Towards improving open-box hallucination detection in large language models (LLMs).

SURESH, M., ALJUNDI, R., NKISI-ORJI, I. and WIRATUNGA, N.

2024

Towards Improving Open-box Hallucination Detection in Large Language Models (LLMs)

Malavika Suresh[1,*], Rahaf Aljundi[2], Ikechukwu Nkisi-Orji[3] and Nirmalie Wiratunga[4]

[1]Robert Gordon University (RGU), Aberdeen, Scotland
[2]Toyota-Motor-Europe (TME), Brussels, Belgium
[3]Robert Gordon University (RGU), Aberdeen, Scotland
[4]Robert Gordon University (RGU), Aberdeen, Scotland

Abstract

Due to the increasing availability of Large Language Models (LLMs) through both proprietary and open-sourced releases of models, the adoption of LLMs across applications has drastically increased making them commonplace in day-to-day lives. Yet, the problem of detecting and mitigating hallucinations in these models remains an open challenge. This work considers the problem of open-box hallucination detection, i.e., detecting hallucinations when there is full access to the generation process. Recent work has shown that simple binary probes constructed on the model activation space can act as reliable hallucination detectors. This work extends probing-based detection methods by considering the activation space at multiple layers, components and token positions during generation. Experiments are conducted across two LLMs and three open-domain fact recall datasets. The results indicate that hallucinations can be detected at various layers as well as token positions during the generation process. This indicates the potential for saving compute costs through early detection as well as for improving detection performance by designing more sophisticated probing methods.

Keywords

Large Language Models (LLMs), Hallucination, Model Probing

1. Introduction

With Large Language Models (LLMs) becoming increasingly accessible, many researchers are now focusing on the problem of detecting and mitigating model *hallucinations* - i.e. the tendency to produce factually inaccurate text. Solving the problem of hallucinations is of high importance not only to prevent the spread of misinformation in this era of LLM-based chatbots and search engines but also to ensure safety when deploying to sensitive applications such as therapeutic chatbots, where LLMs have the potential to create high impact. The widespread adoption of proprietary API-based LLMs has led to the development of *black-box* and *grey-box* methods of hallucination detection and mitigation, which do not require access to the underlying LLM. *Black-box* methods [1, 2, 3] rely on additional LLM prompting, while *grey-box* methods [1, 4, 5] make use of generated token probabilities, where available, to quantify the uncertainty of generated output. Meanwhile, the release of several open-sourced LLMs has also recently motivated the development of *open-box* methods [6, 7, 8], which probe and modify the internal operations during the generation process.

Detection and mitigation using additional LLM prompting, while simple to implement, adds high compute cost and increases latency at inference time. Uncertainty quantification methods, while overcoming these shortcomings, cannot detect confident model hallucinations. With access to model internals, open-box methods have the potential to address the problems of compute efficiency, inference latency as well as model overconfidence. This work identifies and addresses open research questions in open-box hallucination detection. Recent work [6, 9] has shown that binary classifiers (probes) built on the model activation space are good hallucination detectors. While these methods typically consider the activation space at the output of the transformer block, model editing literature [10] has shown that

SICSA REALLM Workshop 2024

[*]Corresponding author.

m.suresh@rgu.ac.uk (M. Suresh); rahaf.al.jundi@toyota-europe.com (R. Aljundi); i.nkisi-orji@rgu.ac.uk (I. Nkisi-Orji); n.wiratunga@rgu.ac.uk (N. Wiratunga)

feed-forward components play a crucial role in fact recall, which raises the question of whether better detection performance can be achieved by probing activations at the level of individual transformer components, especially for factual hallucinations. Similarly, probing attention head activations has been shown to be useful for separating 'truthful' and 'non-truthful' statements [7], indicating that some hallucinations may be detected at attention heads. This work extends recent probing-based hallucination detection methods with the goal of investigating: (1) how can model activations at multiple layers be combined to improve hallucination detection? (2) at which transformer model components and token positions can hallucinations be best detected?

Experiments are conducted on two LLMs, namely Llama-7B and its instruction fine-tuned version Alpaca-7B. Results on three factual question-answering datasets demonstrate that hallucinations can often be detected at the output of multiple model components (i.e. attention head outputs, feed-forward outputs) and token positions during decoding, highlighting the need for further research into sophisticated probing methods to improve open-box hallucination detection. Reasonable detection performance is observed even as early as the generation of the first token of the response, which can help save compute costs in many practical applications.

Section 2 provides an overview of recent methods of hallucination detection and mitigation, introducing relevant notations and equations. The experiments conducted are described in section 3, followed by a discussion of results in section 4. Finally, section 5 concludes the work and provides directions for future research.

2. Related Work

2.1. Uncertainty estimation (grey-box)

This category of approaches utilises the output token probabilities to estimate the uncertainty of a generated sequence. For a given input prompt, generations with lower uncertainty scores are considered to be less hallucinatory. Let x denote an input prompt to an LLM, for which M responses can be sampled from the LLM[1], denoted as $s^1, s^2, ..., s^M$. Broadly, uncertainty can be estimated either for each individual sample s^m or for the entire sampled space of M responses. The former can be used to identify the best response for the prompt as the sample with least uncertainty, while the latter determines whether the LLM is capable of generating any appropriate response at all for the prompt. In other words, prompts that an LLM is able to reliably provide responses for should have low uncertainty in the sampled space, with appropriate and non-hallucinatory responses having the least individual sample uncertainty.

$$\text{Sequence probability of individual sample: } p(s^m|x) = -\prod_i p(s_i^m|s_{<i}^m, x) \tag{1}$$

$$\text{Predictive entropy of individual sample: } PE(s^m|x) = -\sum_i \log p(s_i^m|s_{<i}^m, x) \tag{2}$$

$$\text{Entropy of sampled space: } E(x) = -\frac{1}{M}\sum_{m=1}^{M} \log p(s^m|x) \tag{3}$$

Equations 2 and 3 describe the traditional entropy measures for an individual sample and the sampled space, respectively. Recent works improve upon these measures to take into account relevance of individual tokens [5] and semantic similarity in sampled responses [4]. Equations describing these methods are provided in appendix A. In practice, the major drawbacks with using uncertainty estimates for hallucination detection are the need for a robust validation set to identify an *uncertainty threshold* and the difficulty in achieving a good trade-off between precision and recall. Further, model uncertainty can either be *epistemic*, which indicates lack of relevant knowledge in the model, or *aleotoric*, which indicates inherent uncertainty in the ground truth response space such as when multiple diverse responses can be valid. While the measures mentioned above measure uncertainty as a whole, [11] propose a method that measures only the *epistemic* uncertainty instead.

[1] When using greedy decoding, M=1

2.2. Prompt-based approaches (black-box)

This category of approaches is based on the impressive reasoning performance of LLMs with in-context learning [12] and chain-of-thought prompting [3]. Several works [1, 2, 13] use additional LLM prompting to detect contradictions in originally generated outputs. However, adapting the prompt to specific applications (i.e. prompt-engineering) is not straight-forward and also computationally expensive, since only larger models exhibit reasonable performance with prompt-based detection [1]. In fact, using prompts as a self-correction mechanism has even been shown to be detrimental to the original performance [14], whereby models modify originally correct responses into responses with hallucinations.

2.3. Decoding strategies (open-box)

This category of approaches aims to detect and mitigate hallucinations at generation time through probing and modifications respectively, of the internal transformer operations. Since the detection and modification operations are applied during the forward pass of the prompt at inference time, these approaches have much lower computational cost than some uncertainty approaches that require multiple output samples or prompt-based approaches that require multiple forward passes.

Preliminaries Let x_{l-1} denote the output of layer $l-1$ of the transformer, where individual tokens are represented as $\{x^0_{l-1}, x^1_{l-1}...x^t_{l-1}...x^T_{l-1}\}$. Equations (4) - (7) describe various operations in the subsequent transformer layer l. Equation (4) represents the multi-head-attention component, where Att denotes the attention operation and P and Q represent projection to and from the attention head dimensions. Equation (5) represents the multi-layer perceptron (MLP) operation followed by residual connection to give the final layer activation. Equations (6) and (7) represent the activation projection to vocabulary space to give the next token probability distribution.

$$\text{Attention output: } x_l^{`} = x_{l-1} + \sum_{h=1}^{H} Q_l^h Att_l^h (P_l^h x_{l-1}) \tag{4}$$

$$\text{Layer representation/activation: } x_l = x_l^{`} + MLP(x_l^{`}) \tag{5}$$

$$\text{Distribution over vocab (Layer logits): } \phi(x_l) \tag{6}$$

$$\text{Softmax output: } q_l = \text{Softmax}(\phi(x_l)) \tag{7}$$

Detection Prior work in probing have primarily investigated the layer activations. CCS [9] consists of a linear projection followed by a sigmoid classifier and takes as input the activations of the last input token at the last layer. SAPLMA [6] consists of a a three-layer feed-forward network with ReLu activations followed by a sigmoid classifier and takes as input the activations of the last input token at a given layer.

Mitigation The ITI [7] approach is based on empirical evidence that the attention head activation space contains directions that are correlated with the 'truthfulness' of an input text [9]. The notion of 'truthfulness', as measured in this work, relates to common human misconceptions and as such, is a special case of factual hallucinations. Given a labelled dataset of correct and false answers to a set of questions and the model activations for all answers at each attention head, a linear binary classifier (probe) is trained on the last token activations to separate the correct and false answers (with one probing classifier per attention head). The direction learnt by this probe is then used to shift the activations at inference time, at the top k heads that achieved the highest probing accuracy, with a hyper-parameter α to control the intervention strength. CAD [15] modifies the generation probability of output tokens by contrasting the probability distributions obtained with and without adding an input context to the prompt. The contrast operation encourages the model to generate tokens which are aligned with the input context as opposed to relying on the internal knowledge stored in the model,

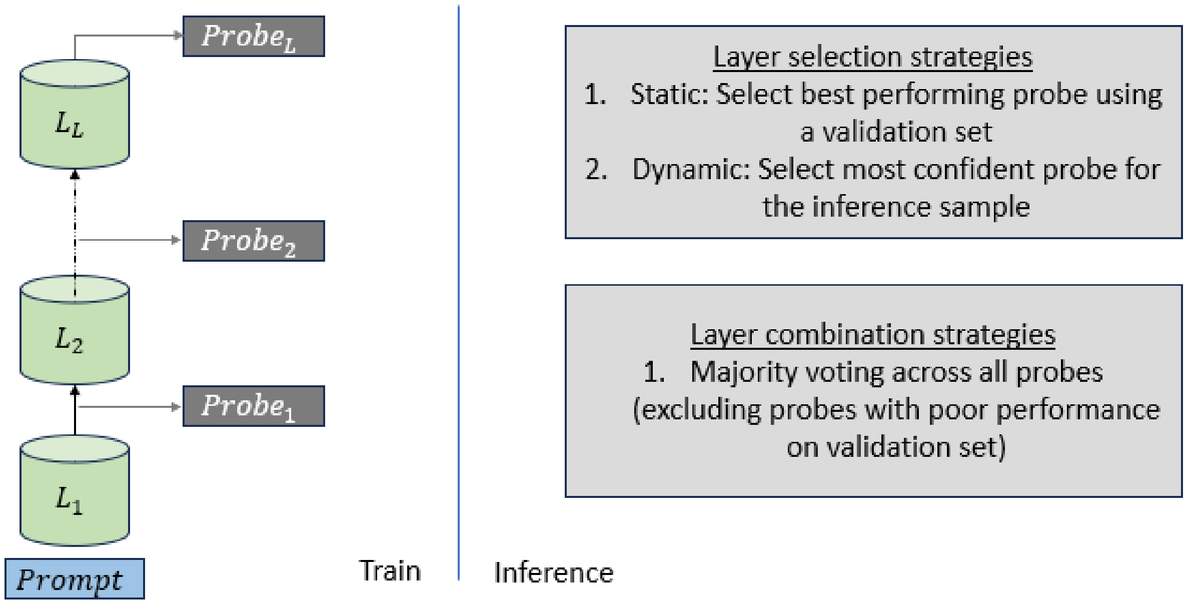

$Probe_L$
L_L
$Probe_2$
L_2
$Probe_1$
L_1
Prompt
Train
Inference
Layer selection strategies
1. Static: Select best performing probe using a validation set
2. Dynamic: Select most confident probe for the inference sample
Layer combination strategies
1. Majority voting across all probes (excluding probes with poor performance on validation set)

Figure 1: Detecting hallucination across layers: At train time, a binary classifier (probe) is constructed at the output of each transformer layer. At inference time, three simple strategies are explored for selecting either a single probe (*layer selection*) or an ensemble of probes (*layer combination*) for hallucination classification.

which is shown to be particularly useful when the context information contradicts the stored knowledge. DoLa [8] builds on the early exit strategy work [16] and modifies the generation probability of output tokens by contrasting the original output probability distribution of each token against the distribution obtained by projecting inner transformer layer outputs on to the vocabulary (i.e. early exit). Specifically, at each token position t, the layer l with maximum distribution divergence from the output layer L is selected for contrast, which is then performed as a subtraction of log probabilities. The approach is based on the observation that factual information tokens are decided at higher layers and shows that contrasting against such layers improves factuality. Unlike CAD which focuses on incorporating new factual information, DoLa elicits factual information that is already stored in the model. Opera [17] modifies the final layer logits to introduce a penalty term when attending to summary tokens to prevent hallucinations in long-form generations. Equations describing these methods are provided in the appendix B.

2.4. Model editing (open-box)

The field of model editing aims to identify and update facts or information stored in the *weights* of a pre-trained transformer model. This line of work [10, 18] permanently modifies the underlying model and is thus orthogonal to work in decoding strategies, where updates are performed on the *activations* directly at inference time.

2.5. Fine-tuning and reinforcement learning with human feedback (open-box)

This category of approaches fine-tunes a model, often using reinforcement learning with human feedback [19, 20], to improve the truthfulness of model responses and encourage abstention when the model is unable to produce a valid response. Fine-tuning large language model is computationally intensive and also requires a large training dataset, making it infeasible for many practical applications where data collection is difficult.

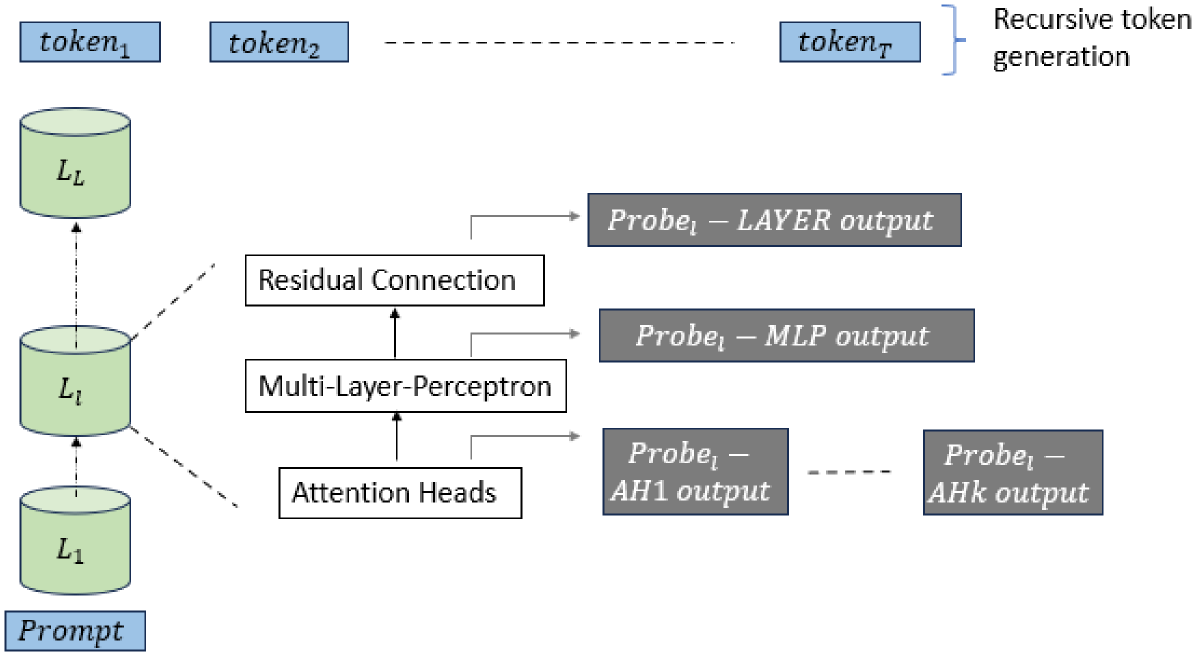


Figure 2: Detecting hallucination across components and tokens: At each model layer, the output of the attention heads and MLP are probed in addition to the final layer output. As tokens are generated recursively, multiple token locations are also considered for probing.

3. Method - Probing Experiments

This section describes the probing experiments conducted to investigate the hallucination detection capability at various points during the generation process[2]. For the first setup, L binary classifiers (i.e. probes) are trained per LLM in a supervised manner with hallucination/non-hallucination labels, where L denotes the number of transformer layers in the model, as shown in figure 1. Each probe takes as input the output activations at the corresponding model layer l for the last generated token (x_l^T). At inference time, three strategies are explored for probe selection: (1)**Most Accurate (MA)**: selects the probe at the most accurate layer, i.e., layer with the best performance[3] on an in-distribution validation set (2)**Most Confident (MC)**: selects the probe with the most confident prediction for the test sample (3)**Majority Vote (MV)**: takes a majority vote across all probes. Probes that always predict the same class on the validation set, if any, are excluded to ensure that we only consider layers where hallucinating and non-hallucinating activations are separable.

For the second setup, probes are constructed at three locations within each transformer layer, as shown in figure 2: (1)**LAY**: at the output of the layer (x_l^T) (2)**MLP**: at the output of the MLP ($MLP(x_l^{T`})$) (3)**AH**: at the output of each attention head ($Att_l^h(P_l^h x_{l-1}^T)$), following ITI [7]. Further, at each layer and model component, the following token positions are considered for probing: (1)**LT**: at the last token of generation (2)**PLT**: at the last token of prompt, i.e. first token of generation (3)**LLT**: at the least likely token, i.e. token location with least output probability (4)**MAX**: maxpooling activations across all token positions

Experiment Setup Experiments are conducted on Llama-7B and its instruction fine-tuned version Alpaca-7B, using two open-domain question answering (QA) tasks - Natural Questions [21] and Trivia QA [22] - and one chain-of-thought (COT) reasoning task - Strategy QA [23]. All datasets are publicly available. Natural Questions and Trivia QA consist of general knowledge questions requiring short factual answers (i.e. who/when/where type questions). StrategyQA consists of general knowledge

[2] Refer section 2 for notations used

[3] we use the macro-F1 score as a measure of performance

questions that require multi-hop reasoning to produce a binary yes/no answer. All datasets are evaluated in a closed-book setting. Responses are extracted using greedy decoding. For the QA tasks, each response is labelled as hallucinated/non-hallucinated using a rouge-1 cut-off of 0.3 against the gold reference answer, following prior work [4, 5]. For the COT task, each response is labelled as hallucinated/non-hallucinated by comparing the final yes/no answer produced against the gold reference answer, following prior work [8]. All tokens of a response share the same label. Prompt formats, dataset statistics and other implementation details are provided in appendix.

Baselines Here we focus on methods that are compute efficient at inference time, i.e. do not require multiple generations or forward passes through the model. **PE** denotes the predictive entropy of the generated tokens (using equation 2) [24]. **LP** denotes a linear probe trained on the activations of the last token at the final transformer layer. **NLP** denotes a non-linear probe trained on the activations of the last token at the final transformer layer, following SAPLMA [6].

4. Results

Detection across layers Table 1 shows the results of the three baseline methods that operate only on the final layer output alongside the layer selection and layer combination strategies. The baseline probes constructed on the last layer activations (LP, NLP) already perform significantly better than the entropy baseline (PE). Non-linear probes do not provide a major improvement over linear probes, indicating perhaps the need to increase the number of training samples to enable learning a non-linear separation. Of the three layer selection strategies, selecting the most accurate layer (MA) provides the best improvement, though minor, on most model-dataset combinations. Taking a majority vote across layers (MV) also provides minor improvements. Future research could investigate any correlation between the hallucination probabilities given by probes across layers and the early exit theory, which could point towards methods for improving both hallucination detection and early exit-based hallucination mitigation. No performance gains are seen when selecting the most confident layer (MC) at inference time.

Detection across components and tokens Table 2 compares the results of probing at different model components and token positions. Probes at the output of the MLP on average perform comparably to probes at the layer output. Probes at the attention head outputs are on average worse, although still performing better than the predictive entropy baseline. Interestingly, probes at the last token of the prompt (PLT) already perform better than the predictive entropy baseline (except for STR on Llama-7B), indicating the potential for early detection of hallucinations before generating the full response. Comparing the results of PLT, LLT and MAX on the Llama-7B model, all three types of probing show similar performance on average. However, on the Alpaca-7B model, for the TQA and NQ datasets, the LLT and MAX probes perform significantly better than the PLT probes on average. Overall, across all model-dataset combinations, the best performance is achieved by probing at the last generated token (LT).

5. Conclusion

This work compares the hallucination detection capability at various layers, transformer components and token positions during the generation process in LLMs, using probing experiments. Specifically, binary probes are constructed on top of activations at the attention head, MLP and layer output for each transformer decoder block in the LLM. Probes are constructed at the first token of generation, at the token with least output probability, as well as using maxpooled activations across all tokens. Results across two LLMs and three factual question-answering datasets show that hallucinations can be detected at varying levels at all considered components and token positions, with the best performance being achieved at the last token position using the layer output activations. Reasonable performance achieved

Table 1
Results: Detection across layers; mean and standard deviation of AUC values computed across 3 random seeds

		Llama-7B			Alpaca-7B		
	TQA	NQ	STR	TQA	NQ	STR	
PE	56.5	54.0	50.4	56.3	59.6	65.9	
LP	79.2 (0.4)	83.0 (0.1)	62.2 (1.3)	85.4 (0.4)	86.2 (0.3)	82.1 (1.8)	
NLP	78.9 (0.4)	83.1 (0.2)	63.7 (1.8)	85.3 (0.5)	86.1 (0.5)	**83.5 (0.3)**	
LP-MA	78.8 (0.3)	**84.5 (0.5)**	**66.5 (1.4)**	85.7 (0.3)	**87.3 (0.6)**	83.5 (1.6)	+1.2
LP-MC	79.7 (0.1)	81.5 (2.1)	62.9 (4.1)	85.2 (0.2)	86.5 (0.9)	81.0 (1.3)	-0.2
LP-MV	**79.9 (0.8)**	83.9 (0.2)	66.4 (0.7)	**85.9 (0.2)**	83.6 (1.4)	83.5 (0.6)	+0.7

Table 2
Results: Detection across components and tokens; AUC values averaged over all layers, mean and standard deviation computed across 3 random seeds

		Llama-7B			Alpaca-7B	
	TQA	NQ	STR	TQA	NQ	STR
LAY-LT	**77.9 (0.3)**	82.0 (0.4)	63.0 (0.1)	**84.6 (0.1)**	**85.7 (0.2)**	81.5 (0.5)
MLP-LT	77.7 (0.1)	**82.1 (0.5)**	**63.1 (0.5)**	84.4 (0.1)	85.0 (0.2)	**81.8 (0.3)**
AH-LT	71.7 (0.1)	75.9 (0.1)	55.0 (0.3)	78.3 (0.1)	78.4 (0.1)	75.8 (0.1)
LAY-PLT	76.8 (0.4)	77.9 (0.4)	50.8 (0.4)	59.9 (0.3)	65.2 (0.5)	67.4 (0.4)
LAY-LLT	74.9 (0.3)	79.0 (0.2)	51.6 (0.9)	78.3 (0.3)	75.2 (0.5)	64.2 (0.4)
LAY-MAX	75.2 (0.1)	79.8 (0.3)	53.0 (0.2)	82.3 (0.1)	84.9 (0.3)	70.2 (0.2)

at the first token indicates the potential for deploying early detection mechanisms, which can help save compute costs. Given the good detection performance achieved at multiple layers in the LLM, simple strategies are explored for selecting and creating an ensemble across layers. Results show that using an in-distribution validation set to identify the layer with the best detection performance, as well as a majority vote ensemble across all layers can provide minor performance gains at inference time. Overall, this work highlights that hallucinations can be detected at various points during the generation process and indicates that future research in developing more sophisticated detection mechanisms on top of model activations can provide further gains. For instance, leveraging activations of sampled responses alongside greedy responses for probe training could help learn a more generalisable separation between hallucinations and non-hallucinations.

References

[1] P. Manakul, A. Liusie, M. J. F. Gales, Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models, 2023. arXiv:2303.08896.

[2] N. Mündler, J. He, S. Jenko, M. Vechev, Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation, arXiv preprint arXiv:2305.15852 (2023).

[3] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, D. Zhou, Self-consistency improves chain of thought reasoning in language models, arXiv preprint arXiv:2203.11171 (2022).

[4] L. Kuhn, Y. Gal, S. Farquhar, Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation, ArXiv abs/2302.09664 (2023). URL: https://api.semanticscholar.org/CorpusID:257039062.

[5] J. Duan, H. Cheng, S. Wang, A. Zavalny, C. Wang, R. Xu, B. Kailkhura, K. Xu, Shifting attention to relevance: Towards the uncertainty estimation of large language models, 2023. arXiv:2307.01379.

[6] A. Azaria, T. Mitchell, The internal state of an LLM knows when it's lying, in: The 2023 Conference

on Empirical Methods in Natural Language Processing, 2023. URL: https://openreview.net/forum?id=y2V6YgLaW7.

[7] K. Li, O. Patel, F. Viégas, H. Pfister, M. Wattenberg, Inference-time intervention: Eliciting truthful answers from a language model, in: Thirty-seventh Conference on Neural Information Processing Systems, 2023. URL: https://openreview.net/forum?id=aLLuYpn83y.

[8] Y.-S. Chuang, Y. Xie, H. Luo, Y. Kim, J. Glass, P. He, Dola: Decoding by contrasting layers improves factuality in large language models, in: The Twelfth International Conference on Learning Representations, 2024. URL: https://openreview.net/forum?id=Th6NyL07na.

[9] C. Burns, H. Ye, D. Klein, J. Steinhardt, Discovering latent knowledge in language models without supervision, arXiv preprint arXiv:2212.03827 (2022).

[10] K. Meng, D. Bau, A. Andonian, Y. Belinkov, Locating and editing factual associations in gpt, in: S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh (Eds.), Advances in Neural Information Processing Systems, volume 35, Curran Associates, Inc., 2022, pp. 17359–17372. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/6f1d43d5a82a37e89b0665b33bf3a182-Paper-Conference.pdf.

[11] Y. A. Yadkori, I. Kuzborskij, A. György, C. Szepesvári, To believe or not to believe your llm, 2024. arXiv:2406.02543.

[12] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[13] S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz, J. Weston, Chain-of-verification reduces hallucination in large language models, arXiv preprint arXiv:2309.11495 (2023).

[14] J. Huang, X. Chen, S. Mishra, H. S. Zheng, A. W. Yu, X. Song, D. Zhou, Large language models cannot self-correct reasoning yet, in: The Twelfth International Conference on Learning Representations, 2024. URL: https://openreview.net/forum?id=IkmD3fKBPQ.

[15] W. Shi, X. Han, M. Lewis, Y. Tsvetkov, L. Zettlemoyer, S. W. tau Yih, Trusting your evidence: Hallucinate less with context-aware decoding, 2023. arXiv:2305.14739.

[16] T. Schuster, A. Fisch, J. Gupta, M. Dehghani, D. Bahri, V. Tran, Y. Tay, D. Metzler, Confident adaptive language modeling, Advances in Neural Information Processing Systems 35 (2022) 17456–17472.

[17] Q. Huang, X. wen Dong, P. Zhang, B. Wang, C. He, J. Wang, D. Lin, W. Zhang, N. H. Yu, Opera: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation, ArXiv abs/2311.17911 (2023). URL: https://api.semanticscholar.org/CorpusID:265498818.

[18] E. Hernandez, B. Z. Li, J. Andreas, Inspecting and editing knowledge representations in language models, in: Arxiv, 2023. URL: https://arxiv.org/abs/2304.00740.

[19] K. Tian, E. Mitchell, H. Yao, C. D. Manning, C. Finn, Fine-tuning language models for factuality, arXiv preprint arXiv:2311.08401 (2023).

[20] T. Yu, Y. Yao, H. Zhang, T. He, Y. Han, G. Cui, J. Hu, Z. Liu, H.-T. Zheng, M. Sun, T.-S. Chua, Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 13807–13816.

[21] K. Lee, M.-W. Chang, K. Toutanova, Latent retrieval for weakly supervised open domain question answering, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 6086–6096. URL: https://www.aclweb.org/anthology/P19-1612. doi:10.18653/v1/P19-1612.

[22] M. Joshi, E. Choi, D. Weld, L. Zettlemoyer, triviaqa: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension, arXiv e-prints (2017) arXiv:1705.03551. arXiv:1705.03551.

[23] M. Geva, D. Khashabi, E. Segal, T. Khot, D. Roth, J. Berant, Did Aristotle Use a Laptop? A Question

Answering Benchmark with Implicit Reasoning Strategies, Transactions of the Association for Computational Linguistics (TACL) (2021).

[24] A. Malinin, M. Gales, Uncertainty estimation in autoregressive structured prediction, 2021. URL: https://arxiv.org/abs/2002.07650. arXiv:2002.07650.

A. Equations describing uncertainty estimation methods

Predictive entropy of individual sample with token relevance weighting [5]:

$$PE(s^m|x) = -\sum_{i} \log p(s_i^m|s_{<i}^m, x) Rele(s^m, i) \tag{8}$$

Entropy of sampled space:

$$E(x) = -\sum_{m=1}^{M} p(s^m|x) \log p(s^m|x) \tag{9}$$

Applying Monte Carlo Integration with importance sampling:

$$E(x) = -\frac{1}{M}\sum_{m=1}^{M} \log p(s^m|x) \tag{10}$$

Entropy of sampled space with sample relevance weighting [5]:

$$E(x) = -\frac{1}{M}\sum_{m=1}^{M} \log\left(p(s^m|x) + \frac{\sum_{k,k\neq m} Sim(s^m, s^k)p(s^m|x))}{t}\right. \tag{11}$$

Semantic entropy of sampled space [4]:

$$SE(x) = -\sum_{c} p(c|x) \log p(c|x) \tag{12}$$

where $p(c|x) = \sum_{s\in c} p(s|x)$

B. Equations describing decoding strategies

ITI [7] for mitigation:

$$\text{Modified attention head activations: } x_{l+1} = x_l + \sum_{h=1}^{H} Q_l^h(Att_l^h(P_l^h x_l) + \alpha\sigma_l^h\theta_l^h) \tag{13}$$

Opera [17] for mitigation:

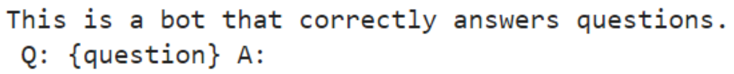
$$\text{Penalty at final layer: } q_L(x^t|x^{<t}) = \text{Softmax}(\phi(x_{l+1}^t) - penalty) \tag{14}$$

DoLa [8] for mitigation:

$$\text{Modified distribution at final layer: } q'_L(x^t|x^{<t}) = \text{Softmax}(F(q_L(x^t|x^{<t}), q_l(x^t|x^{<t}))) \tag{15}$$

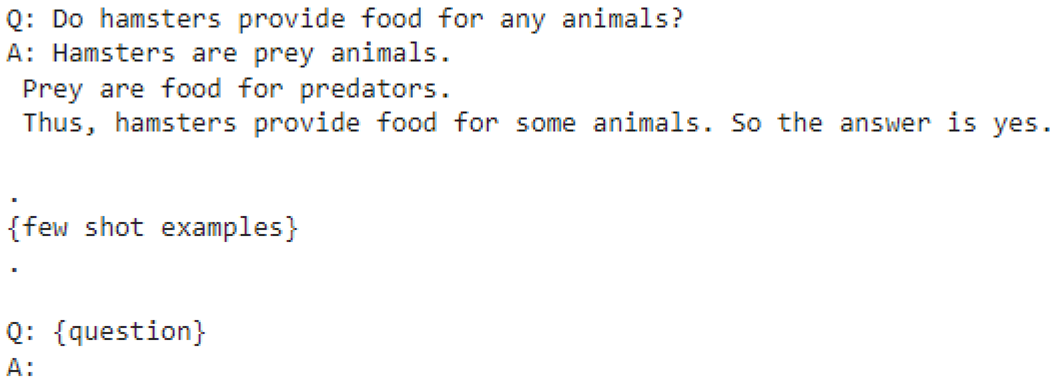
C. Experiment Setup

Prompt Formats Prompt formats used are shown in figure 3 [4] and figure 4 [8] for the QA and COT tasks, respectively.



```
This is a bot that correctly answers questions.
 Q: {question} A:
```

Figure 3: Prompt Format - Natural Questions, Trivia QA



```
Q: Do hamsters provide food for any animals?
A: Hamsters are prey animals.
 Prey are food for predators.
 Thus, hamsters provide food for some animals. So the answer is yes.


.
{few shot examples}
.

Q: {question}
A:
```

Figure 4: Prompt Format - StrategyQA

Table 3
Dataset statistics

	TQA		NQ		STRQA	
	Llama7B	Alpaca7B	Llama7B	Alpaca7B	Llama7B	Alpaca7B
# train prompts	2000	2000	2000	2000	1832	1832
# test prompts	1800	1800	1800	1800	458	458
train accuracy	52.7	29.4	24.7	9.2	60.2	42.5
test accuracy	57.3	34.2	22.6	10.1	60.0	43.4

Dataset Statistics Table 3 shows the number of train and test samples used per dataset. Accuracy indicates the percentage of questions answered correctly by the model (i.e. non-hallucinations).

Implementation Details All probes are trained with a batch size of 128, using AdamW optimiser with linear warm-up for 5 epochs and cosine annealing for a maximum of 50 epochs. For each dataset and method, learning rate is selected from a coarse grid search $\in [0.5, 0.05, 0.005, 0.0005, 0.00005]$ using a held-out validation set. For each probing method, mean and standard deviation of AUC values are reported, averaged over 3 random seeds.