA (Kembhavi et al., 2017) dataset and another resource dataset (Gerald et al., 2026) were used for creating a counterfactual image corpus. These datasets focused on educational question answering, primarily involving diagrams and charts. To ensure that the question meets our criteria for image independence, the question and its answers are modified before the pipeline. Notice that, for the resource proposed in (Gerald et al., 2026), we generated the questions and answers using an prompting and LLM as a judge from the images in the dataset. Details is provided in Appendix D. The study is conducted only on context-independent question, as one that can be answered correctly even without the image.

The pipeline consists of three main steps.

- **Step 1:** Using the Qwen3.5 9B model (Qwen Team, 2026) to generate instructions that can be used to change parts of the image, given an image, question and answer. The model was prompted to understand the image and the question and to suggest editing plans with respect to the question and answer.

From the given instructions the target entity in the image and the suggested replacement entity is identified.

- **Step 2:** Using a object identifying model (Minderer et al., 2023) and OCR model², the target object and it’s corresponding labels are identified
- **Step 3:** Using OpenCV³ the match found by the OCR model is erased and suggested replacement labels are added. Then, using the object retriever of (Schuhmann et al., 2022), the replacement object is retrieved and pasted on top of the Target object.

²<https://pypi.org/project/easyocr/>

³<https://pypi.org/project/opencv-python/>

Examples of the data created and the pipeline are shown in Appendix G. We collected 2419 samples applying the previous method.

Conflict-QA Dataset We also used a subset of the dataset defined in (Xie et al., 2024), which contains 7326 question-answer pairs.

Validation of created QA Corpus. To assess the quality of the created dataset, a random sample of 50 instances was manually evaluated. For the textual QA dataset, the main evaluation criteria were grammatical inconsistency in generation and the absence of a correct answer in the generated counterfactual context. Besides manual evaluation, an automatic verification step was enabled using Spacy on the TQA corpus created. For each instance in the counterfactual context, it checks whether the correct reference answer is present. For instance, when the original answer was present in the counterfactual context, that instance was removed from the corpus. As the corpus used contains only one-word answers as ground truth, this approach of exact token matching is an effective way to evaluate our corpus.

Likewise, for the Visual QA dataset, a random 50 samples were manually evaluated. The key criterion was to assess whether the original answer was absent in the counterfactual image. In the sample evaluated, the ground-truth answer was absent in the counterfactual images. In addition, the extracted target⁴ and replacement⁵ entity labels, were verified automatically that the target is the ground truth-answer and the replacement is not the ground-truth answer, using the same method as of TQA.

4 Probing approach for classification

Prior to training the probes, labels were assigned to the counterfactual dataset. Notice that in this works labels are dependent on the model. In the second step, we trained the probes on the representation collected during the labeling/generation phase.

4.1 Labeling the generation

TQA and VQA From the datasets created for TQA and VQA, we associated each sample with a label indicating the source of the model’s generated answer: contextual-based or parametric-based

⁴From the step 1 of the pipeline, target entities are the ground truth answer

⁵From the step 1 of the pipeline, replacement entities are the suggested replacement object by the model model

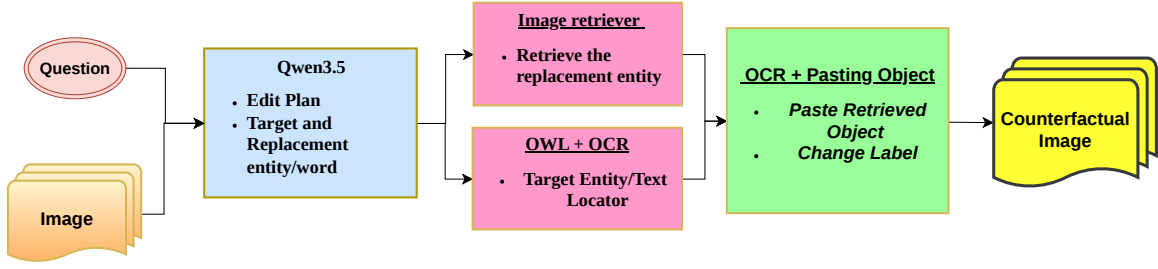


Figure 1: Image Substitution Pipeline

generation. The decision is made based on the generation of the correct answer, or the counterfactual answer provided in the context. Since our dataset contains primarily closed-ended questions with short answers, typically one or two words, an automatic evaluation based on approximate matching can be used to inspect the generation. To classify examples into these labels, we used a fuzzy-matching approach with Spacy-based tokenization. For the evaluation, generated answers and the reference answers are converted into tokens using Spacy. Later using the RapidFuzz library⁶ to classify the generated answers against the correct answer and the counterfactual answer. Based on the matching, three labels are assigned to the generation;

- **Original:** When the generated answer matches the correct answer. These examples were labeled with parametric-based generation
- **Counterfactual:** when the generated answer matches the Counterfactual answer. These examples were labeled with contextual-based generation
- **Other:** When the generated answer matches neither the counterfactual nor the correct answer. These examples were discarded.

ConflictQA. For conflictQA we followed the same validation of the generated answer used by (Zhao et al., 2024). We used the NLI model (Laurer et al., 2023) for checking the entailment between the generated answer and the reference answers. In order to avoid false positives, we check the entailment score of the generated answer against the correct and counterfactual reference answers provided in ConflictQA. We disregarded cases where the score assigned to the generated answer is similar for the correct reference and the counterfactual

reference —specifically, below the 0.3 threshold. Also the labels are only assigned to their corresponding labels if it have a entailment score more than 0.75. This ensure that the labels assigned are correct.

Final Probing Corpus For labeling, we used two models, Qwen3.5-9B and Qwen3.5-4B we applied on each of the three datasets. We also ensure balance between the two classes, by rejecting samples from the majority class until both class contains the same number of examples.

Detailed information on the sizes and numbers of training, validation, and test samples for each source dataset and model is provided in Table 1.

Data	Model	Size	Train/Valid/Test
TQA	Qn-9B	4400	3080 / 660 / 660
TQA	Qn-4B	3290	2303 / 493 / 493
VQA	Qn-9B	920	644 / 138 / 138
VQA	Qn-4B	1112	778 / 167 / 167
ConQA	Qn-9B	408	285 / 61 / 62
ConQA	Qn-4B	348	243 / 52 / 53
Multi	Qn-9B	1840	1288 / 276 / 276
Multi	Qn-4B	2224	1556 / 334 / 334

Table 1: Size of various corpus used to train the probe and their train, validation, and test size with respective to the different model. Here; **Qn** stands for Qwen3.5

For the different datasets, we use the following acronyms :

- **VQA:** Visual Question Answering dataset
- **TQA:** Textual Question Answering dataset
- **ConQA:** ConflictQA dataset
- **Multi:** Combination of TQA and VQA dataset with equal representation from both modality

⁶<https://pypi.org/project/RapidFuzz/>

4.2 Probe inputs

The probe is trained on different kinds of information extracted from the model during the labeling phase. The first is based on the attention, the second on hidden representation at different layers, and the third on the pre-LM-Head representation.

Context Attention. Preliminary experiments show that the average attention given to the input context across the query token, with respect to the prompt as well as the generated response, has a different attention pattern when generated answer contradicts the given context, illustrated in Figure 6. The figure and further details of the preliminary experiments are provided in the Appendix C.

Later, for the probing experiment, we created the Context Attention (CA) feature, which is based on the attention of the last input token associated with the context (not the question) for each layer. The attention for the last input tokens is normalized and summed across layers.

More formally, let $A^{l,h} \in \mathbb{R}^{N \times N}$ the attention matrix for h^{th} head at layer l (for the whole input sequence). First, the attention weights are averaged over the last input token:

$$A^l = \frac{1}{H} \sum_{h=1}^H A^{l,h} \quad (1)$$

The resulting attention distribution is then normalized and summed over the context tokens:

$$CA^l = \sum_{c \in C} \frac{A_{N,c}^l}{\sum_{j=1}^N A_{N,j}^l + \epsilon}, \quad (2)$$

With C the set of indices of context tokens. The input provided to the probe is then sequence containing the CA^l for each layer:

$$CA = [CA^1(x), CA^2(x), \dots, CA^L(x)]. \quad (3)$$

Hidden representation. We also use as input the hidden representation at each layer for the token C (last input/context token), that we will call e_C^l for the layer l and the hidden vector immediately preceding the language modeling head.

4.3 Probe architecture

CNN-MLP based. This architecture is designed to capture the sequential pattern in the model’s in-

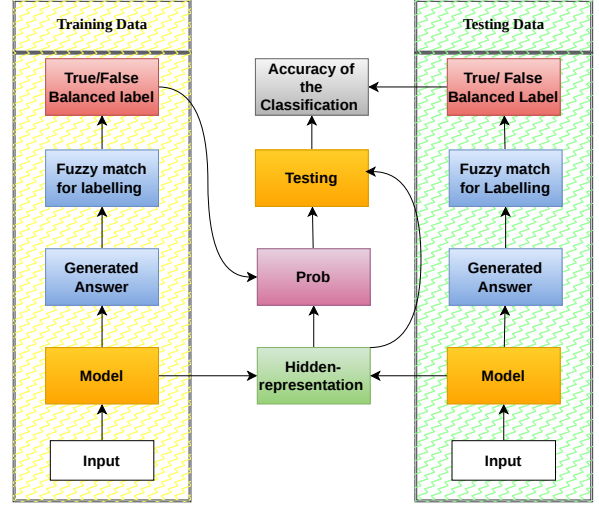


Figure 2: Probing architecture

put, ⁷. It is trained on the visual and textual modalities using CA feature across all self-attention layers, and on the output after the residual stream across all model layers. This architecture uses 3 CNN blocks and 3 MLP blocks (shown in Figure 7). The design of this architecture was selected based on preliminary results on CA features.

Linear Based prob. We also experiment with a simpler architecture, considering only linear classification on the models’ representation. This linear probing method is trained on the Visual and Textual QA dataset for binary classification of the two generation classes. The probe is trained on three different kind of input:

1. The hidden vector representation for the input of the LM-head.
2. All internal representations of all layers of the model after the residual stream is added (All layers individually as well as combined).
3. Score calculated by CA metrics on the self-attention layers (All calculated layers individually as well as combined).

To summarize the experimental pipeline, we illustrate the different steps for training and evaluating the probes in Figure 2.

5 Experimental Protocol

For the initial experiments and probing tasks, we use models from the Qwen3.5 series with different

⁷For example, the pattern captured by CAR metrics: the gradual increase and decrease in the CAR when the generated answer is correct, compared to the counterfactual context

sizes: Qwen3.5-9B, Qwen3.5-4B. For the dataset, we considered the corpus established in Table 1 paired with a specific model.

The probes were trained on the different corpora, stopping their training with no improvement on validation after 15 epochs; the best validation was selected for the evaluation on the test set. We used Adam as the optimizer with cross-entropy loss. The learning rate was chosen based on validation performance and depends on the dataset and the probe.

6 Experimental Results

For the convenience of the reader, we define the acronyms used in the different experiments in this first part. For the input features, we used the following acronyms:

- **In.R** : Internal Representation of each layer after the residual is added.
- **PLH**: Internal representation before the LM-Head module.
- **CA** : Context Attention score

The different probing methods are referred to :

- **LPL**: Linear probe for single layer
- **LPA**: Linear Probe when all layer given
- **CM** : Architecture given the whole layer input shown in the Figure 7

6.1 Probing performances

Probing internal representation. Table 2 presents the probing accuracy obtained from the internal representations of the Qwen-9B model⁸. The evaluation covers the different probe architectures and settings across multiple dataset modalities, including VQA, TQA, and ConflictQA.

From these results, we can see that training the probe in a unimodal setting yields better performance, indicating that introducing both modalities leads to a noisier signal.

On textual data, the probes perform very well, achieving accuracies above 90% across all probing methods. However, for visual-based questions, the results are mitigated, reaching up to 85% accuracy.

Visual modality probing performance largely depends on the probing methodology and, more

specifically, on the input representation. For unimodal, relatively better performance is seen when trained layer-wise, than when trained on the whole layer. This can be explained by Figure 3, where the first few layers are giving almost random guesses, whereas the last and intermediate layers are increasing the accuracy. So, combining all the layers might introduce noise, thereby degrading the model’s performance.

On the ConflictQA dataset, the performances are very close to TQA, suggesting that the type of question (open-ended or closed-ended) does not impact classification. Furthermore, it could signal that both probes (TQA and conflictQA) share the same features for classification.

The difference in accuracy between modalities highlights that the embeddings used for parametric vs. contextual knowledge detection differ, indicating that modality has a significant impact on internal model representations.

QMT	Input	Probe	Accuracy	L
VQA	In.R	LPL	85.51 %	21
		LPA	73.19 %	-
		CM	78.99%	-
	PLH	LP	77.54 %	-
TQA	In.R	LPL	93.69 %	29
		LPA	90.91 %	-
		CM	92.96 %	-
	PLH	LP	91.06%	-
ConQA	In.R	LPL	91.80%	26
		LPA	93.55%	-
		CM	95.16%	-
	PLH	LP	85.48%	-
Multi	In.R	LPL	78.26 %	20
		LPA	82.61%	-
		CM	84.42 %	-
	PLH	LP	81.52 %	-

Table 2: Results on probing the Qwen3.5-9B model on it’s internal representation of each layer.

Probing on attention Even though the initial experiment, which measured the average attention to the context relative to the generated answer, showed some flex between the two generation classes. The *CA* metric didn’t provide promising results compared to the other probing inputs. The accuracy of the probes trained on CA metrics is provided in the Table 3. The result shows relatively better performances when the classifier is trained in CA sequence of all layers than individually. How-

⁸Additional results for other models are reported in Appendix E

ever, the performances remain much lower than the representation-based input. It could mean that the evidence on the use of context or parametric knowledge is not contained in the attention output, at least for the last input token, or that we need to extract a finer-grained feature from the attention matrix.

QMT	Probe	Acc	L
VQA	LPA	67.39 %	-
	LPL	69.57 %	6 (24)
	CM	69.57 %	-
TQA	LPA	69.21 %	-
	LPL	51.61 %	1 (4)
	CM	71.11 %	-
ConQA	LPA	70.97 %	-
	LPL	51.61 %	1 (4)
	CM	62.90 %	-
Multi	LPA	65.58 %	-
	LPL	57.25 %	5 (20)
	CM	68.12 %	-

Table 3: Results on probing the Qwen3.5-9B model on CA metrics on each layer.

6.2 Comparing generalization on unseen modalities

In this second experiment, we attempt to determine whether the signal extracted from one modality remains valid for another. In this setting, we consider only the best layer and compare the performance of a probe trained on one modality when tested on the other.

We reported the performances of the probes in Table 4. While performances are much lower than those reported in the previous section, we can see that training on VQA can achieve up to 71.67% accuracy on the CM probe using hidden representation when applied to TQA, indicating that the VQA modality contains a signal that is also applicable to a certain extent to the text model. The results are much lower than with the linear probe in the same setting, indicating that the common signal captured is not linearly separable. On the other hand, applying the probe trained on the textual modality (TQA) to visual questions yielded poorer results, indicating that the textual signal does not transfer to the visual modality. In addition, results on ConflictQA show that when the probes are trained on TQA, linear probing can reach up to 73.94% accuracy. Thus, both tasks, i.e., open-ended and closed QA,

Train	Test	Input	Probe	Accuracy
VQA	TQA	CA	CM	50.30 %
			LPL	57.42 %
			LPA	61.21 %
		In.R	CM	71.67 %
			LPL	51.52 %
			LPA	56.06 %
TQA	VQA	PLH	LP	65.45 %
			CM	50.00 %
			LPL	50.0 %
		CA	LPA	50.0 %
			CM	55.08 %
			LPL	55.8 %
TQA	ConQA	PLH	LPA	52.9 %
			LP	50.0 %
			CM	50.00 %
		CA	LPL	59.15 %
			LPA	50.35 %
			CM	50.35 %
TQA	ConQA	In.R	LPL	73.94 %
			LPA	62.35 %
			LP	67.25 %

Table 4: Cross Modality results of Qwen3.5-9B.

share common signals about the use of context for generation.

6.3 What layers contain information about parametric and contextual knowledge

In this third experiment, we propose studying which layer contains sufficient information to determine whether context is used for generation. To explore the question, we plot the linear probe’s performance for each layer. The results are reported in the Figure 3 for TQA, the results for VQA and ConflictQA follow the same trend⁹. We can clearly observe that performance starts to increase at the mid-layer and is random for the first ones. These results confirm the behavior observed in the previous study. Meaning that the use of contextual or parametric knowledge is decided late in the model.

6.4 Embedding representation

The probes trained on one modality underperform on the other. We propose examining the representations of both TQA and VQA and determining whether a separation between the two classes holds independently of modality. Figure 5 presents the PCA of both modalities according to their gold

⁹See appendix H

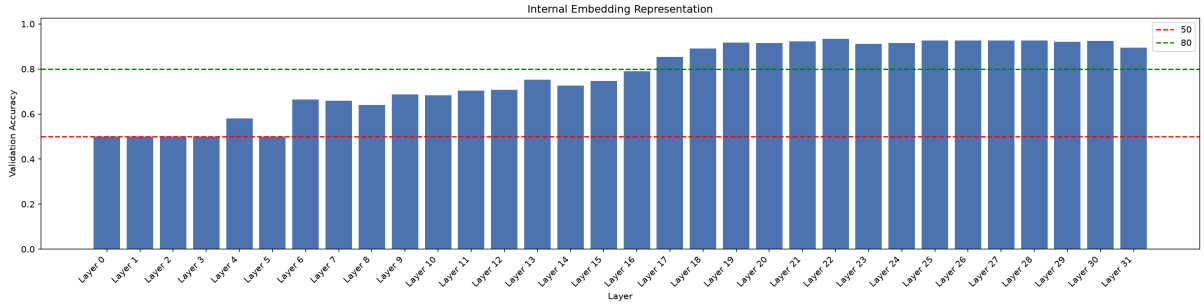


Figure 3: Per layer accuracy of the probe trained on the internal representation of all the layer for Qwen3.5-9B model on TQA corpus

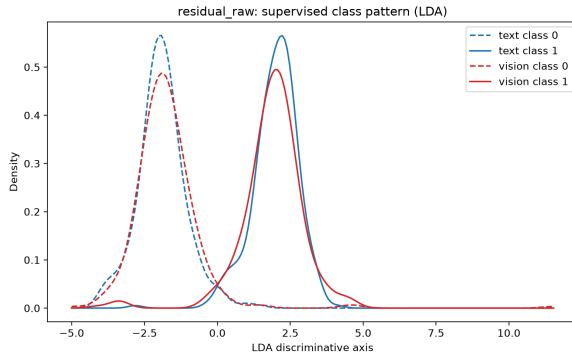


Figure 4: LDA distribution of the two Classes

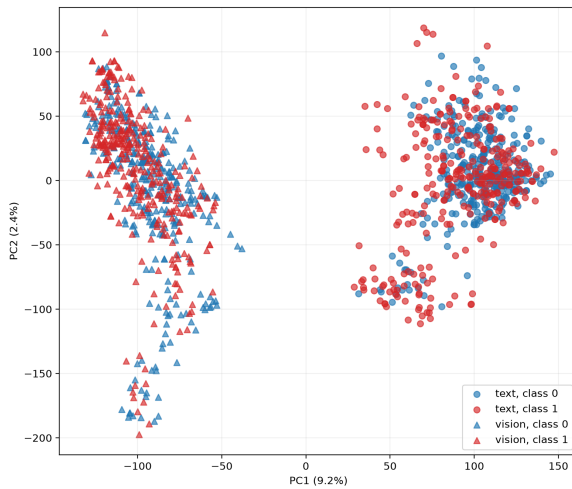


Figure 5: PCA of the two Classes

class. It clearly shows that the main variance directions are not those that discriminate against the use of specific knowledge. Nevertheless, the results indicate that this signal exists; accordingly, we propose studying the LDA distribution, as these methods allow us to assess whether there is a clear separation. We observe in Figure 4 that the classes are almost perfectly linearly separable. This indicates the internal representation shares some features across modalities for information source clas-

sification.

7 Conclusion

In this work, we study probing methods to automatically detect signals on the use of parametric or contextual knowledge. In contrast to previous work, we consider a bimodal setting that compares images with text-based context. We mainly focus on open-ended questions and therefore propose a method for creating a conflict-based corpus based on models. We also compare against a well-established corpus, namely the ConflictQA corpus. We studied different probing methods based on probe architectures and probe inputs¹⁰.

The results show that determining whether a model uses parametric knowledge can be inferred from the input context’s internal embeddings, meaning this choice can be detected without looking at the answer (before generation). Also, the conclusion regarding the importance of the last layers for detection reported in previous works remains valid, including for visual questions. Surprisingly, training a probe on visual-based QA and testing on the textual modality yields discriminative results, indicating that both types of QA share a common signal for detecting the use of parametric or contextual knowledge. Consequently, this means that training such detection tools on images would suffice to generalize to text, even having a smaller corpus for visual questions. However, the reverse is not true, and training on text-only QA cannot generalize to visual questions. The probing results show that linear probes can be effective for unimodal, whereas classification may have limitations for bimodal representations. While we demonstrated that the proposed methods can detect the type of information leveraged for generation, we cannot

¹⁰Resources and code are available at https://gitlab.lisn.upsaclay.fr/arakkal/uncertainlp_paper

yet control model generation; this question will be addressed in future work. Also, we plan to explore attribution methods to obtain a more fine-grained explanation of the generation process and determine which parts of the context are leveraged.

Limitations

In this study we acknowledge some of the limitations of this study such as;

Full qualitative evaluation of dataset. Our study mainly focused on the counterfactual dataset that we have created. This is mainly due to the data gap, from the entire dataset of conflictQA(Xie et al., 2024) subsection GPT4, we could only use 348-408 samples shown in Table 1. Since the dataset created is only evaluated automatically, it might contain some abnormality in the generated Counterfactual context as well as in labels. We are planning to perform a thorough manual annotation of the dataset created, to make it in gold standard.

Focus on Open-ended Questions. We also acknowledge the fact that the type of questions we focused on are mostly one word answer type focused on the education domain. Further experiments need to be done on descriptive type questions, and other type of questions.

References

- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. [BERT: pre-training of deep bidirectional transformers for language understanding](#). *CoRR*, abs/1810.04805.
- Thomas Gerald, Sahar Ghannay, Julie Lascar, Paul Lerner, and Anne Vilnat. 2026. Can multimodal llms generate pedagogical questions? In *LREC 2026*.
- Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. [PubMedQA: A dataset for biomedical research question answering](#). 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 2567–2577, Hong Kong, China. Association for Computational Linguistics.
- Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. [TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.
- Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. [Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension](#). *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5376–5384.
- Sunjun Kweon, Sooyohn Nam, Hyunseung Lim, Hwajung Hong, and Edward Choi. 2025. [A large-scale real-world evaluation of an LLM-based virtual teaching assistant](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 850–864, Vienna, Austria. Association for Computational Linguistics.
- Moritz Laurer, Wouter Van Attevelde, Andreu Casas, and Kasper Welbers. 2023. [Less Annotating, More Classifying: Addressing the Data Scarcity Issue of Supervised Machine Learning with Deep Transfer Learning and BERT-NLI](#). *Political Analysis*, pages 1–33.
- Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. [Entity-based knowledge conflicts in question answering](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Taffjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*.
- Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. 2023. [Scaling open-vocabulary object detection](#). *Preprint*, arXiv:2306.09683.
- Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. [MS MARCO: A human generated machine reading comprehension dataset](#). *CoRR*, abs/1611.09268.
- Qwen Team. 2026. [Qwen3.5: Towards native multimodal agents](#).
- Gorjan Radevski, Kiril Gashteovski, Shahbaz Syed, Christopher Malon, Sebastien Nicolas, Chia-Chien Hung, Timo Sztyler, Verena Heußer, Wiem Ben Rim, Masafumi Enomoto, et al. 2025. On synthesizing data for context attribution in question answering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16929–16950.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [SQuAD: 100,000+ questions for machine comprehension of text](#). In *Proceedings of*

the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. 2022. [LAION-5b: An open large-scale dataset for training next generation image-text models](#). In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Zineddine Tighidet, Andrea Mogini, Jiali Mei, Benjamin Piwowarski, and Patrick Gallinari. 2024. [Probing language models on their knowledge source](#). In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 604–614, Miami, Florida, US. Association for Computational Linguistics.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. [Crowdsourcing multiple choice science questions](#). In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, Copenhagen, Denmark. Association for Computational Linguistics.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2024. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *Proceedings of ICLR*.

Yu Zhao, Xiaotang Du, Giwon Hong, Aryo Pradipta Gema, Alessio Devoto, Hongru Wang, Xuanli He, Kam-Fai Wong, and Pasquale Minervini. 2024. Analysing the residual stream of language models under knowledge conflicts. *arXiv preprint arXiv:2410.16090*.

A Data before balancing

M T	Model	Before	After
VQA	Qwen3.5-9B	2419	920
VQA	Qwen3.5-4B	2419	1112
ConflictQA	Qwen3.5-9B	7326	408
ConflictQA	Qwen3.5-4B	7326	348
TQA	Qwen3.5-9B	4555	4400
TQA	Qwen3.5-4B	4555	4494

Table 5: Table showing the sample size across the modality, before and after balancing the corpus for training the classifier for each model.

B Spacy Bert Pipeline

Textual Counterfactual QA To create the counterfactual dataset, the initial pipeline consists of using Spacy Named entity recognition (NER) and

Bert (Devlin et al., 2018). The main idea is to take a question Q , and its associated context C and the true answer A . Using Spacy for named entity recognition, a list of entities E is created, where E is present in both context and answer but not in the question, shown in equation 6. These identified entities E are later masked in the context. As a final step, the masked context is given to the Bert model (Devlin et al., 2018) to predict a replacement token for the masked entities, thus creating a counterfactual context.

$$Q = \text{question}, C = \text{context}, A = \text{answer}, \quad (4)$$

$$NER(X) = \text{Spacy NER}, \quad (5)$$

$$E = NER(A) \cap NER(C) \setminus NER(Q). \quad (6)$$

However, the approach using Spacy and Bert has several limitations. A manual evaluation of a random sample of 50 showed that the created counterfactual context often lacks fluency and contains semantically inconsistent substitutions. These inconsistencies includes type swap substitution when an entity of one type is replaced with another type entity (example: Date to Person) and also unusable words .

C Initial Experiments

Initial Experiment The main aim of this experiment is to examine if there is any signal in the hidden layers of the model when the generation is from Context or from its parametric knowledge. We created a metrics called CAR defined by the equation 7 and calculated by the equation 7, 10, 9, and 8 :

$$CAR^l = \frac{1}{Q} \sum_{j=1}^Q \text{Att_query}_q^l, \quad (7)$$

$$A_{q,k}^l = \frac{1}{H} \sum_{h=1}^H A_{q,k}^{l,h} \quad (8)$$

$$\hat{A}_{q,k}^l = \frac{A_{q,k}^l}{\sum_{j=1}^Q A_{q,j}^l}, \quad (9)$$

$$\text{Att_query}_q^l = \sum_{k=1}^C \hat{A}_{q,k}^l, \quad (10)$$

The CAR measures the average attention given by the generated answer to the context, for textual models. Using the TQA dataset created as input we evaluated the change in CAR for the various generation. Particularly, we measured the CAR score

for all the layers for the two different instances for the original context as well as the counterfactual context as input.

Based on the generated answer, we created two classes for the two instances. For the generation where Counterfactual context is given we have A_{CF} when the answer is from the counterfactual context and A_{CF}^{real} when the answer is correct¹¹. For the instances when the Original context is given we labelled A_{Orig} when the answer is from the context and A_{other} when the answer not from the context¹²

To test the CAR, we ignored the other label A_{other} and checked the generations of A_{CF} , A_{CF}^{real} and A_{Orig} . We tested three different Qwen3.5-9B, Qwen3.5-4B and Qwen3.5-2B (Qwen Team, 2026). We observed a pattern which is similar in all the three models and dataset shown in figure 6. When the model is generating answer against the given context there is a flex in it's CAR in the middle layers compared when the generation is from the context.

From this we can see that across the model sizes there exists a pattern when the generation is from the parametric knowledge, 6 red line. The pattern of CAR is almost flat in the intermediate layers when the generation is follows the given context, ie A_{CF} and A_{Orig} have same pattern.

D Question creation for VQA dataset

For the resources collected for creating VQA data, there was not enough closed-ended question which are image independent. To amplify the closed-ended question and thereafter to create a counterfactual dataset, we used images from the collected resources to formulate objective-type questions that can also be answered independently only from the question. For each image Qwen3.5-9B model create a list of question that are;

- **Image Independent** : Possible to answer without an image. This exclude question like; *What is the label A presented in the image?*
- **Answer presence** : The answer to the Question is present in the given image.

¹¹Generated answer is from the parametric knowledge against the counterfactual context

¹²When the generation is labelled as other, not the correct answer

- **One word answer**: Answer to the question is one to two words
- **Unique answer** : The question cannot have an another answer from the image.

According to these rules the model generated the list of questions and their answer. These question are later evaluated by LLM as judge which looks into the question and the answer and checks;

- **Answer presence** : if the answer is present in the image.
- **Image Independence** :Check if the question is answerable without the image.
- **Unique answer** : The generated answer is the only answer to the question from the image.
- **Educational question** : the generated question and answer is rooted on the focusing on factual question than image recognition.

Based on these the LLM judge make binary classification (true/false) for the four criteria and give a score for the overall quality. We take the questions which fits all the criteria and have a score above 8 and rest of the questions are removed. Qwen3.5-9B model used for the Question generation and Judge. For the questions in TextbookQA model check if the question fits the criteria, if not it modifies the question and answer to fit the criteria.

E Classifiers performance on other model sizes

The results of the classifier added to the Qwen3.5-4B model is presented in the section. Table 6 presents the performance in the internal representation, whereas the Table 7 shows the accuracy on the CA feature extracted from the attention.

F CNN-MLP Classifier architecture

The CNN-MLP Classifier architecture is shown in the table, 7

G Counterfactual Corpus

In this section examples of the counterfactual dataset and Prompt used for the creation are presented. Figure 8 shows an example from VQA corpus created. and Figure 9 present an example from TQA corpus. Prompt is presented in the Figure 10

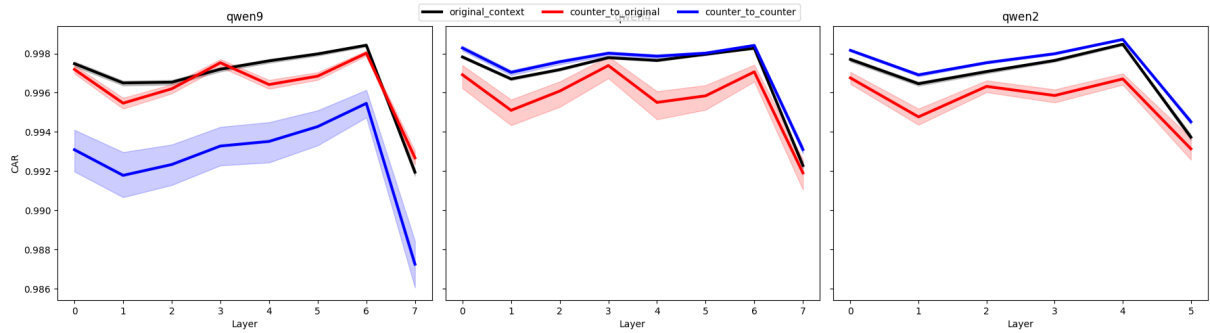


Figure 6: Initial experiment result of CAR: Attention Flex observed in various Qwen3.5 series model, tested on TQA dataset. Left: Qwen3.5-9B model, Center: Qwen3.5-4B, Right: Qwen3.5-2B. **Red line:** When the generated Answer is not from the context, when the context is counterfactual. **Blue Line:** When the answer is generated from the context, when context is counterfactual. **Black Line:** When the answer is generated from the context, when the context is correct.

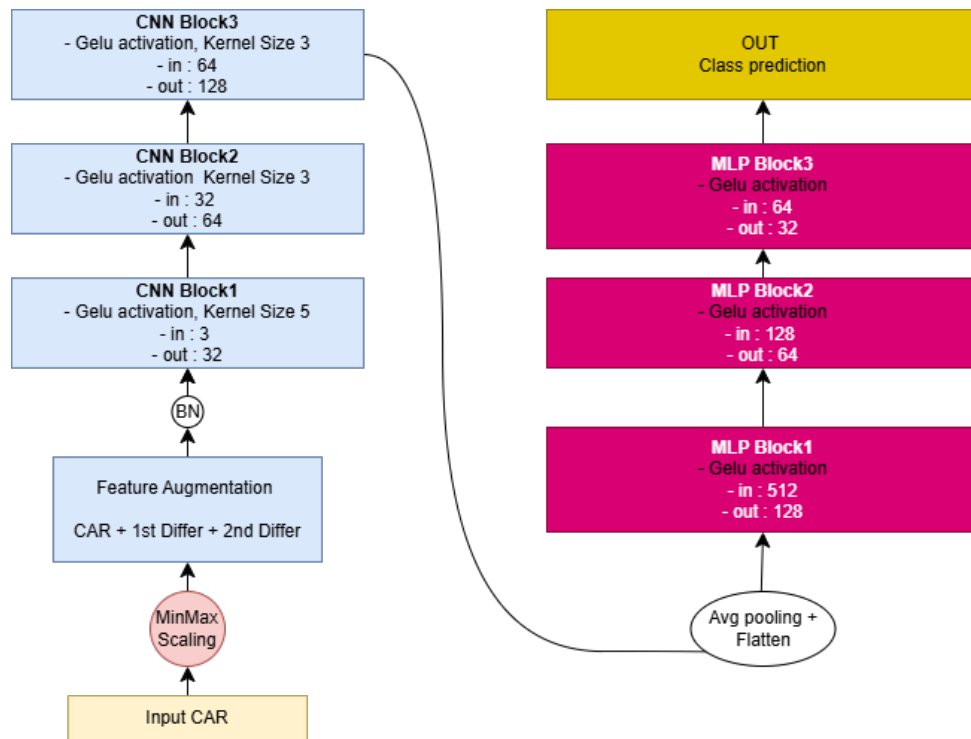


Figure 7: CNN-MLP architecture used for classification

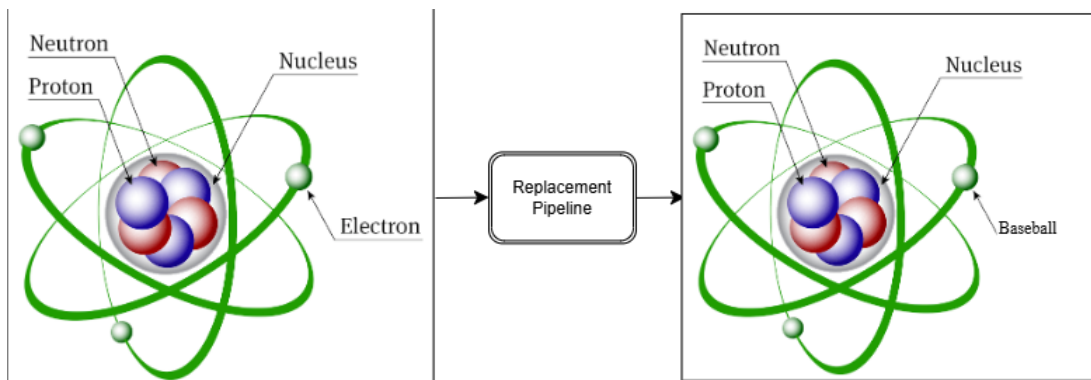


Figure 8: Example of the Counterfactual Image Created from the pipeline. **Question:**What is the Negative Charged particles seen outside an Atom?

Question: What is the least dangerous radioactive decay?

Original Context: All radioactive decay is dangerous to living things, but **alpha decay** is the least dangerous.

Counterfactual Context: All radioactive decay is dangerous to living things, but **beta decay** is the least dangerous.

Correct Answer: Alpha decay

Reference Answer: Beta decay

Figure 9: Example of a counterfactual textual sample used in the dataset.

Prompt:

You are a helpful assistant. You are given a Context (C), Question (Q) and an Answer (A). You must alter the original context into a new Context (SC) which is factually incorrect, such that:

1. The resulting new context (SC) must be grammatically fluent.
2. The resulting context (SC) must give a different answer to the same question (Q).
3. You can create the new context by changing entities or applying negation to the answer sentence.
4. You can't change or replace words present in the context that are also present in the Question.
5. If substitution is not possible, add negation to the sentence in the context supporting the answer of the question.

Figure 10: System and user prompt used to generate counterfactual contexts.

QMT	Input	Probe	Acc	L
VQA	In.R	LPL	83.83 %	21
		LPA	80.24 %	-
		CNN-MLP	83.83%	-
	PLH	LP	80.84 %	-
TQA	In.R	LPL	92.31 %	29
		LPA	91.50 %	-
		CNN-MLP	93.72 %	-
	PLH	LP	89.68%	-
ConQA	In.R	LPL	96.15%	22
		LPA	88.68%	-
		CNN-MLP	84.91%	-
	PLH	LP	86.79%	-
Multi	In.R	LPL	79.04 %	30
		LPA	84.43%	-
		CNN-MLP	83.53%	-
	PLH	LP	81.74 %	-

Table 6: Results on probing the Qwen3.5-4B model on it's internal representation of each layer.

QMT	Probe	Acc	L
VQA	LPA	68.86 %	-
	LPL	61.08 %	6 (24)
	CNN-MLP	71.86 %	-
TQA	LPA	74.29 %	-
	LPL	50.10 %	1 (4)
	CNN-MLP	76.92 %	-
ConQA	LPA	54.72 %	-
	LPL	50.72 %	1 (4)
	CNN-MLP	77.36 %	-
Multi	LPA	76.05 %	-
	LPL	55.99 %	5 (20)
	CNN-MLP	73.95 %	-

Table 7: Results on probing the Qwen3.5-4B model on CA metrics on each layer.

H Linear classifier trained on internal representation: performance results per layer

The performance of the linear probe trained on each layer of the model for the classification is plotted in the Figure 11 for classifier trained on Qwen3.5-4B,

Figures (12 & 13) present the accuracy per layer for the probe trained on multimodal data from the generation of Qwen3.5-4B and Qwen3.5-9B respectively. The plots (15 & 14) presents the accuracy for the Qwen3.5-4B and Qwen3.5-9B for Visual Modalities.

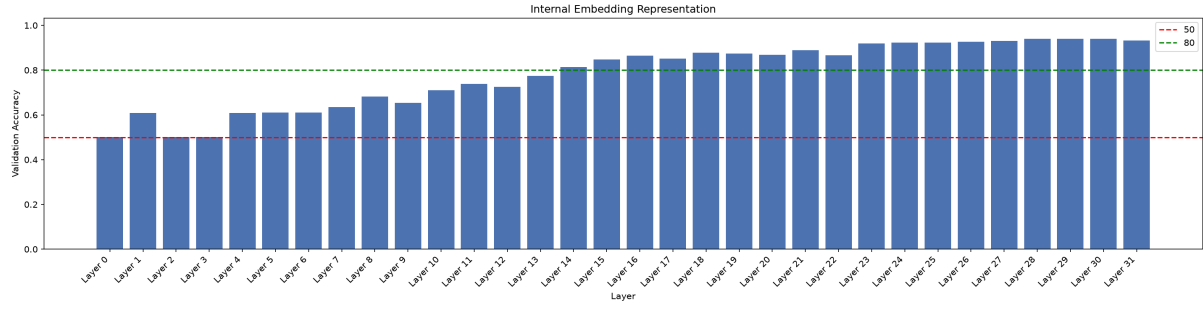


Figure 11: Per layer accuracy of the probe trained on the internal representation of all the layer for Qwen3.5-4B model on TQA

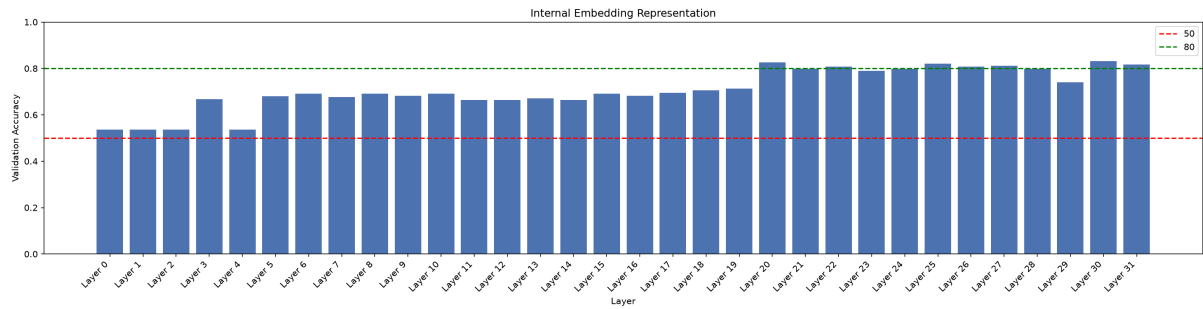


Figure 12: Per layer accuracy of the probe trained on the internal representation of all the layer for Qwen3.5-4B model on Multimodal classifier

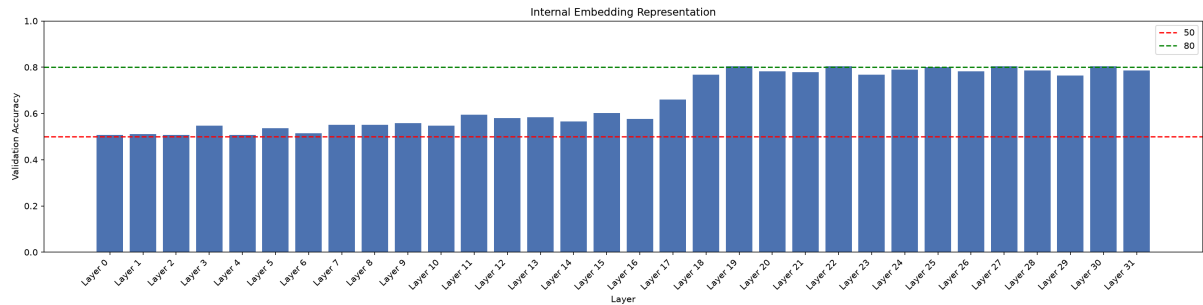


Figure 13: Per layer accuracy of the probe trained on the internal representation of all the layer for Qwen3.5-9B model on Multimodal classifier

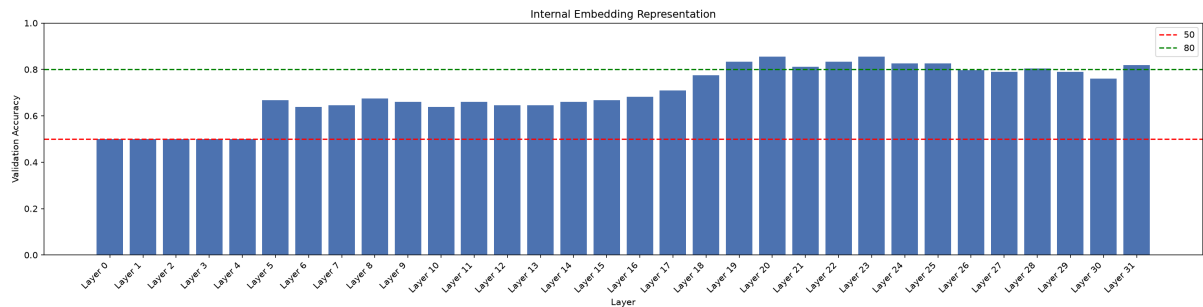


Figure 14: Per layer accuracy of the probe trained on the internal representation of all the layer for Qwen3.5-9B model on VQA classifier

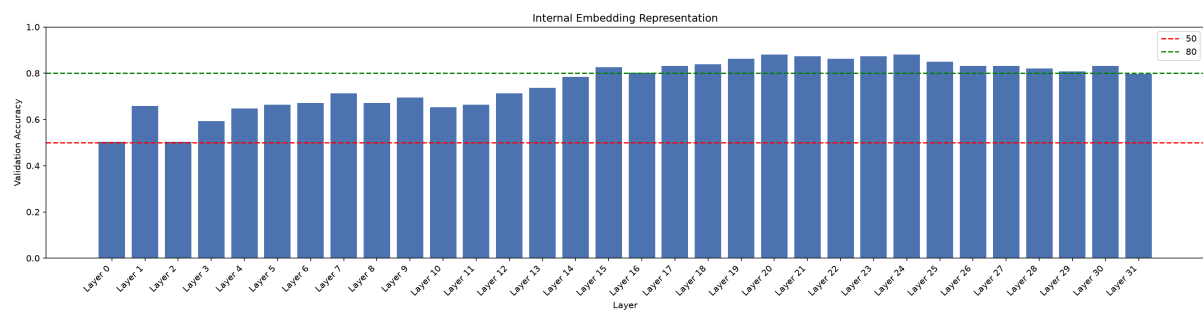


Figure 15: Per layer accuracy of the probe trained on the internal representation of all the layer for Qwen3.5-4B model on VQA classifier