Hallucination Detection in LLMs Using Spectral Features of Attention Maps

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have demonstrated remarkable performance across various tasks but remain prone to hallucinations. Detecting hallucinations is essential for safety-critical applications, and recent methods leverage attention map properties to this end, though their effectiveness remains limited. In this work, we investigate the spectral features of attention maps by interpreting them as adjacency matrices of graph structures. We propose the $\mathrm{LapEigvals}$ method, which utilises the top-k eigenvalues of the Laplacian matrix derived from the attention maps as an input to hallucination detection probes. Empirical evaluations demonstrate that our approach achieves state-of-the-art hallucination detection performance among attention-based methods. Extensive ablation studies further highlight the robustness and generalisation of $\mathrm{LapEigvals}$, paving the way for future advancements in the hallucination detection domain.

1 Introduction

The recent surge of interest in Large Language Models (LLMs), driven by their impressive performance across various tasks, has led to significant advancements in their training, fine-tuning, and application to real-world problems. Despite progress, many challenges remain unresolved, particularly in safety-critical applications where the cost of errors is high. A significant issue is that LLMs are prone to hallucinations, i.e. generating "content that is nonsensical or unfaithful to the provided source content" (Farquhar et al., 2024; Huang et al., 2023). Since eliminating hallucinations is impossible (Lee, 2023; Xu et al., 2024), there is a pressing need for methods to detect when a model produces hallucinations. In addition, uncovering internal behaviour while studying hallucinations of LLMs might reveal significant progress in understanding their characteristics, fostering further development in the field. Recent studies have shown that hallucinations can be detected using internal states of the model, e.g., hidden states (Chen et al., 2024) or attention maps (Chuang et al., 2024a), and that LLMs can internally "know when they do not know" (Azaria and Mitchell, 2023; Orgad et al., 2025). We provide new insights showing that spectral features of attention maps coincide with hallucinations, and based on that observation, we introduce a novel method for detecting hallucinations.

As highlighted by (Barbero et al., 2024), attention maps can be viewed as weighted adjacency matrices of graphs. Building on this perspective, we performed statistical analysis and demonstrated that the eigenvalues of a Laplacian matrix derived from attention maps serve as good predictors of hallucinations. We propose the $\mathrm{LapEigvals}$ method, which utilises the top-k eigenvalues of the Laplacian as input features of a probing model to detect hallucinations. We share full implementation in a public repository: `https://anonymous.4open.science/r/lapeig-acl-2025`.

We summarise our contributions as follows:

(1) We perform statistical analysis of the Laplacian matrix derived from attention maps and show that it could serve as a better predictor of hallucinations compared to the previous method relying on the log-determinant of the maps.

(2) Building on that analysis and advancements in the graph-processing domain, we propose leveraging the top-k eigenvalues of the Laplacian matrix as features for hallucination detection probes and empirically show that it achieves state-of-the-art performance among attention-based approaches.

(3) Through extensive ablation studies, we demonstrate properties, robustness and generalisation of $\mathrm{LapEigvals}$ and suggest promising directions for further development.

2 Motivation

Considering the attention matrix as an adjacency matrix representing a set of Markov Chains, each corresponding to one layer of an LLM (Barbero et al., 2024) (Figure 2), we can leverage its spectral properties, as was done in many successful graph-based methods (Mohar, 1997; von Luxburg, 2007; Bruna et al., 2013; Topping et al., 2022). In particular, it was shown that graph Laplacian might help to describe several graph properties, like the presence of bottlenecks (Topping et al., 2022; Black et al., 2023). We hypothesise that hallucinations may be related to disturbance of information flow caused by some form of bottleneck.

To assess whether our hypothesis holds, we measured if graph spectral features provide a stronger coincidence with hallucinations than the previous attention-based method - AttentionScore (Sriramanan et al., 2024). We prompted an LLM with questions from the TriviaQA dataset (Joshi et al., 2017) and extracted attention maps, differentiating by layers and heads. We then computed the spectral features, i.e., the 10 largest eigenvalues of the Laplacian matrix from each head and layer. Further, we conducted a two-sided Mann-Whitney U test to compare whether Laplacian eigenvalues and the values of AttentionScore are different between hallucinated and non-hallucinated examples. Figure 1 shows p-values for all layers and heads, indicating that AttentionScore often results in higher p-values compared to Laplacian eigenvalues. Overall, we studied 6 datasets and 5 LLMs and found similar results, and present all results in Appendix A. Based on these findings, we propose leveraging top-k Laplacian eigenvalues as features for a hallucination probe.

3 Method

In our method, we train a hallucination probe using only attention maps, which we extracted during LLM inference, as illustrated in Figure 2. The attention map is a matrix containing attention scores for all tokens processed during inference, while the hallucination probe is a logistic regression model that uses features derived from attention maps as input. This work's core contribution is using the top-k eigenvalues of the Laplacian matrix as input features, which we detail below.

Denote $\mathbf{A}^{(l,h)} \in \mathbb{R}^{T\times T}$ as the attention map matrix for layer $l \in \{1 \dots L\}$ and attention head $h \in \{1 \dots H\}$, where T is the total number of tokens generated by an LLM (including input tokens), L the number of layers (transformer blocks), and H the number of attention heads. The attention matrix is row-stochastic, meaning each row sums to 1 ($\sum_{j=0}^{T} \mathbf{A}^{(l,h)}_{:,j} = \mathbf{1}$). It is also lower triangular ($a^{(l,h)}_{ij} = 0$ for all $j > i$) and non-negative ($a^{(l,h)}_{ij} \geq 0$ for all i, j). We can view $\mathbf{A}^{(l,h)}$ as a weighted adjacency matrix of a directed graph, where each node represents processed token, and each directed edge from token i to token j is weighted by the attention score, as depicted in Figure 2.

Then, we define the Laplacian of a layer l and attention head h as:

$$\mathbf{L}^{(l,h)} = \mathbf{D}^{(l,h)} - \mathbf{A}^{(l,h)}, \tag{1}$$

where $\mathbf{D}^{(l,h)}$ is a diagonal degree matrix. Since the attention map defines a directed graph, we distinguish between the *in-degree* and *out-degree* matrices. The *in-degree* is computed as the sum of attention scores from preceding tokens, and due to the softmax normalization, it is uniformly 1. Therefore, we define $\mathbf{D}^{(l,h)}$ as the *out-degree* matrix, which quantifies the total attention a token receives from tokens that follow it. To ensure these values remain independent of the sequence length, we normalize them by the number of subsequent tokens (i.e., the number of outgoing edges).

$$d^{(l,h)}_{ii} = \frac{\sum_u a^{(l,h)}_{ui}}{T - i}, \tag{2}$$

where $i, u \in \{0 \dots (T-1)\}$ denote token indices. Such defined Laplacian is bounded, i.e. $\mathbf{L}^{(l,h)}_{ij} \in [-1, 1]$ (see Appendix B). Intuitively, the resulting Laplacian for each processed token represents the average attention score to previous tokens reduced by the attention score to itself. As eigenvalues of the Laplacian can encode information about information flow in graph (von Luxburg, 2007; Topping et al., 2022), we take eigenvalues of $\mathbf{L}^{(l,h)}$, which are diagonal entries, due to the lower triangularity of the Laplacian matrix, and sort them:

$$\tilde{z}^{(l,h)} = \operatorname{sort}\left(\operatorname{diag}\left(\mathbf{L}^{(\mathrm{l,h})}\right)\right) \tag{3}$$

Recently, (Zhu et al., 2024) found features from the entire token sequence, rather than a single token, improving hallucination detection. Similarly, (Kim et al., 2024) demonstrated that information from all

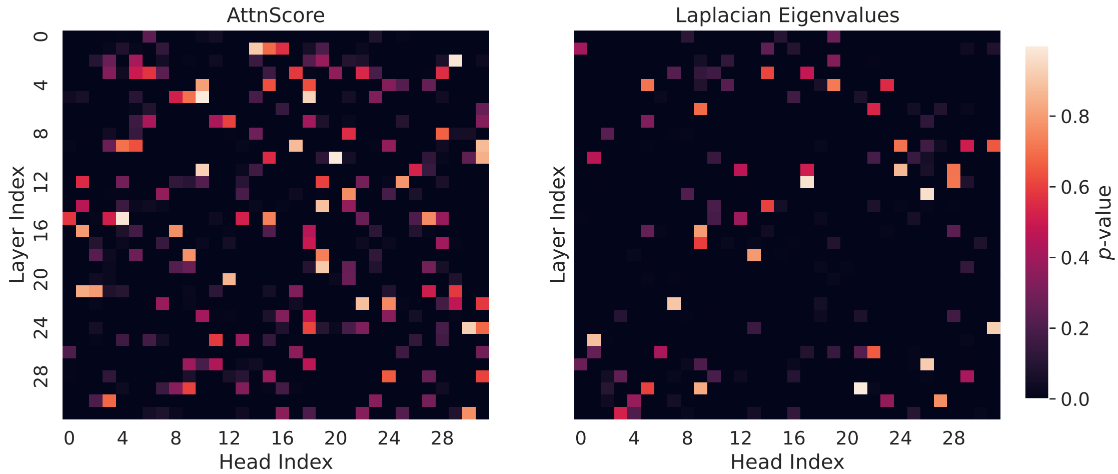


Figure 1: Visualisation of p-values from the two-sided Mann-Whitney U test for all layers and heads of `Llama-3.1-8B` across two feature types: $\mathrm{AttentionScore}$ and the $k = 10$ Laplacian eigenvalues. These features were derived from attention maps collected when the LLM answered questions from the TriviaQA dataset. Higher p-values indicate no significant difference in feature values between hallucinated and non-hallucinated examples. For $\mathrm{AttentionScore}$, 80% of heads have $p < 0.05$, while for Laplacian eigenvalues, this percentage is 91%. Therefore, Laplacian eigenvalues may be better predictors of hallucinations, as feature values across more heads exhibit statistically significant differences between hallucinated and non-hallucinated examples.

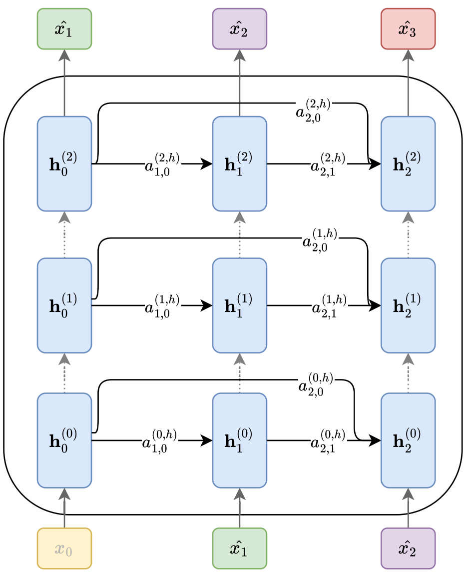


Figure 2: The autoregressive inference process in an LLM is depicted as a graph for a single attention head h (as introduced by (Vaswani, 2017)) and three generated tokens ($\hat{x}_1, \hat{x}_2, \hat{x}_3$). Here, $\mathbf{h}_i^{(l)}$ represents the hidden state at layer l for the input token i, while $a_{i,j}^{(l,h)}$ denotes the scalar attention score between tokens i and j at layer l and attention head h. Arrows direction refers to information flow during inference.

layers, instead of a single one in isolation, yields better results on this task. Motivated by these findings, our method uses features from all tokens and all layers as input to the probe. Therefore, we take the top-k largest values from each head and layer, and concatenate them into a single feature vector z, where k is a hyperparameter of our method:

$$z = \mathop{\Big\|}_{\forall l \in L, \forall h \in H} \left[\tilde{z}_T^{(l,h)}, \tilde{z}_{T-1}^{(l,h)}, \ldots, \tilde{z}_{T-k}^{(l,h)} \right] \quad (4)$$

Since LLMs contain dozens of layers and heads, the probe input vector $z \in \mathbb{R}^{T*L*H*k}$ would suffer from large dimensionality. Thus, we project it to lower dimensionality using the PCA (Jolliffe and Cadima, 2016). We call our approach LapEigvals.

4 Experimental setup

4.1 Dataset construction

We use annotated QA datasets to construct the hallucination detection datasets and label incorrect LLM answers as hallucinations. To determine whether the generated answers were correct, we adopted the *llm-as-judge* approach (Zheng et al., 2023), as in previous studies (Orgad et al., 2025). Specifically, we prompted a large LLM to classify each response as either *hallucination*, *non-hallucination*, or *rejected*, where *rejected* indicates that it was unclear whether the answer was correct, e.g., the model refused to answer due to insufficient knowledge. Based on the manual qualitative inspection of several LLMs, we employed `gpt-4o-mini` (OpenAI et al., 2024) as the judge model since it

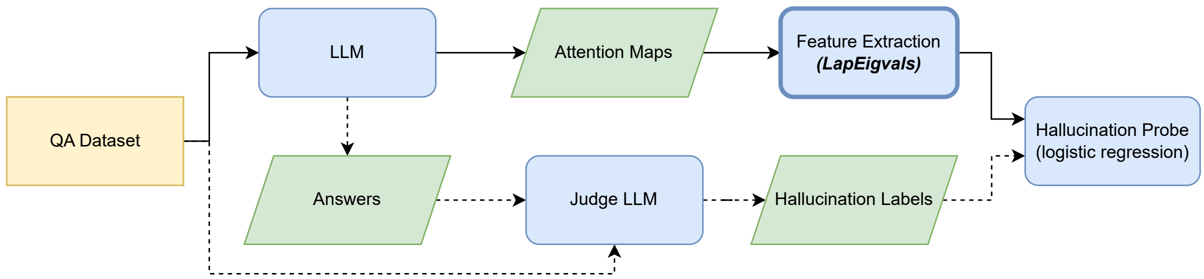


Figure 3: Overview of the methodology used in this work. Solid lines indicate the test-time pipeline, while dashed lines represent additional pipeline steps for generating labels for training the hallucination probe (logistic regression). The primary contribution of this work is leveraging the top-k eigenvalues of the Laplacian as features for the hallucination probe, highlighted with a bold box on the diagram.

provides the best trade-off between accuracy and cost.

For experiments, we selected 6 QA datasets previously utilised in the context of hallucination detection (Chen et al., 2024; Kossen et al., 2024; Chuang et al., 2024b; Mitra et al., 2024). Specifically, we used the validation set of NQOpen (Kwiatkowski et al., 2019), comprising 3610 question-answer pairs, and the validation set of TriviaQA (Joshi et al., 2017), containing 7983 pairs. To evaluate our method on longer inputs, we employed the development set of CoQA (Reddy et al., 2019) and the *rc.nocontext* portion of the SQuADv2 (Rajpurkar et al., 2018) datasets, with 5928 and 9960 examples, respectively. Additionally, we incorporated the QA part of the HaluEval (Li et al., 2023) dataset, containing 10000 examples, and the `generation` part of the TruthfulQA (Lin et al., 2022) benchmark with 817 examples. For TriviaQA, CoQA, and SQuADv2, we followed the same preprocessing procedure as (Chen et al., 2024).

We generate answers using 5 open-source LLMs: `Llama-3.1-8B`[1] and `Llama-3.2-3B`[2] (Grattafiori et al., 2024), `Phi-3.5`[3] (Abdin et al., 2024), `Mistral-Nemo`[4] (Mistral AI Team and NVIDIA, 2024), `Mistral-Small-24B`[5] (Mistral AI Team, 2025). We use two `softmax` temperatures for each LLM when decoding ($temp \in \{0.1, 1.0\}$) and one prompt (showed on Listing 3). Overall, we evaluated hallucination detection probes on 10 LLM configurations and 6 QA datasets. We present the frequency of classes for answers from each configuration in Figure 9 (Appendix E).

[1]hf.co/meta-llama/Llama-3.1-8B-Instruct
[2]hf.co/meta-llama/Llama-3.2-3B-Instruct
[3]hf.co/microsoft/Phi-3.5-mini-instruct
[4]hf.co/mistralai/Mistral-Nemo-Instruct-2407
[5]hf.co/mistralai/Mistral-Small-24B-Instruct-2501

4.2 Hallucination Probe

As a hallucination probe, we take a logistic regression model, using the implementation from scikit-learn (Pedregosa et al., 2011) with all parameters default, except for $max\_iter = 2000$ and $class\_weight = ''balanced''$. For top-$k$ eigenvalues, we tested 5 values of $k \in \{5, 10, 20, 50, 100\}$[6] and selected the result with the highest efficacy. All eigenvalues are projected with PCA onto 512 dimensions, except in *per-layer* experiments where there may be fewer than 512 features. In these cases, we apply PCA projection to match the input feature dimensionality, i.e., decorrelating them. As an evaluation metric, we use AUROC on the test split.

4.3 Baselines

Our method is a supervised approach to detect hallucinations solely from attention maps. For a fair comparison, we modify unsupervised $\mathrm{AttentionScore}$ (Sriramanan et al., 2024) to take log-determinants for each head as features instead of summing them. We also add original $\mathrm{AttentionScore}$ with the summation over heads for a reference. To evaluate the effectiveness of our proposed Laplacian eigenvalues, we also compare it to using raw attention maps and call it $\mathrm{AttnEigvals}$. Additionally, in Appendix F.1, we provide results for each approach but *per-layer*, and in Appendix F.3 we showcase comparison with method relying on hidden states. We provide implementation and hardware details in Appendix C.

Table 1: Test AUROC for LapEigvals and several baseline methods. AUROC values were obtained in the single run of logistic regression training on features from a dataset generated with $temp = 1.0$. We marked results for AttentionScore in gray as it is unsupervised approach, not directly comparable to the other ones. In **bold**, we highlight the best performance individually for each dataset and LLM. See Appendix F for extended results.

LLM	Feature	Test AUROC (↑)					
		CoQA	HaluevalQA	NQOpen	SQuADv2	TriviaQA	TruthfulQA
Llama3.1-8B	AttentionScore	0.493	0.589	0.556	0.538	0.532	0.541
Llama3.1-8B	AttnLogDet	0.769	0.827	0.793	0.748	0.842	0.814
Llama3.1-8B	AttnEigvals	0.782	0.819	0.790	0.768	0.843	**0.833**
Llama3.1-8B	LapEigvals	**0.830**	**0.874**	**0.827**	**0.791**	**0.889**	0.829
Llama3.2-3B	AttentionScore	0.509	0.588	0.546	0.530	0.515	0.581
Llama3.2-3B	AttnLogDet	0.700	0.801	0.690	0.734	0.789	**0.795**
Llama3.2-3B	AttnEigvals	0.724	0.819	**0.694**	0.749	0.804	0.723
Llama3.2-3B	LapEigvals	**0.812**	**0.828**	0.693	**0.757**	**0.832**	0.787
Phi3.5	AttentionScore	0.520	0.541	0.594	0.504	0.540	0.554
Phi3.5	AttnLogDet	0.745	0.818	0.815	0.769	0.848	0.755
Phi3.5	AttnEigvals	0.771	0.829	0.798	0.782	0.850	**0.802**
Phi3.5	LapEigvals	**0.821**	**0.836**	**0.826**	**0.795**	**0.872**	0.777
Mistral-Nemo	AttentionScore	0.493	0.531	0.529	0.510	0.532	0.494
Mistral-Nemo	AttnLogDet	0.728	0.798	0.769	0.772	0.812	**0.852**
Mistral-Nemo	AttnEigvals	0.778	0.781	0.761	0.758	0.821	0.802
Mistral-Nemo	LapEigvals	**0.835**	**0.833**	**0.795**	**0.812**	**0.865**	0.828
Mistral-Small-24B	AttentionScore	0.516	0.504	0.462	0.455	0.463	0.451
Mistral-Small-24B	AttnLogDet	0.766	0.842	0.747	0.753	0.833	0.735
Mistral-Small-24B	AttnEigvals	0.805	0.848	0.751	0.760	0.844	**0.765**
Mistral-Small-24B	LapEigvals	**0.861**	**0.882**	**0.791**	**0.820**	**0.876**	0.748

5 Results

Table 1 presents the results of our method compared to the baselines. LapEigvals achieved the best performance among all tested methods on 5 out of 6 datasets. Moreover, our method consistently performs well across all 5 LLM architectures ranging from 3 up to 24 billion parameters. TruthfulQA was the only exception where LapEigvals was the second-best approach, yet it might stem from the small size of the dataset or severe class imbalance (depicted in Figure 9). In contrast, using eigenvalues of vanilla attention maps in AttnEigvals leads to worse performance, which suggests that transformation to Laplacian is the crucial step to uncover latent features of an LLM corresponding to hallucinations. In Appendix F, we show that LapEigvals consistently demonstrates a smaller generalisation gap, i.e., the difference between training and test performance is smaller for our method. While the AttentionScore method performed poorly, it is fully unsupervised and should not be directly compared to other approaches. However, its supervised counterpart – AttnLogDet – remains inferior to methods based on spectral features, namely LapEigvals and AttnEigvals. In Table 4 in Appendix F.1, we present extended results, including *per-layer* and *all-layers* breakdowns, two temperatures used during answer generation, and a comparison between training and test AUROC. Moreover, compared to probes based on hidden states, our method performs best in most of the tested settings, as shown in Appendix F.3.

6 Ablation studies

To better understand the behaviour of our method under different conditions, we conduct a comprehensive ablation study. This analysis provides valuable insights into the factors driving the LapEigvals performance and highlights the robustness of our approach across various scenarios. In order to ensure reliable results, we perform all studies on the TriviaQA dataset, which has a reasonable input size and number of examples.

6.1 How does the number of eigenvalues influence performance?

First, we verify how the number of eigenvalues influences the performance of the hallucination probe and present results for `Mistral-Small-24B` in Figure 4 (results for all models are showcased in

[6] For datasets with examples having less than 100 tokens, we stop at $k = 50$

Figure 10 in Appendix G). Generally, using more eigenvalues improves performance, but there is less variation in performance among different values of k for LapEigvals. Moreover, LapEigvals achieves significantly better performance with smaller input sizes, as AttnEigvals with the largest $k = 100$ fails to surpass LapEigvals's performance at $k = 5$. These results confirm that spectral features derived from the Laplacian carry a robust signal indicating the presence of hallucinations and highlight the strength of our method.

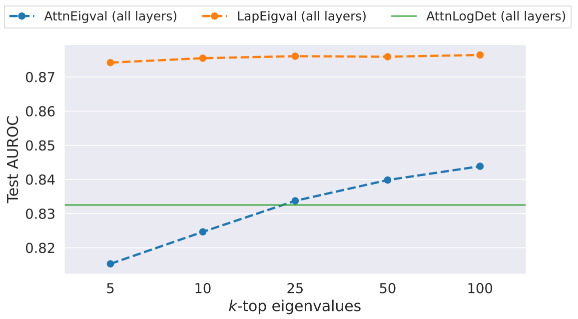


Figure 4: Probe performance across different top-k eigenvalues: $k \in \{5, 10, 25, 50, 100\}$ for TriviQA dataset with $temp = 1.0$ and `Mistral-Small-24B` LLM.

6.2 Does using all layers at once improve performance?

Second, we demonstrate that using all layers of an LLM instead of a single one improves performance. In Figure 5, we compare *per-layer* to *all-layer* efficacy for `Mistral-Small-24B` (results for all models are showcased in Figure 11 in Appendix G). For the *per-layer* approach, better performance is generally achieved in later LLM layers. Notably, peak performance varies across LLMs, requiring an additional search for each new LLM. In contrast, the *all-layer* probes consistently outperform the best *per-layer* probes across all LLMs. This finding suggests that information indicating hallucinations is spread across many layers of LLM, and considering them in isolation limits detection accuracy. Further, Table 4 in Appendix F summarises outcomes for the two variants on all datasets and LLM configurations examined in this work.

6.3 Does sampling temperature influence results?

Here, we compare LapEigvals to baselines on hallucination datasets produced with several temperatures used during decoding. Higher temperatures

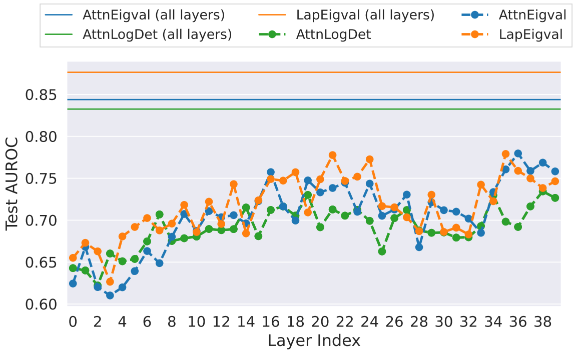


Figure 5: Analysis of model performance across different layers for `Mistral-Small-24B` and TriviaQA dataset with $temp = 1.0$ and $k = 100$ top eigenvalues (results for models operating on all layers provided for reference).

typically produce more hallucinated examples (Lee, 2023; Renze, 2024), leading to dataset imbalance. Thus, to mitigate the effect of data imbalance, we sample a subset of 1000 hallucinated and 1000 non-hallucinated examples 10 times for each temperature and train hallucination probes. Interestingly, in Figure 6, we observe that all models improve their performance at higher temperatures, but LapEigvals consistently achieves the best accuracy on all considered temperature values. The correlation of efficacy with temperature may be attributed to differences in the characteristics of hallucinations at higher temperatures compared to lower ones (Renze, 2024). Also, hallucination detection might be facilitated at higher temperatures due to underlying properties of `softmax` function (Veličković et al., 2024), and further exploration of this direction is left for future work.

6.4 How does LapEigvals generalizes?

To check whether our method generalises across datasets, we trained the hallucination probe on features from the training split of one QA dataset and evaluated it on the features from the test split of a different QA dataset. Due to space limitations, we present results for selected datasets and provide extended results and absolute efficacy values in Appendix H. Figure 7 showcases the percentage drop in Test AUROC when using a different training dataset compared to training and testing on the same QA dataset. We can observe that LapEigvals provides a performance drop comparable to other baselines, and in several cases, it generalises best. Interestingly, all methods exhibit poor generalisation on TruthfulQA, possibly due to dataset size

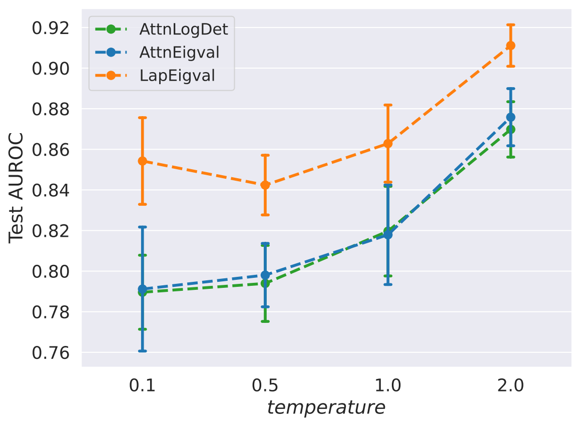


Figure 6: Test AUROC for different sampling *temp* values during answer decoding on the TriviaQA dataset, using $k = 100$ eigenvalues for LapEigvals and AttnEigvals with the `Llama-3.1-8B` LLM. Error bars indicate the standard deviation over 10 balanced samples containing $N = 1000$ examples per class.

or imbalance. Additionally, in Appendix H, we show that LapEigvals achieves the highest test performance in all scenarios (except for TruthfulQA).

6.5 How does performance vary across prompts?

Lastly, to assess the stability of our method across different prompts used for answer generation, we compared the results of the hallucination probes trained on features from four distinct prompts, the content of which is included in Appendix I. As shown in Table 2, LapEigvals consistently outperforms all baselines across all four prompts. While we can observe variations in performance across prompts, LapEigvals demonstrates the lowest standard deviation (0.05) compared to AttnLogDet (0.016) and AttnEigvals (0.07), indicating its greater robustness.

Table 2: Test AUROC across four different prompts for answers on the TriviaQA dataset using `Llama-3.1-8B` with $temp = 1.0$ and $k = 50$ (some prompts have led to fewer than 100 tokens). Prompt $\boldsymbol{p_3}$ was the main one used to compare our method to baselines, as presented in Tables 1.

Feature	Test AUROC (↑)			
	p_1	p_2	$\boldsymbol{p_3}$	p_4
AttnLogDet	0.847	0.855	0.842	0.860
AttnEigvals	0.840	0.870	0.842	0.875
LapEigvals	**0.882**	**0.890**	**0.888**	**0.895**

7 Related Work

Hallucinations in LLMs were proved to be inevitable (Xu et al., 2024), and to detect them, one can leverage either *black-box* or *white-box* approaches. The former approach uses only the outputs from an LLM, while the latter uses hidden states, attention maps, or logits corresponding to generated tokens.

Black-box approaches focus on the text generated by LLMs. For instance, (Li et al., 2024) verified the truthfulness of factual statements using external knowledge sources, though this approach relies on the availability of additional resources. Alternatively, *SelfCheckGPT* (Manakul et al., 2023) generates multiple responses to the same prompt and evaluates their consistency, with low consistency indicating potential hallucination.

White-box methods have emerged as a promising approach for detecting hallucinations (Farquhar et al., 2024; Azaria and Mitchell, 2023; Arteaga et al., 2024; Orgad et al., 2025). These methods are universal across all LLMs and do not require additional domain adaptation compared to black-box ones (Farquhar et al., 2024). They draw inspiration from seminal works on analysing the internal states of simple neural networks (Alain and Bengio, 2016), which introduced *linear classifier probes* – models operating on the internal states of neural networks. Linear probes have been widely applied to the internal states of LLMs, e.g., for detecting hallucinations.

One of the first such probes was SAPLMA (Azaria and Mitchell, 2023), which demonstrated that one could predict the correctness of generated text straight from LLM's hidden states. Further, the INSIDE method (Chen et al., 2024) tackled hallucination detection by sampling multiple responses from an LLM and evaluating consistency between their hidden states using a normalised sum of the eigenvalues from their covariance matrix. Also, (Farquhar et al., 2024) proposed a complementary probabilistic approach, employing entropy to quantify the model's intrinsic uncertainty. Their method involves generating multiple responses, clustering them by semantic similarity, and calculating Semantic Entropy using an appropriate estimator. To address concerns regarding the validity of LLM probes, (Marks and Tegmark, 2024) introduced a high-quality QA dataset with simple *true/false* answers and causally demonstrated that the truthfulness of such statements is linearly represented in

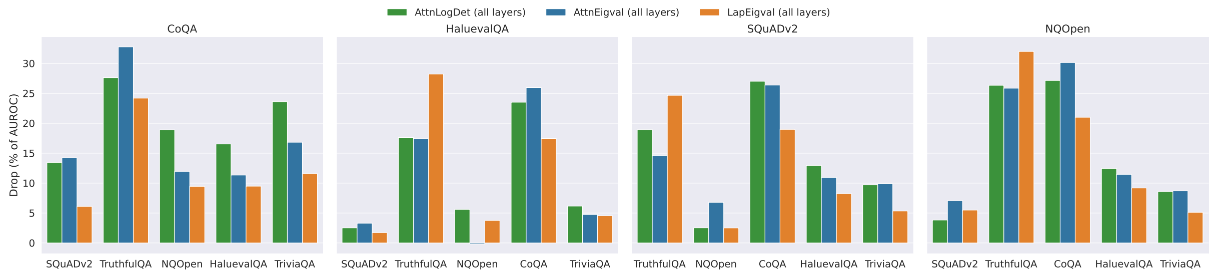


Figure 7: Generalisation across datasets measured as a per cent performance drop in Test AUROC (less is better) when trained on one dataset and tested on the other. Training datasets are indicated in the plot titles, while test datasets are shown on the x-axis. Results computed on `Llama-3.1-8B` with $k = 100$ top eigenvalues and $temp = 1.0$. Results for all datasets are presented in Appendix H.

LLMs, which supports the use of probes for short texts.

Self-consistency methods (Liang et al., 2024), like INSIDE or Semantic Entropy, require multiple runs of an LLM for each input example, which substantially lowers their applicability. Motivated by this limitation, (Kossen et al., 2024) proposed to use *Semantic Entropy Probe*, which is a small model trained to predict expensive Semantic Entropy (Farquhar et al., 2024) from LLM's hidden states. Notably, (Orgad et al., 2025) explored how LLMs encode information about truthfulness and hallucinations. First, they revealed that truthfulness is concentrated in specific tokens. Second, they found that probing classifiers on LLM representations do not generalise well across datasets, especially across datasets requiring different skills. Lastly, they showed that the probes could select the correct answer from multiple generated answers with reasonable accuracy, which they concluded with the LLM making mistakes at the decoding stage besides knowing the correct answer.

Recent studies have started to explore hallucination detection exclusively from attention maps. (Chuang et al., 2024a) introduced the *lookback ratio*, which measures how much attention LLMs allocate to relevant input parts when answering questions based on the provided context. The work most closely related to ours is (Sriramanan et al., 2024), which introduces the AttentionScore method. Although the process is unsupervised and computationally efficient, the authors note that its performance can depend highly on the specific layer from which the score is extracted. Compared to AttentionScore, our method is fully supervised and grounded in graph theory, as we interpret inference in LLM as a graph. While AttentionScore aggregates only the attention diagonal to compute its log-determinant, we instead derive features from the graph Laplacian, which captures all attention scores (see Eq. (1) and (2)). Additionally, we utilize all layers for detecting hallucination rather than a single one, demonstrating effectiveness of this approach. We also demonstrate that it performs poorly on the datasets we evaluated. Nonetheless, we drew inspiration from their approach, particularly using the lower triangular structure of matrices when constructing features for the hallucination probe.

8 Conclusions

In this work, we demonstrated that the spectral features of LLMs' attention maps, specifically the eigenvalues of the Laplacian matrix, carry a signal capable of detecting hallucinations. Specifically, we proposed the LapEigvals method, which employs the top-k eigenvalues of the Laplacian as input to the hallucination detection probe. Through extensive evaluations, we empirically showed that our method consistently achieves state-of-the-art performance among all tested approaches. Furthermore, multiple ablation studies demonstrated that our method remains stable across varying numbers of eigenvalues, diverse prompts, and generation temperatures while offering reasonable generalisation.

In addition, we hypothesise that self-supervised learning (Balestriero et al., 2023) could yield a more robust and generalisable approach while uncovering non-trivial intrinsic features of attention maps. Notably, results such as those in Section 6.3 suggest intriguing connections to recent advancements in LLM research (Veličković et al., 2024; Barbero et al., 2024), highlighting promising directions for future investigation.

Limitations

Supervised method In our approach, one must provide labelled hallucinated and non-hallucinated examples to train the hallucination probe. While this can be handled by the *llm-as-judge*, it might introduce some noise or pose a risk of overfitting. ***Limited generalisation across LLM architectures*** The method is incompatible with LLMs having different head and layer configurations. Developing architecture-agnostic hallucination probes is left for future work. ***Minimum length requirement*** Computing top-k Laplacian eigenvalues demands attention maps of at least k tokens (e.g., $k = 100$ require 100 tokens). ***Open LLMs*** Our method requires access to the internal states of LLM thus it cannot be applied to closed LLMs. ***Risks*** Please note that the proposed method was tested on selected LLMs and English data, so applying it to untested domains and tasks carries a considerable risk without additional validation.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. *arXiv preprint*. ArXiv:2404.14219 [cs].

Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes.

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, CK Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Michael Suo, Phil Tillet, Eikan Wang, Xiaodong Wang, William Wen, Shunting Zhang, Xu Zhao, Keren Zhou, Richard Zou, Ajit Mathews, Gregory Chanan, Peng Wu, and Soumith Chintala. 2024. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM.

Gabriel Y. Arteaga, Thomas B. Schön, and Nicolas Pielawski. 2024. Hallucination Detection in LLMs: Fast and Memory-Efficient Finetuned Models. In *Northern Lights Deep Learning Conference 2025*.

Amos Azaria and Tom Mitchell. 2023. The Internal State of an LLM Knows When It's Lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore. Association for Computational Linguistics.

Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. 2023. A Cookbook of Self-Supervised Learning. *arXiv preprint*. ArXiv:2304.12210 [cs].

Federico Barbero, Andrea Banino, Steven Kapturowski, Dharshan Kumaran, João G. M. Araújo, Alex Vitvitskyi, Razvan Pascanu, and Petar Veličković. 2024. Transformers need glasses! Information over-squashing in language tasks. *arXiv preprint*. ArXiv:2406.04267 [cs].

Mitchell Black, Zhengchao Wan, Amir Nayyeri, and Yusu Wang. 2023. Understanding Oversquashing in GNNs through the Lens of Effective Resistance. In *International Conference on Machine Learning*, pages 2528–2547. PMLR. ArXiv:2302.06835 [cs].

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2013. Spectral Networks and Locally Connected Networks on Graphs. *CoRR*.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024. INSIDE: LLMs' Internal States Retain the Power of Hallucination Detection. In *The Twelfth International Conference on Learning Representations*.

Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James R. Glass. 2024a. Lookback Lens: Detecting and Mitigating Contextual Hallucinations in Large Language Models Using Only Attention Maps. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1419–1436, Miami, Florida, USA. Association for Computational Linguistics.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2024b. DoLa: Decoding by Contrasting Layers Improves Factuality in Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630. Publisher: Nature Publishing Group.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc

Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The Llama 3 Herd of Models. *arXiv preprint*. ArXiv:2407.21783 [cs].

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *arXiv preprint*. ArXiv:2311.05232 [cs].

Ian T. Jolliffe and Jorge Cadima. 2016. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202. Publisher: Royal Society.

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.

Hazel Kim, Adel Bibi, Philip Torr, and Yarin Gal. 2024. Detecting LLM Hallucination Through Layer-wise Information Deficiency: Analysis of Unanswerable Questions and Ambiguous Prompts. *arXiv preprint*. ArXiv:2412.10246 [cs].

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. 2024. Semantic Entropy Probes: Robust and Cheap Hallucination Detection in LLMs. *arXiv preprint*. ArXiv:2406.15927 [cs].

Ruslan Kuprieiev, skshetry, Peter Rowland, Dmitry Petrov, Pawel Redzynski, Casper da Costa-Luis, David de la Iglesia Castro, Alexander Schepanovski, Ivan Shcheklein, Gao, Batuhan Taskaya, Jorge Orpinel, Fábio Santos, Daniele, Ronan Lamy, Aman Sharma, Zhanibek Kaimuldenov, Dani Hodovic, Nikita Kodenko, Andrew Grigorev, Earl, Nabanita Dash, George Vyshnya, Dave Berenbaum, maykulkarni, Max Hora, Vera, and Sanidhya Mangal. 2025. DVC: Data Version Control - Git for Data & Models.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew

Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:452–466. Place: Cambridge, MA Publisher: MIT Press.

Minhyeok Lee. 2023. A Mathematical Investigation of Hallucination and Creativity in GPT Models. *Mathematics*, 11(10):2320.

Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024. The Dawn After the Dark: An Empirical Study on Factuality Hallucination in Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10879–10899, Bangkok, Thailand. Association for Computational Linguistics.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. HaluEval: A Large-Scale Hallucination Evaluation Benchmark for Large Language Models. *arXiv preprint*. ArXiv:2305.11747 [cs].

Xun Liang, Shichao Song, Zifan Zheng, Hanyu Wang, Qingchen Yu, Xunkai Li, Rong-Hua Li, Feiyu Xiong, and Zhiyu Li. 2024. Internal Consistency and Self-Feedback in Large Language Models: A Survey. *CoRR*, abs/2407.14507.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Samuel Marks and Max Tegmark. 2024. The Geometry of Truth: Emergent Linear Structure in Large Language Model Representations of True/False Datasets. In *First Conference on Language Modeling*.

Mistral AI Team. 2025. Mistral-small-24B-instruct-2501.

Mistral AI Team and NVIDIA. 2024. Mistral-nemo-instruct-2407.

Kushan Mitra, Dan Zhang, Sajjadur Rahman, and Estevam Hruschka. 2024. FactLens: Benchmarking Fine-Grained Fact Verification. *arXiv preprint*. ArXiv:2411.05980 [cs].

Bojan Mohar. 1997. Some applications of Laplace eigenvalues of graphs. In Geňa Hahn and Gert Sabidussi, editors, *Graph Symmetry*, pages 225–275. Springer Netherlands, Dordrecht.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, C. J. Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. GPT-4 Technical Report. *arXiv preprint*. ArXiv:2303.08774 [cs].

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. 2025. LLMs Know More Than They Show: On the Intrinsic Representation of LLM Hallucinations. In *The Thirteenth International Conference on Learning Representations*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A Conversational Question Answering Challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266. Place: Cambridge, MA Publisher: MIT Press.

Matthew Renze. 2024. The Effect of Sampling Temperature on Problem Solving in Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7346–7356, Miami, Florida, USA. Association for Computational Linguistics.

Gaurang Sriramanan, Siddhant Bharti, Vinu Sankar Sadasivan, Shoumik Saha, Priyatham Kattakinda, and Soheil Feizi. 2024. LLM-Check: Investigating Detection of Hallucinations in Large Language Models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

The pandas development team. 2020. pandas-dev/pandas: Pandas.

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein. 2022. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Petar Veličković, Christos Perivolaropoulos, Federico Barbero, and Razvan Pascanu. 2024. softmax is not enough (for sharp out-of-distribution). *arXiv preprint*. ArXiv:2410.01104 [cs].

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Ulrike von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

Michael L. Waskom. 2021. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021. Publisher: The Open Journal.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is Inevitable: An Innate Limitation of Large Language Models. *arXiv preprint*. ArXiv:2401.11817.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and Chatbot Arena. In *Proceedings of the 37th International Conference* *on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc. Event-place: New Orleans, LA, USA.

Derui Zhu, Dingfan Chen, Qing Li, Zongxiong Chen, Lei Ma, Jens Grossklags, and Mario Fritz. 2024. PoLLMgraph: Unraveling Hallucinations in Large Language Models via State Transition Dynamics. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4737–4751, Mexico City, Mexico. Association for Computational Linguistics.

A Details of motivational study

We present a detailed description of the procedure used to obtain the results presented in Section 2 along with additional results for other datasets and LLM.

To test whether there is a statistically significant difference in the values of AttnEigvals and Laplacian eigenvalues, we first took QA datasets and ran inference with three LLMs, namely `Llama-3.1-8B` `Llama-3.2-3B` and `Phi-3.5`. Then, we extracted attention maps and computed AttentionScore (Sriramanan et al., 2024), i.e., log-determinant of attention maps. Unlike original work, we did not sum the scores over heads as we performed analysis at a single-head level of granularity. Also, we computed the Laplacian according to the definition presented in Section 3, took the 10 largest eigenvalues for each head, and treated each eigenvalue as a separate example. Finally, we ran the Mann–Whitney U test, leveraging SciPy implementation (Virtanen et al., 2020), and gathered p-values presented in Figure 1.

Table 3 presents the percentage of heads having a statistically significant difference in feature values between hallucinated and non-hallucinated examples, as indicated by $p < 0.05$ from the Mann-Whitney U test. These results show that the Laplacian eigenvalues better distinguish between the two classes for all considered LLMs and datasets.

Table 3: Percentage of heads having a statistically significant difference in feature values between hallucinated and non-hallucinated examples, as indicated by $p < 0.05$ from the Mann-Whitney U test. Results were obtained for AttentionScore and the 10 largest Laplacian eigenvalues on 6 datasets and 5 LLMs.

LLM	Dataset	% of $p < 0.05$	
		AttnScore	Laplacian eigvals
Llama3.1-8B	CoQA	40	87
Llama3.1-8B	HaluevalQA	91	93
Llama3.1-8B	NQOpen	78	83
Llama3.1-8B	SQuADv2	70	81
Llama3.1-8B	TriviaQA	80	91
Llama3.1-8B	TruthfulQA	40	60
Llama3.2-3B	CoQA	50	79
Llama3.2-3B	HaluevalQA	91	93
Llama3.2-3B	NQOpen	81	84
Llama3.2-3B	SQuADv2	69	74
Llama3.2-3B	TriviaQA	81	87
Llama3.2-3B	TruthfulQA	40	62
Phi3.5	CoQA	45	81
Phi3.5	HaluevalQA	80	86
Phi3.5	NQOpen	73	80
Phi3.5	SQuADv2	81	82
Phi3.5	TriviaQA	86	92
Phi3.5	TruthfulQA	41	53
Mistral-Nemo	CoQA	35	78
Mistral-Nemo	HaluevalQA	78	82
Mistral-Nemo	NQOpen	64	57
Mistral-Nemo	SQuADv2	54	56
Mistral-Nemo	TriviaQA	71	74
Mistral-Nemo	TruthfulQA	40	50
Mistral-Small-24B	CoQA	28	78
Mistral-Small-24B	HaluevalQA	68	70
Mistral-Small-24B	NQOpen	45	51
Mistral-Small-24B	SQuADv2	75	82
Mistral-Small-24B	TriviaQA	65	70
Mistral-Small-24B	TruthfulQA	43	52

B Bounds of the Laplacian

In the following section, we prove that the Laplacian defined in 3 is bounded and has at least one zero eigenvalue. Here, we denote eigenvalues as λ_i, and provide derivation for a single layer and head, which holds also after stacking them together into a single graph (set of per-layer graphs). For clarity we omit superscript (l, h) denoting layer and head.

Lemma 1. *The Laplacian eigenvalues are bounded:* $-1 \leq \lambda_i \leq 1$.

Proof. Due to the lower-triangular structure of the Laplacian, its eigenvalues lie on the diagonal and are given by:

$$\lambda_i = \mathbf{L}_{ii} = d_{ii} - a_{ii}$$

The out-degree are defined as:

$$d_{ii} = \frac{\sum_u a_{ui}}{T - i},$$

Since $0 \leq a_{ui} \leq 1$, the sum in the numerator is upper bounded by $T - i$, therefore $d_{ii} \leq 1$, and consequently $\lambda_i = \mathbf{L}_{ii} \leq 1$, which concludes upper-bound part of the proof.

Recall, that eigenvalues lie on the main diagonal of the Laplacian, hence $\lambda_i = \frac{\sum_u a_{uj}}{T-i} - a_{ii}$. To find the lower bound of λ_i, we need to minimize $X = \frac{\sum_u a_{uj}}{T-i}$ and maximize $Y = a_{ii}$. First, we note that X's denominator is always positive $T - i > 0$, since $i \in \{0 \ldots (T-1)\}$ (as defined by Eq. (2)). For numerator, we recall that $0 \leq a_{ui} \leq 1$, therefore the sum has its minimum at 0, hence $X \geq 0$. Second, to maximize $Y = a_{ii}$, we can take maximum of $0 \leq a_{ii} \leq 1$ which is 1. Finally, $X - Y = -1$, consequently $\mathbf{L}_{ii} \geq -1$, which concludes the lower-bound part of the proof. □

Lemma 2. *For every* $\mathbf{L}_{ii}$ *there exists at least one zero-eigenvalue and it corresponds to the last token* T*, i.e.,* $\lambda_T = 0$.

Proof. Recall, that eigenvalues lie on the main diagonal of the Laplacian, hence $\lambda_i = \frac{\sum_u a_{uj}}{T-i} - a_{ii}$. Consider last token, wherein the sum in the numerator reduces to $\sum_u a_{uj} = a_{TT}$, denominator becomes $T - i = T - (T - 1) = 1$, thus $\lambda_T = \frac{a_{TT}}{1} - a_{TT} = 0$. □

C Implementation details

In our experiments, we used HuggingFace Transformers (Wolf et al., 2020), PyTorch (Ansel et al., 2024), and scikit-learn (Pedregosa et al., 2011). We utilised Pandas (team, 2020) and Seaborn (Waskom, 2021) for visualisations and analysis. To version data, we employed DVC (Kuprieiev et al., 2025). We acquired attention maps using a single Nvidia A40 with 40GB VRAM, except for `Mistral-Small-24B` for which we used Nvidia H100 with 96GB VRAM. Training hallucination probe was done using CPU only. To compute labels using the *llm-as-judge* approach, we leveraged `gpt-4o-mini` model available through OpenAI API. Detailed hyperparameter configurations and code to reproduce the experiments is available in the public Git repository.

D Details of QA datasets

In this work, we used 6 open and publicly available question answering datasets: NQ-Open (Kwiatkowski et al., 2019) (CC-BY-SA-3.0 license), SQuADv2 (Rajpurkar et al., 2018) (CC-BY-SA-4.0 license), TruthfulQA (Apache-2.0 license) (Lin et al., 2022), HALUEval (MIT license) (Li et al., 2023), CoQA (Reddy et al., 2019) (domain-dependent licensing, detailed on `https://stanfordnlp.github.io/coqa/`), while TriviaQA (lacks clear licensing information, but was primarily shared as public benchmark). Research purposes fall into the intended use of these datasets. To preprocess and filter TriviaQA, CoQA, and SQuADv2 we utilized open-source code of (Chen et al., 2024)[7], which also borrows from (Farquhar et al., 2024)[8]. In Figure 8, we provide histogram plots of the number of tokens for $question$ and $answer$ of each dataset computed with `meta-llama/Llama-3.1-8B-Instruct` tokenizer.

[7] `https://github.com/alibaba/eigenscore` (MIT license)

[8] `https://github.com/lorenzkuhn/semantic_uncertainty` (MIT license)

E Hallucination dataset sizes

In Figure 9, we show the number of examples for each label determined with the *llm-as-judge* heuristic. It is worth noting that different generation configurations result in different splits, as LLMs might produce different answers. All examples classified as $Rejected$ were discarded from further experiments. We can observe that in most cases, datasets are imbalanced, underrepresenting non-hallucinated examples. Only for TriviaQA, there is an approximately balanced number of examples or even more non-hallucinated ones, depending on the configuration used. We split each dataset into 80% training examples and 20% test examples. Splits were stratified according to hallucination labels.

F Extended results

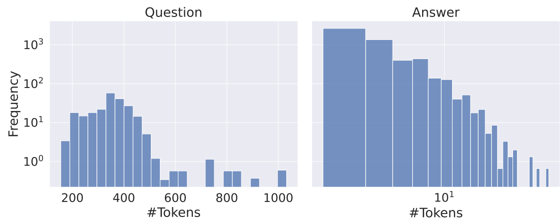
F.1 Extended method comparison

In Tables 4 and 5, we present the extended results from Table 1 in the main part of this paper. These results cover probes trained with both *all-layers* and *per-layer* variants across all models, as well as lower temperature ($temp \in \{0.1, 1.0\}$). In all cases, the *all-layers* variant outperforms the *per-layer* variant, suggesting that hallucination-related information is distributed across multiple layers. Additionally, we observe a smaller generalisation gap (measured as the difference between test and training performance) for the LapEigvals method, indicating more robust features present in the Laplacian eigenvalues. Finally, as demonstrated in Section 6, increasing the temperature during answer generation improves probe performance, which is also evident in Table 4, where probes trained on answers generated with $temp = 1.0$ consistently outperform those trained on data generated with $temp = 0.1$.

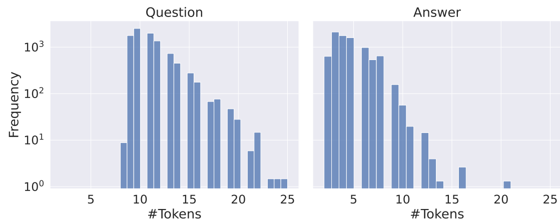
F.2 Best found hyperparameters

We present the hyperparameter values corresponding to the results in Table 1 and Table 4. Table 6 shows the optimal hyperparameter k for selecting the top-k eigenvalues from either the attention maps in AttnEigvals or the Laplacian matrix in LapEigvals. While fewer eigenvalues were sufficient for optimal performance in some cases, the best results were generally achieved with the highest tested value, $k = 100$.

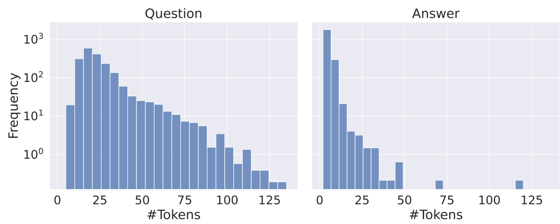
Table 7 reports the layer indices that yielded the highest performance for the *per-layer* models. Performance typically peaked in layers above the



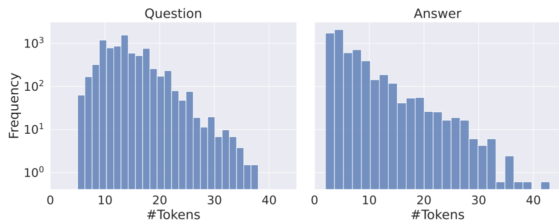
(a) CoQA



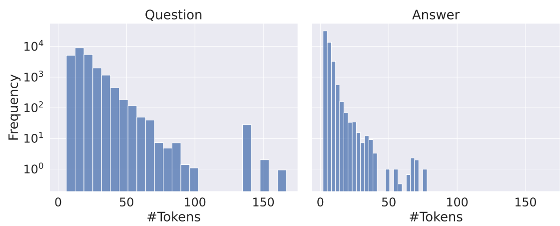
(b) NQOpen



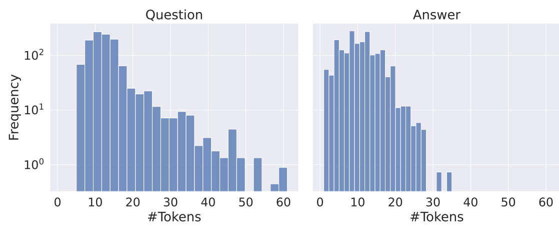
(c) HaluEval



(d) Squadv2



(e) TriviaQA



(f) TruthfulQA

Figure 8: Token count histograms for datasets used in the experiments. Number of tokens determined separately for $question$ (left-hand-side plots) and gold $answer$ (right-hand-side plots) of each example in the datasets with `meta-llama/Llama-3.1-8B-Instruct` tokeniser (whenever multiple possible answers occurred, they were flattened).

10th, especially for `Llama-3.1-8B`, where attention maps from the final layers more often led to better hallucination detection. Interestingly, the first layer's attention maps also produced strong performance in a few cases. Overall, no clear pattern emerges regarding the optimal layer, and as noted in prior work, selecting the best layer in the *per-layer* setup often requires a search.

F.3 Comparison with hidden-states-based baselines

We take approach considered in the previous works (Azaria and Mitchell, 2023; Orgad et al., 2025) and aligned to our evaluation protocol. Specifically, we trained a logistic regression classifier on PCA-projected hidden states to predict whether the model is hallucinating or not. To this end, we select the last token of the answer. While we also tested the last token of the prompt, we observed significantly lower performance, which aligns with results presented by (Orgad et al., 2025). We considered hidden states either from all layers or a single layer corresponding to the selected token. In all-layer scenario we use concatenation of hidden states of all layers, and in per-layer scenario we use hidden states of each layer separately and select the best performing layer.

In Table 8 we show obtained results. The all-layer version is consistently worse than our LapEigvals. Although the per-layer version outperforms LapEigvals, we argue that it should be treated as a rough reference since it is not a fully fair comparison. Note that our LapEigvals is designed specifically to operate on attention maps and shows the best performance among all attention-based methods. Our work is one of the first to detect hallucinations solely using attention maps, providing an important insight about behaviour of LLMs, and it motivates further theoretical research on information flow patterns inside these models.

G Extended results of ablations

In the following section, we extend the ablation results presented in Section 6.1 and Section 6.2. Figure 10 compares the top k eigenvalues across all five LLMs. In Figure 11 we present a layer-wise performance comparison for each model.

H Extended results of generalisation study

We present the complete results of the generalisation ablation discussed in Section 6.4 of the main paper. Table 9 reports the absolute Test AUROC values for each method and test dataset. Except for TruthfulQA, LapEigvals achieves the highest performance across all configurations. Notably, some methods perform close to random, whereas LapEigvals consistently outperforms this baseline. Regarding relative performance drop (Figure 12), LapEigvals remains competitive, exhibiting the lowest drop in nearly half of the scenarios. These results indicate that our method is robust but warrants further investigation across more datasets, particularly with a deeper analysis of TruthfulQA.

I QA prompts

Following, we describe all prompts for QA used to obtain the results presented in this work:

- prompt p_1 – medium-length one-shot prompt with single example of QA task (Listing 1),
- prompt p_2 – medium-length zero-shot prompt without examples (Listing 2),
- prompt p_3 – long few-shot prompt; the main prompt used in this work; modification of prompt used by (Kossen et al., 2024) (Listing 3),
- prompt p_4 – short-length zero-shot prompt without examples (Listing 4).

J LLM-as-Judge prompt

During hallucinations dataset construction we leveraged *llm-as-judge* approach to label answers generated by the LLMs. To this end, we utilised `gpt-4o-mini` with prompt in Listing 5, which is an adapted version of the prompt used by (Orgad et al., 2025).

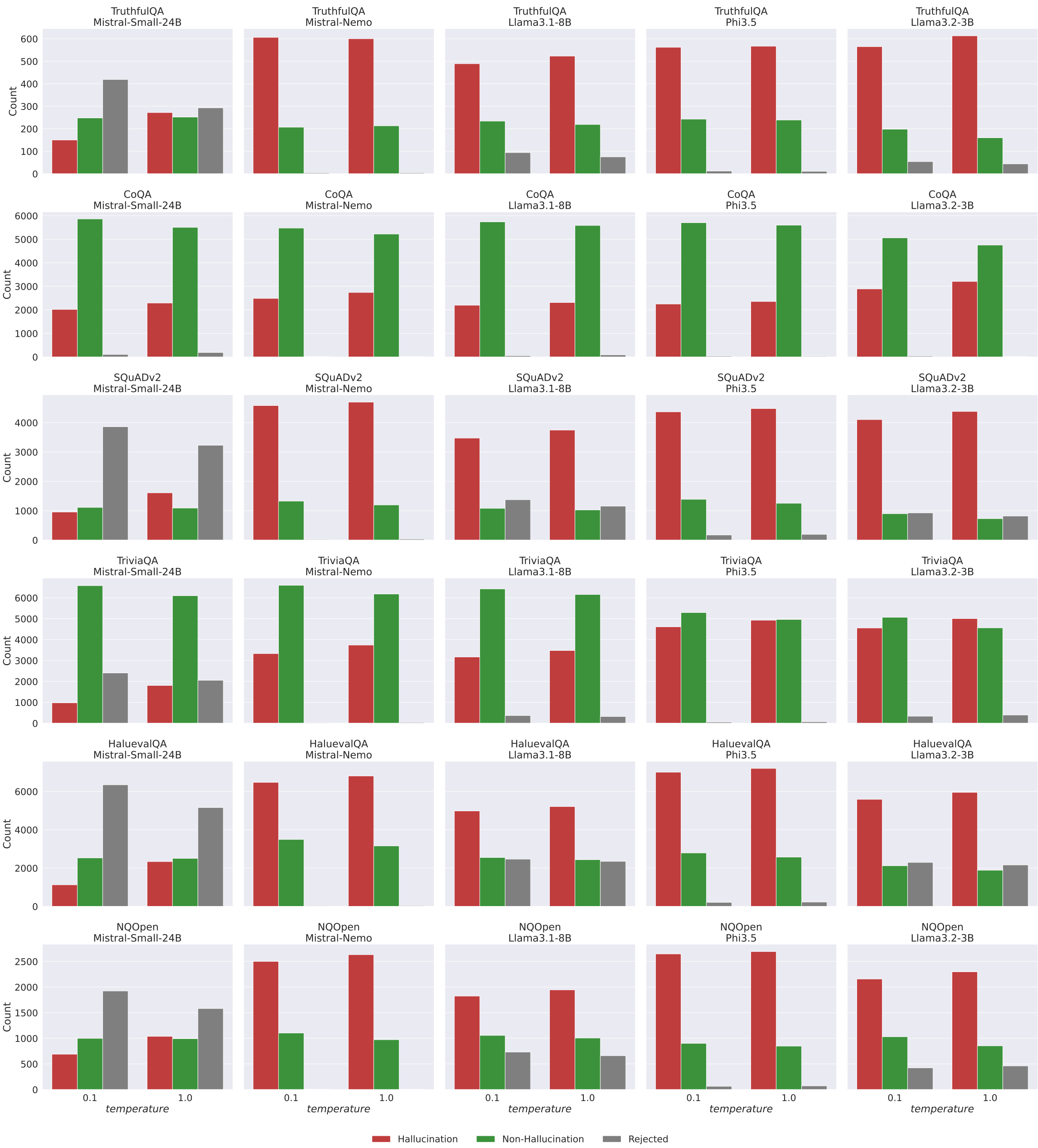


Figure 9: Number of examples per each label in generated datasets ($Hallucination$ - number of hallucinated examples, $Non{-}Hallucination$ - number of truthful examples, $Rejected$ - number of examples unable to evaluate).

Table 4: (Part I) Performance comparison of methods evaluated in this work on an extended set of configurations. We marked results for AttentionScore in gray as it is unsupervised approach, not directly comparable to the other ones. In **bold**, we highlight the best performance on the test split of data, individually for each dataset, LLM, and temperature.

LLM	Temp	Feature	*all-layers*	*per-layer*	Train AUROC						Test AUROC					
					CoQA	HaluevalQA	NQOpen	SQuADv2	TriviaQA	TruthfulQA	CoQA	HaluevalQA	NQOpen	SQuADv2	TriviaQA	TruthfulQA
Llama3.1-8B	0.1	AttentionScore		✓	0.509	0.667	0.607	0.556	0.567	0.563	0.541	0.653	0.631	0.575	0.571	0.650
Llama3.1-8B	0.1	AttentionScore	✓		0.494	0.614	0.568	0.522	0.522	0.489	0.504	0.587	0.558	0.521	0.511	0.537
Llama3.1-8B	0.1	AttnLogDet		✓	0.574	0.776	0.702	0.688	0.739	0.709	0.606	0.770	0.713	0.708	0.741	0.777
Llama3.1-8B	0.1	AttnLogDet	✓		0.843	0.884	0.851	0.839	0.861	0.913	0.770	0.837	0.768	0.758	0.827	0.820
Llama3.1-8B	0.1	AttnEigvals		✓	0.764	0.828	0.713	0.742	0.793	0.680	0.729	0.799	0.728	0.749	0.773	0.790
Llama3.1-8B	0.1	AttnEigvals	✓		0.861	0.895	0.878	0.858	0.867	0.979	0.776	0.838	0.755	0.781	0.822	0.819
Llama3.1-8B	0.1	LapEigvals		✓	0.758	0.817	0.698	0.707	0.781	0.708	0.757	0.793	0.711	0.733	0.780	0.764
Llama3.1-8B	0.1	LapEigvals	✓		0.869	0.901	0.864	0.855	0.896	0.903	**0.836**	**0.867**	**0.793**	**0.782**	**0.872**	**0.822**
Llama3.1-8B	1.0	AttentionScore		✓	0.514	0.640	0.607	0.558	0.578	0.533	0.525	0.642	0.607	0.572	0.602	0.629
Llama3.1-8B	1.0	AttentionScore	✓		0.507	0.602	0.580	0.534	0.535	0.546	0.493	0.589	0.556	0.538	0.532	0.541
Llama3.1-8B	1.0	AttnLogDet		✓	0.596	0.755	0.704	0.697	0.750	0.757	0.597	0.763	0.757	0.686	0.754	0.771
Llama3.1-8B	1.0	AttnLogDet	✓		0.848	0.882	0.856	0.846	0.867	0.930	0.769	0.827	0.793	0.748	0.842	0.814
Llama3.1-8B	1.0	AttnEigvals		✓	0.762	0.820	0.758	0.754	0.800	0.796	0.723	0.784	0.732	0.728	0.796	0.770
Llama3.1-8B	1.0	AttnEigvals	✓		0.867	0.889	0.873	0.867	0.876	0.972	0.782	0.819	0.790	0.768	0.843	**0.833**
Llama3.1-8B	1.0	LapEigvals		✓	0.760	0.803	0.732	0.722	0.795	0.751	0.743	0.789	0.725	0.724	0.794	0.764
Llama3.1-8B	1.0	LapEigvals	✓		0.879	0.896	0.866	0.857	0.901	0.918	**0.830**	**0.874**	**0.827**	**0.791**	**0.889**	0.829
Llama3.2-3B	0.1	AttentionScore		✓	0.526	0.697	0.592	0.570	0.570	0.569	0.547	0.714	0.643	0.582	0.551	0.564
Llama3.2-3B	0.1	AttentionScore	✓		0.506	0.635	0.523	0.515	0.534	0.473	0.519	0.644	0.573	0.561	0.510	0.489
Llama3.2-3B	0.1	AttnLogDet		✓	0.573	0.762	0.692	0.682	0.719	0.725	0.579	0.774	0.735	0.698	0.711	0.674
Llama3.2-3B	0.1	AttnLogDet	✓		0.782	0.868	0.845	0.827	0.824	0.918	0.695	0.843	0.763	**0.749**	0.796	0.678
Llama3.2-3B	0.1	AttnEigvals		✓	0.675	0.782	0.750	0.725	0.755	0.727	0.626	0.792	0.734	0.695	0.724	0.720
Llama3.2-3B	0.1	AttnEigvals	✓		0.814	0.873	0.872	0.852	0.842	0.963	0.723	0.844	0.772	0.744	0.788	0.688
Llama3.2-3B	0.1	LapEigvals		✓	0.681	0.774	0.733	0.708	0.733	0.722	0.676	0.781	0.736	0.697	0.732	0.690
Llama3.2-3B	0.1	LapEigvals	✓		0.831	0.875	0.837	0.832	0.852	0.895	**0.801**	**0.857**	**0.779**	0.736	**0.826**	**0.743**
Llama3.2-3B	1.0	AttentionScore		✓	0.532	0.668	0.588	0.578	0.553	0.555	0.557	0.637	0.592	0.593	0.558	0.675
Llama3.2-3B	1.0	AttentionScore	✓		0.512	0.606	0.554	0.529	0.517	0.484	0.509	0.588	0.546	0.530	0.515	0.581
Llama3.2-3B	1.0	AttnLogDet		✓	0.578	0.738	0.677	0.720	0.716	0.739	0.597	0.724	0.678	0.707	0.711	0.742
Llama3.2-3B	1.0	AttnLogDet	✓		0.784	0.869	0.816	0.839	0.831	0.924	0.700	0.801	0.690	0.734	0.789	**0.795**
Llama3.2-3B	1.0	AttnEigvals		✓	0.642	0.777	0.716	0.747	0.763	0.735	0.641	0.756	0.696	0.703	0.746	0.748
Llama3.2-3B	1.0	AttnEigvals	✓		0.819	0.878	0.847	0.876	0.847	0.978	0.724	0.819	**0.694**	0.749	0.804	0.723
Llama3.2-3B	1.0	LapEigvals		✓	0.695	0.764	0.683	0.719	0.727	0.682	0.715	0.754	0.671	0.711	0.738	0.767
Llama3.2-3B	1.0	LapEigvals	✓		0.842	0.885	0.803	0.850	0.863	0.911	**0.812**	**0.828**	0.693	**0.757**	**0.832**	0.787
Phi3.5	0.1	AttentionScore		✓	0.517	0.559	0.565	0.606	0.625	0.601	0.528	0.551	0.637	0.621	0.628	0.637
Phi3.5	0.1	AttentionScore	✓		0.499	0.538	0.532	0.473	0.539	0.522	0.505	0.511	0.578	0.458	0.534	0.554
Phi3.5	0.1	AttnLogDet		✓	0.583	0.732	0.741	0.711	0.757	0.720	0.585	0.726	0.785	0.726	0.772	0.765
Phi3.5	0.1	AttnLogDet	✓		0.845	0.863	0.905	0.852	0.875	0.981	0.723	0.802	0.802	0.759	0.842	0.716
Phi3.5	0.1	AttnEigvals		✓	0.760	0.781	0.793	0.745	0.802	0.854	0.678	0.764	0.790	0.747	0.791	**0.774**
Phi3.5	0.1	AttnEigvals	✓		0.862	0.867	0.904	0.861	0.881	0.999	0.728	0.802	0.787	0.740	0.838	0.761
Phi3.5	0.1	LapEigvals		✓	0.734	0.758	0.737	0.704	0.775	0.759	0.716	0.757	0.761	0.732	0.768	0.741
Phi3.5	0.1	LapEigvals	✓		0.856	0.860	0.897	0.841	0.884	0.965	**0.810**	**0.819**	**0.815**	**0.791**	**0.858**	0.717
Phi3.5	1.0	AttentionScore		✓	0.499	0.567	0.615	0.626	0.637	0.618	0.533	0.581	0.630	0.645	0.642	0.626
Phi3.5	1.0	AttentionScore	✓		0.489	0.540	0.566	0.469	0.553	0.541	0.520	0.541	0.594	0.504	0.540	0.554
Phi3.5	1.0	AttnLogDet		✓	0.587	0.733	0.773	0.722	0.766	0.753	0.557	0.762	0.784	0.736	0.772	0.763
Phi3.5	1.0	AttnLogDet	✓		0.842	0.868	0.921	0.859	0.879	0.971	0.745	0.818	0.815	0.769	0.848	0.755
Phi3.5	1.0	AttnEigvals		✓	0.755	0.794	0.820	0.790	0.809	0.864	0.710	0.795	0.787	0.752	0.799	0.747
Phi3.5	1.0	AttnEigvals	✓		0.858	0.871	0.924	0.876	0.887	0.998	0.771	0.829	0.798	0.782	0.850	**0.802**
Phi3.5	1.0	LapEigvals		✓	0.733	0.755	0.755	0.718	0.779	0.713	0.723	0.769	0.755	0.732	0.792	0.732
Phi3.5	1.0	LapEigvals	✓		0.856	0.863	0.911	0.849	0.889	0.961	**0.821**	**0.836**	**0.826**	**0.795**	**0.872**	0.777

Table 5: (Part II) Performance comparison of methods evaluated in this work on an extended set of configurations. We marked results for AttentionScore in gray as it is unsupervised approach, not directly comparable to the other ones. In **bold**, we highlight the best performance on the test split of data, individually for each dataset, LLM, and temperature.

LLM	Temp	Feature	*all-layers*	*per-layer*	Train AUROC						Test AUROC					
					CoQA	HaluevalQA	NQOpen	SQuADv2	TriviaQA	TruthfulQA	CoQA	HaluevalQA	NQOpen	SQuADv2	TriviaQA	TruthfulQA
Mistral-Nemo	0.1	AttentionScore		✓	0.504	0.574	0.591	0.509	0.550	0.546	0.515	0.559	0.587	0.527	0.545	0.681
Mistral-Nemo	0.1	AttentionScore	✓		0.508	0.536	0.537	0.507	0.520	0.535	0.484	0.523	0.533	0.495	0.505	0.631
Mistral-Nemo	0.1	AttnLogDet		✓	0.584	0.716	0.702	0.675	0.689	0.744	0.583	0.723	0.688	0.668	0.722	0.731
Mistral-Nemo	0.1	AttnLogDet	✓		0.828	0.842	0.861	0.858	0.854	0.963	0.734	0.786	0.752	0.709	0.822	**0.776**
Mistral-Nemo	0.1	AttnEigvals		✓	0.708	0.751	0.749	0.749	0.747	0.797	0.672	0.740	0.701	0.704	0.738	0.717
Mistral-Nemo	0.1	AttnEigvals	✓		0.845	0.842	0.878	0.864	0.859	0.996	0.768	0.789	0.743	0.716	0.809	0.752
Mistral-Nemo	0.1	LapEigvals		✓	0.763	0.772	0.732	0.723	0.781	0.725	0.759	0.760	0.697	0.696	0.769	0.710
Mistral-Nemo	0.1	LapEigvals	✓		0.868	0.862	0.875	0.869	0.886	0.977	**0.823**	**0.821**	**0.755**	**0.767**	**0.858**	0.737
Mistral-Nemo	1.0	AttentionScore		✓	0.502	0.586	0.606	0.546	0.553	0.570	0.525	0.587	0.588	0.564	0.570	0.632
Mistral-Nemo	1.0	AttentionScore	✓		0.493	0.541	0.552	0.503	0.521	0.531	0.493	0.531	0.529	0.510	0.532	0.494
Mistral-Nemo	1.0	AttnLogDet		✓	0.591	0.723	0.716	0.717	0.717	0.741	0.581	0.730	0.703	0.711	0.707	0.801
Mistral-Nemo	1.0	AttnLogDet	✓		0.829	0.851	0.870	0.860	0.857	0.963	0.728	0.798	0.769	0.772	0.812	**0.852**
Mistral-Nemo	1.0	AttnEigvals		✓	0.704	0.762	0.742	0.757	0.752	0.806	0.670	0.749	0.742	0.719	0.737	0.804
Mistral-Nemo	1.0	AttnEigvals	✓		0.844	0.851	0.893	0.864	0.862	0.996	0.778	0.781	0.761	0.758	0.821	0.802
Mistral-Nemo	1.0	LapEigvals		✓	0.765	0.790	0.749	0.740	0.804	0.779	0.738	0.763	0.708	0.723	0.785	0.818
Mistral-Nemo	1.0	LapEigvals	✓		0.876	0.877	0.884	0.881	0.901	0.978	**0.835**	**0.833**	**0.795**	**0.812**	**0.865**	0.828
Mistral-Small-24B	0.1	AttentionScore		✓	0.520	0.538	0.517	0.577	0.535	0.571	0.525	0.552	0.592	0.625	0.533	0.724
Mistral-Small-24B	0.1	AttentionScore	✓		0.520	0.472	0.449	0.510	0.449	0.491	0.493	0.493	0.467	0.556	0.461	0.645
Mistral-Small-24B	0.1	AttnLogDet		✓	0.585	0.674	0.659	0.724	0.685	0.698	0.586	0.684	0.695	0.752	0.682	0.721
Mistral-Small-24B	0.1	AttnLogDet	✓		0.851	0.817	0.799	0.820	0.861	0.898	0.762	0.760	0.725	0.763	0.778	0.767
Mistral-Small-24B	0.1	AttnEigvals		✓	0.734	0.722	0.667	0.745	0.757	0.732	0.720	0.707	0.697	0.773	0.758	0.765
Mistral-Small-24B	0.1	AttnEigvals	✓		0.872	0.873	0.923	0.903	0.899	0.993	0.793	0.771	**0.731**	0.803	0.809	0.796
Mistral-Small-24B	0.1	LapEigvals		✓	0.802	0.720	0.646	0.714	0.742	0.694	0.800	0.719	0.674	0.784	0.757	**0.827**
Mistral-Small-24B	0.1	LapEigvals	✓		0.887	0.870	0.901	0.887	0.905	0.979	**0.852**	**0.808**	0.722	**0.821**	**0.831**	0.757
Mistral-Small-24B	1.0	AttentionScore		✓	0.511	0.555	0.582	0.561	0.562	0.542	0.535	0.566	0.576	0.567	0.574	0.606
Mistral-Small-24B	1.0	AttentionScore	✓		0.497	0.503	0.463	0.519	0.451	0.493	0.516	0.504	0.462	0.455	0.463	0.451
Mistral-Small-24B	1.0	AttnLogDet		✓	0.591	0.727	0.710	0.732	0.720	0.677	0.600	0.771	0.714	0.726	0.734	0.687
Mistral-Small-24B	1.0	AttnLogDet	✓		0.850	0.847	0.827	0.856	0.853	0.877	0.766	0.842	0.747	0.753	0.833	0.735
Mistral-Small-24B	1.0	AttnEigvals		✓	0.757	0.743	0.728	0.764	0.779	0.741	0.723	0.780	0.733	0.734	0.780	0.718
Mistral-Small-24B	1.0	AttnEigvals	✓		0.877	0.878	0.923	0.911	0.895	0.997	0.805	0.848	0.751	0.760	0.844	**0.765**
Mistral-Small-24B	1.0	LapEigvals		✓	0.814	0.762	0.733	0.790	0.766	0.703	0.805	0.790	0.712	0.781	0.779	0.725
Mistral-Small-24B	1.0	LapEigvals	✓		0.895	0.890	0.898	0.910	0.907	0.965	**0.861**	**0.882**	**0.791**	**0.820**	**0.876**	0.748

Table 6: Values of k hyperparameter, denoting how many highest eigenvalues are taken from the Laplacian matrix, corresponding to the best results in Table 1 and Table 4.

LLM	Temp	Feature	*all-layers*	*per-layer*	top-k eigenvalues					
					CoQA	HaluevalQA	NQOpen	SQuADv2	TriviaQA	TruthfulQA
Llama3.1-8B	0.1	AttnEigvals		✓	50	100	25	100	100	10
Llama3.1-8B	0.1	AttnEigvals	✓		100	100	100	100	50	100
Llama3.1-8B	0.1	LapEigvals		✓	50	100	10	100	100	100
Llama3.1-8B	0.1	LapEigvals	✓		10	100	100	100	100	100
Llama3.1-8B	1.0	AttnEigvals		✓	100	100	100	100	100	100
Llama3.1-8B	1.0	AttnEigvals	✓		100	100	100	100	100	100
Llama3.1-8B	1.0	LapEigvals		✓	100	100	100	100	100	100
Llama3.1-8B	1.0	LapEigvals	✓		100	25	100	100	100	100
Llama3.2-3B	0.1	AttnEigvals		✓	100	100	100	100	100	10
Llama3.2-3B	0.1	AttnEigvals	✓		100	25	100	100	100	100
Llama3.2-3B	0.1	LapEigvals		✓	100	100	100	100	50	5
Llama3.2-3B	0.1	LapEigvals	✓		25	100	100	100	100	100
Llama3.2-3B	1.0	AttnEigvals		✓	100	100	100	100	100	50
Llama3.2-3B	1.0	AttnEigvals	✓		100	100	100	100	100	100
Llama3.2-3B	1.0	LapEigvals		✓	100	100	10	100	100	25
Llama3.2-3B	1.0	LapEigvals	✓		25	100	100	100	100	100
Phi3.5	0.1	AttnEigvals		✓	100	100	100	100	100	100
Phi3.5	0.1	AttnEigvals	✓		100	10	10	25	100	50
Phi3.5	0.1	LapEigvals		✓	100	100	100	100	100	100
Phi3.5	0.1	LapEigvals	✓		10	50	100	100	100	100
Phi3.5	1.0	AttnEigvals		✓	100	100	100	100	100	100
Phi3.5	1.0	AttnEigvals	✓		100	100	10	100	100	50
Phi3.5	1.0	LapEigvals		✓	100	100	100	100	100	50
Phi3.5	1.0	LapEigvals	✓		10	100	100	100	100	100
Mistral-Nemo	0.1	AttnEigvals		✓	100	100	100	100	100	100
Mistral-Nemo	0.1	AttnEigvals	✓		100	100	100	100	100	100
Mistral-Nemo	0.1	LapEigvals		✓	100	100	100	100	100	10
Mistral-Nemo	0.1	LapEigvals	✓		10	25	100	50	100	100
Mistral-Nemo	1.0	AttnEigvals		✓	100	100	100	100	100	100
Mistral-Nemo	1.0	AttnEigvals	✓		100	100	100	100	50	100
Mistral-Nemo	1.0	LapEigvals		✓	100	100	50	100	100	100
Mistral-Nemo	1.0	LapEigvals	✓		10	50	100	100	100	100
Mistral-Small-24B	0.1	AttnEigvals		✓	100	100	10	100	50	25
Mistral-Small-24B	0.1	AttnEigvals	✓		100	100	100	100	100	25
Mistral-Small-24B	0.1	LapEigvals		✓	100	100	50	100	100	10
Mistral-Small-24B	0.1	LapEigvals	✓		25	100	100	100	10	100
Mistral-Small-24B	1.0	AttnEigvals		✓	100	100	100	100	100	100
Mistral-Small-24B	1.0	AttnEigvals	✓		100	100	100	100	100	100
Mistral-Small-24B	1.0	LapEigvals		✓	100	100	100	50	100	50
Mistral-Small-24B	1.0	LapEigvals	✓		10	50	10	10	100	50

Table 7: Values of a layer index (numbered from 0) corresponding to the best results for *per-layer* models in Table 4.

LLM	$temp$	Feature	Layer index					
			CoQA	HaluevalQA	NQOpen	SQuADv2	TriviaQA	TruthfulQA
Llama3.1-8B	0.1	AttentionScore	13	10	0	0	0	28
Llama3.1-8B	0.1	AttnLogDet	7	13	16	11	29	21
Llama3.1-8B	0.1	AttnEigvals	22	31	26	31	31	7
Llama3.1-8B	0.1	LapEigvals	15	14	20	29	31	20
Llama3.1-8B	1.0	AttentionScore	29	10	0	0	0	23
Llama3.1-8B	1.0	AttnLogDet	17	11	13	29	29	30
Llama3.1-8B	1.0	AttnEigvals	22	31	31	31	31	31
Llama3.1-8B	1.0	LapEigvals	15	14	31	29	29	29
Llama3.2-3B	0.1	AttentionScore	15	12	12	12	21	14
Llama3.2-3B	0.1	AttnLogDet	12	13	24	10	25	14
Llama3.2-3B	0.1	AttnEigvals	27	14	14	25	27	17
Llama3.2-3B	0.1	LapEigvals	11	8	12	25	12	14
Llama3.2-3B	1.0	AttentionScore	24	12	0	24	21	14
Llama3.2-3B	1.0	AttnLogDet	12	26	23	25	25	12
Llama3.2-3B	1.0	AttnEigvals	11	27	25	25	27	10
Llama3.2-3B	1.0	LapEigvals	11	18	12	25	25	11
Phi3.5	0.1	AttentionScore	7	15	0	0	0	19
Phi3.5	0.1	AttnLogDet	20	18	16	17	13	23
Phi3.5	0.1	AttnEigvals	18	19	15	19	18	28
Phi3.5	0.1	LapEigvals	18	28	28	19	31	28
Phi3.5	1.0	AttentionScore	19	0	1	0	0	19
Phi3.5	1.0	AttnLogDet	12	29	14	19	13	14
Phi3.5	1.0	AttnEigvals	18	30	17	31	31	31
Phi3.5	1.0	LapEigvals	18	28	15	19	31	31
Mistral-Nemo	0.1	AttentionScore	2	18	35	0	30	35
Mistral-Nemo	0.1	AttnLogDet	37	17	15	38	38	33
Mistral-Nemo	0.1	AttnEigvals	38	38	18	18	15	31
Mistral-Nemo	0.1	LapEigvals	16	37	37	18	37	8
Mistral-Nemo	1.0	AttentionScore	10	16	28	14	30	21
Mistral-Nemo	1.0	AttnLogDet	18	20	18	18	15	18
Mistral-Nemo	1.0	AttnEigvals	38	39	39	18	15	18
Mistral-Nemo	1.0	LapEigvals	16	37	37	18	37	18
Mistral-Small-24B	0.1	AttentionScore	14	39	33	35	0	30
Mistral-Small-24B	0.1	AttnLogDet	16	38	18	16	38	11
Mistral-Small-24B	0.1	AttnEigvals	36	36	19	16	38	20
Mistral-Small-24B	0.1	LapEigvals	21	35	24	36	35	34
Mistral-Small-24B	1.0	AttentionScore	15	1	0	1	0	30
Mistral-Small-24B	1.0	AttnLogDet	14	27	17	24	38	34
Mistral-Small-24B	1.0	AttnEigvals	36	27	21	24	36	23
Mistral-Small-24B	1.0	LapEigvals	21	36	16	21	35	34

Table 8: Results of the probe trained on the hidden state features from the last generated token.

LLM	Temp	Features	*per-layer*	*all-layers*	Test AUROC (↑)					
					CoQA	HaluevalQA	NQOpen	SQuADv2	TriviaQA	TruthfulQA
Llama3.1-8B	0.1	HiddenStates	✓		0.835	0.840	0.766	0.736	0.820	**0.834**
Llama3.1-8B	0.1	HiddenStates		✓	0.821	0.825	0.728	0.723	0.791	0.785
Llama3.1-8B	0.1	LapEigvals	✓		0.757	0.793	0.711	0.733	0.780	0.764
Llama3.1-8B	0.1	LapEigvals		✓	**0.836**	**0.867**	**0.793**	**0.782**	**0.872**	0.822
Llama3.1-8B	1.0	HiddenStates	✓		**0.836**	0.850	0.786	0.754	0.850	0.823
Llama3.1-8B	1.0	HiddenStates		✓	0.835	0.847	0.757	0.749	0.838	0.808
Llama3.1-8B	1.0	LapEigvals	✓		0.743	0.789	0.725	0.724	0.794	0.764
Llama3.1-8B	1.0	LapEigvals		✓	0.830	**0.874**	**0.827**	**0.791**	**0.889**	**0.829**
Llama3.2-3B	0.1	HiddenStates	✓		0.800	0.808	0.732	0.750	0.782	0.760
Llama3.2-3B	0.1	HiddenStates		✓	0.790	0.784	0.709	0.721	0.760	**0.770**
Llama3.2-3B	0.1	LapEigvals	✓		0.676	0.774	0.730	0.727	0.712	0.690
Llama3.2-3B	0.1	LapEigvals		✓	**0.801**	**0.844**	**0.771**	**0.778**	**0.821**	0.743
Llama3.2-3B	1.0	HiddenStates	✓		0.778	0.758	0.679	0.719	0.773	0.716
Llama3.2-3B	1.0	HiddenStates		✓	0.773	0.753	0.657	0.681	0.761	0.618
Llama3.2-3B	1.0	LapEigvals	✓		0.715	0.765	0.696	0.696	0.738	0.767
Llama3.2-3B	1.0	LapEigvals		✓	**0.812**	**0.857**	**0.798**	**0.751**	**0.836**	**0.787**
Phi3.5	0.1	HiddenStates	✓		**0.841**	**0.845**	0.813	0.781	**0.886**	0.737
Phi3.5	0.1	HiddenStates		✓	0.833	0.840	0.806	0.774	0.878	0.689
Phi3.5	0.1	LapEigvals	✓		0.716	0.757	0.761	0.732	0.768	**0.741**
Phi3.5	0.1	LapEigvals		✓	0.810	0.819	**0.815**	**0.791**	0.858	0.717
Phi3.5	1.0	HiddenStates	✓		**0.872**	**0.850**	0.821	0.806	**0.891**	**0.822**
Phi3.5	1.0	HiddenStates		✓	0.853	0.844	0.804	0.790	0.887	0.752
Phi3.5	1.0	LapEigvals	✓		0.723	0.769	0.755	0.732	0.792	0.732
Phi3.5	1.0	LapEigvals		✓	0.821	0.836	**0.826**	0.795	0.872	0.777
Mistral-Nemo	0.1	HiddenStates	✓		0.818	0.814	0.734	0.731	0.821	**0.792**
Mistral-Nemo	0.1	HiddenStates		✓	0.805	0.784	0.722	0.730	0.793	0.699
Mistral-Nemo	0.1	LapEigvals	✓		0.759	0.760	0.697	0.696	0.769	0.710
Mistral-Nemo	0.1	LapEigvals		✓	**0.823**	**0.821**	**0.755**	**0.767**	**0.858**	0.737
Mistral-Nemo	1.0	HiddenStates	✓		0.793	0.777	0.738	0.719	0.783	0.722
Mistral-Nemo	1.0	HiddenStates		✓	0.771	0.771	0.706	0.685	0.779	0.644
Mistral-Nemo	1.0	LapEigvals	✓		0.738	0.763	0.708	0.723	0.785	0.818
Mistral-Nemo	1.0	LapEigvals		✓	**0.835**	**0.833**	**0.795**	**0.812**	**0.865**	**0.828**
Mistral-Small-24B	0.1	HiddenStates	✓		0.838	0.744	0.680	0.700	0.749	0.735
Mistral-Small-24B	0.1	HiddenStates		✓	0.815	0.703	0.632	0.629	0.726	0.589
Mistral-Small-24B	0.1	LapEigvals	✓		0.800	0.719	0.674	0.784	0.757	**0.827**
Mistral-Small-24B	0.1	LapEigvals		✓	**0.852**	**0.808**	**0.722**	**0.821**	**0.831**	0.757
Mistral-Small-24B	1.0	HiddenStates	✓		0.801	0.720	0.665	0.603	0.684	0.581
Mistral-Small-24B	1.0	HiddenStates		✓	0.770	0.703	0.617	0.575	0.659	0.485
Mistral-Small-24B	1.0	LapEigvals	✓		0.805	0.790	0.712	0.781	0.779	0.725
Mistral-Small-24B	1.0	LapEigvals		✓	**0.861**	**0.882**	**0.791**	**0.820**	**0.876**	**0.748**

Table 9: Full results of the generalisation study. By gray color we denote results obtained on test split from the same QA dataset as training split, otherwise results are from test split of different QA dataset. We highlight the best performance in **bold**.

Feature	Train Dataset	Test AUROC (↑)					
		CoQA	HaluevalQA	NQOpen	SQuADv2	TriviaQA	TruthfulQA
AttnLogDet	CoQA	0.758	0.687	0.644	0.646	0.640	0.587
AttnEigvals	CoQA	0.782	0.726	0.696	0.659	0.702	0.560
LapEigvals	CoQA	0.830	**0.790**	**0.748**	**0.743**	**0.786**	**0.629**
AttnLogDet	HaluevalQA	0.580	0.823	0.750	0.727	0.787	0.668
AttnEigvals	HaluevalQA	0.579	0.819	0.792	0.743	0.803	**0.688**
LapEigvals	HaluevalQA	**0.685**	0.873	**0.796**	**0.778**	**0.848**	0.595
AttnLogDet	NQOpen	0.552	0.720	0.794	0.717	0.766	0.597
AttnEigvals	NQOpen	0.546	0.725	0.790	0.714	0.770	**0.618**
LapEigvals	NQOpen	**0.656**	**0.792**	0.827	**0.748**	**0.843**	0.564
AttnLogDet	SQuADv2	0.553	0.716	0.774	0.746	0.757	0.658
AttnEigvals	SQuADv2	0.576	0.730	0.737	0.768	0.760	**0.711**
LapEigvals	SQuADv2	**0.673**	**0.801**	**0.806**	0.791	**0.841**	0.625
AttnLogDet	TriviaQA	0.565	0.761	0.793	0.736	0.838	0.572
AttnEigvals	TriviaQA	0.577	0.770	0.786	0.742	0.843	**0.616**
LapEigvals	TriviaQA	**0.702**	**0.813**	**0.818**	**0.773**	0.889	0.522
AttnLogDet	TruthfulQA	0.550	0.597	**0.603**	0.604	0.662	0.811
AttnEigvals	TruthfulQA	0.538	**0.600**	0.595	**0.646**	**0.685**	0.833
LapEigvals	TruthfulQA	**0.590**	0.552	0.529	0.569	0.631	0.829

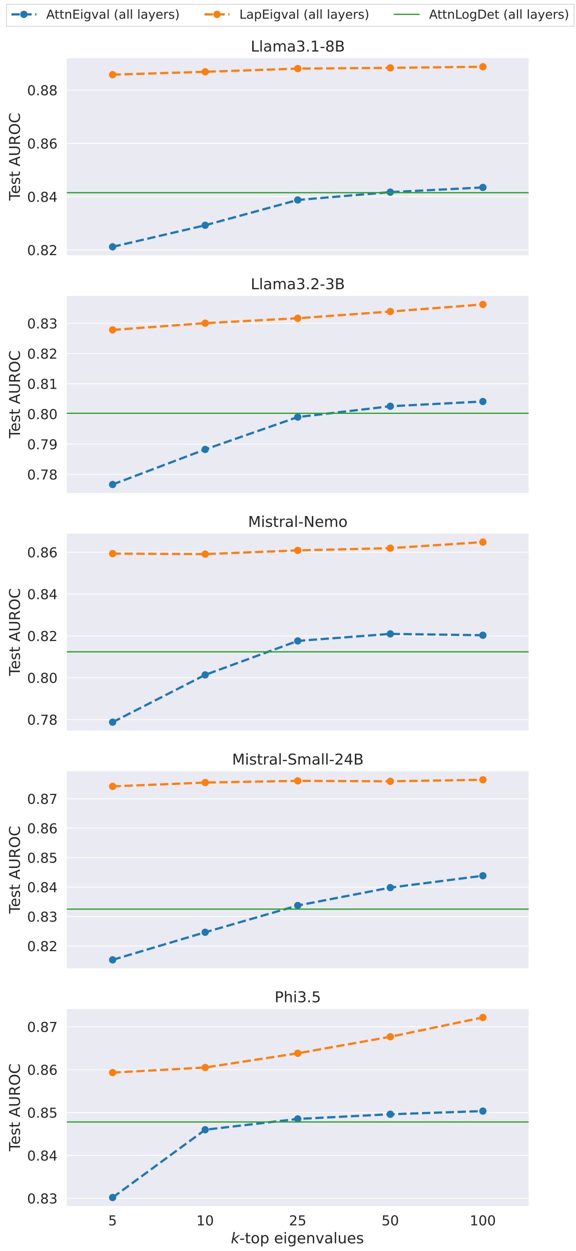


Figure 10: Probe performance across different top-k eigenvalues: $k \in \{5, 10, 25, 50, 100\}$ for TriviQA dataset with $temp = 1.0$ and 5 considered LLMs.

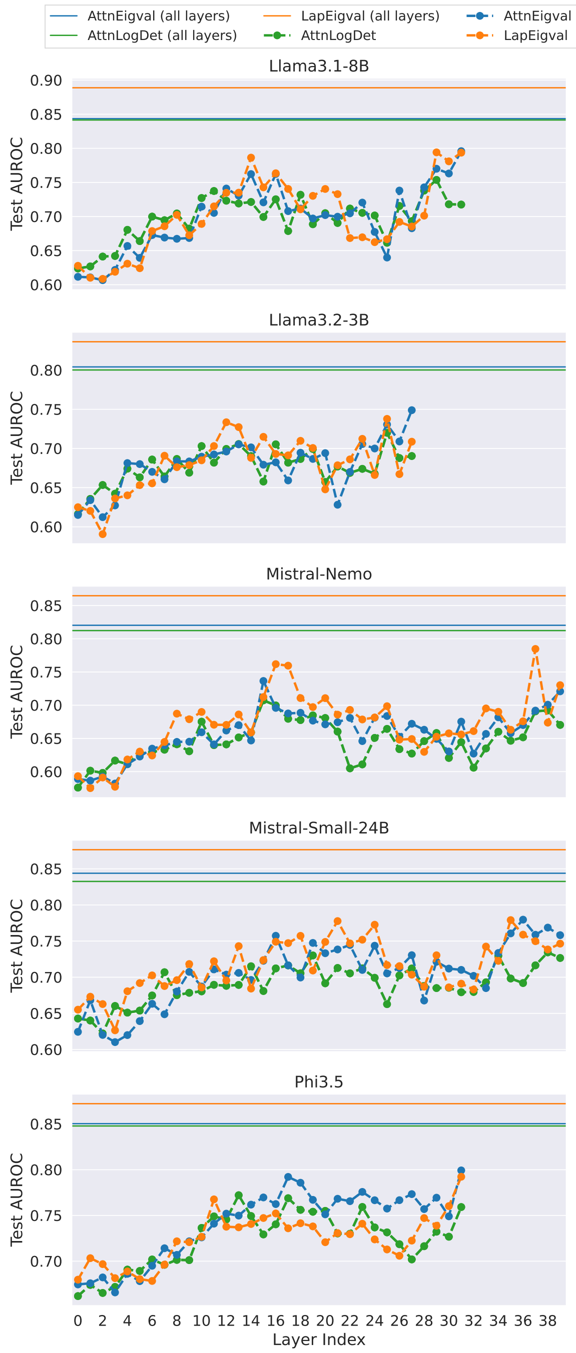


Figure 11: Analysis of model performance across different layers for and 5 considered LLMs and TriviaQA dataset with $temp = 1.0$ and $k = 100$ top eigenvalues (results for models operating on all layers provided for reference).

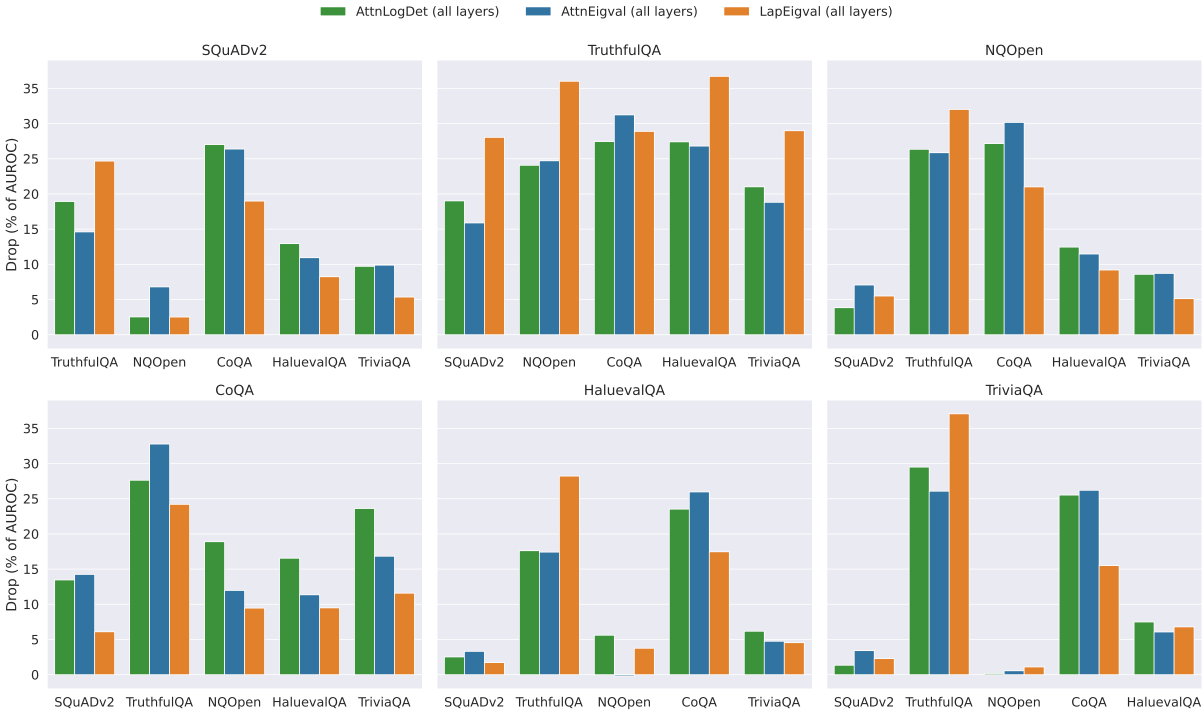


Figure 12: Generalisation across datasets measured as a per cent performance drop in Test AUROC (less is better) when trained on one dataset and tested on the other. Training datasets are indicated in the plot titles, while test datasets are shown on the x-axis. Results computed on `Llama-3.1-8B` with $k = 100$ top eigenvalues and $temp = 1.0$.

Listing 1: One-shot QA (prompt p_1)

```
Deliver a succinct and straightforward answer to the question below. Focus on being
    ↪ brief while maintaining essential information. Keep extra details to a
    ↪ minimum.

Here is an example:
Question: What is the Riemann hypothesis?
Answer: All non-trivial zeros of the Riemann zeta function have real part 1/2

Question: {question}
Answer:
```

Listing 2: Zero-sho QA (prompt p_2).

```
Please provide a concise and direct response to the following question, keeping your
    ↪  answer as brief and to-the-point as possible while ensuring clarity. Avoid
    ↪ any unnecessary elaboration or additional details.
Question: {question}
Answer:
```

Listing 3: Few-shot QA prompt (prompt p_3), modified version of prompt used by (Kossen et al., 2024).

```
Answer the following question as briefly as possible.
Here are several examples:
Question: What is the capital of France?
Answer: Paris

Question: Who wrote *Romeo and Juliet*?
Answer: William Shakespeare

Question: What is the boiling point of water in Celsius?
Answer: 100°C

Question: How many continents are there on Earth?
Answer: Seven

Question: What is the fastest land animal?
Answer: Cheetah

Question: {question}
Answer:
```

Listing 4: Zero-shot shor QA prompt (prompt p_4).

```
Answer the following question as briefly as possible.
Question: {question}
Answer:
```

Listing 5: Prompt used in *llm-as-judge* approach for determining halucination labels. Prompt is a modified version of the one used by (Orgad et al., 2025).

```
You will evaluate answers to questions. For each question, I will provide a model's
    ↪ answer and one or more correct reference answers.
You would have to determine if the model answer is correct, incorrect, or model
    ↪ refused to answer. The model answer to be correct has to match from one to
    ↪ all of the possible correct answers.
If the model answer is correct, write 'correct' and if it is not correct, write '
    ↪ incorrect'. If the Model Answer is a refusal, stating that they don't have
    ↪ enough information, write 'refuse'.
For example:

Question: who is the young guitarist who played with buddy guy?
Ground Truth: [Quinn Sullivan, Eric Gales]
Model Answer: Ronnie Earl
Correctness: incorrect

Question: What is the name of the actor who plays Iron Man in the Marvel movies?
Ground Truth: [Robert Downey Jr.]
Model Answer: Robert Downey Jr. played the role of Tony Stark/Iron Man in the Marvel
    ↪  Cinematic Universe films.
Correctness: correct

Question: what is the capital of France?
Ground Truth: [Paris]
Model Answer: I don't have enough information to answer this question.
Correctness: refuse

Question: who was the first person to walk on the moon?
Ground Truth: [Neil Armstrong]
Model Answer: I apologise, but I cannot provide an answer without verifying the
    ↪ historical facts.
Correctness: refuse

Question: {{question}}
Ground Truth: {{gold_answer}}
Model Answer: {{predicted_answer}}
Correctness:
```