

Detecting Hallucination in Large Language Models through Deep Internal Representation Analysis

Anonymous submission #1316

Abstract

1 Large language models (LLMs) have demonstrated
2 exceptional performance across various domains.
3 However, LLMs are prone to hallucinate facts and
4 generate non-factual responses, which can under-
5 mine their reliability in real-world applications.
6 Current hallucination detection methods suffer from
7 external resource demands, substantial time over-
8 head, difficulty overcoming LLMs’ intrinsic limita-
9 tion, and insufficient modeling. In this paper, we
10 propose MHAD, a novel internal-representation-
11 based hallucination detection method. MHAD uti-
12 lizes linear probing to select neurons and layers
13 within LLMs. The selected neurons and layers are
14 demonstrated with significant awareness of hallu-
15 cinations at the initial and final generation steps.
16 By concatenating the outputs from these selected
17 neurons of selected layers at the initial and final
18 generation steps, a hallucination awareness vector
19 is formed, enabling precise hallucination detection
20 via a multi-layer perceptron (MLP). Additionally,
21 we introduce SOQHD, a novel benchmark for eval-
22 uating hallucination detection in Open-Domain QA
23 (ODQA). Extensive experiments show that MHAD
24 outperforms existing hallucination detection meth-
25 ods across multiple LLMs, demonstrating superior
26 effectiveness.

27 1 Introduction

28 Although large language models (LLMs) have demonstrated
29 remarkable performance across diverse fields [Wu *et al.*,
30 2023; Thirunavukarasu *et al.*, 2023; Hedderich *et al.*, 2024;
31 Wang *et al.*, 2024], they are known to have a risk of gener-
32 ating hallucinations [Bang *et al.*, 2023; Shen *et al.*, 2023].
33 Hallucinations—instances where LLMs generate responses
34 that appear plausible but are factually incorrect—hinder the
35 adoption of LLMs in real-world applications that require high
36 reliability and factual correctness [Ji *et al.*, 2023; Huang *et*
37 *al.*, 2023]. Detecting hallucinations helps reliably assess the
38 truthfulness of LLM-generated responses.

39 Current research on hallucination detection for LLMs can
40 be broadly classified into four categories: retrieval-based

41 methods, sampling-based methods, uncertainty-based meth- 41
42 ods, and internal-representation-based methods. **Retrieval-** 42
43 **based methods** [Li *et al.*, 2023a; Min *et al.*, 2023] evaluate 43
44 the veracity of the response generated by LLMs against exter- 44
45 nal knowledge sources. However, these methods rely heavily 45
46 on external knowledge sources, which may not always be ac- 46
47 cessible. **Sampling-based methods** [Manakul *et al.*, 2023; 47
48 Mündler *et al.*, 2024] assess information consistency among 48
49 multiple sampled responses from the same LLM. How- 49
50 ever, these methods are impractical for real-time scenarios 50
51 due to the excessive time overhead of multiple sampling. 51
52 **Uncertainty-based methods** [Zhang *et al.*, 2023; Manakul 52
53 *et al.*, 2023] evaluate the factual accuracy of LLM-generated 53
54 responses by calculating the probability or entropy of tokens 54
55 within them. Although these methods eliminate the need 55
56 for additional resources like external knowledge sources or 56
57 sampled responses from LLMs, they present significant chal- 57
58 lenges in addressing the intrinsic limitation of LLMs: LLMs 58
59 can generate hallucinations with high confidence [Azaria and 59
60 Mitchell, 2023; Schulman, 2023]. **Internal-representation-** 60
61 **based methods** [Azaria and Mitchell, 2023; Su *et al.*, 2024; 61
62 Chen *et al.*, 2024; Du *et al.*, 2024] detect hallucinations 62
63 based on the internal representation of LLMs. Azaria and 63
64 Mitchell [2023] show that the internal representation demon- 64
65 strates greater reliability than uncertainty. However, these 65
66 methods still suffer from insufficient modeling as they neglect 66
67 the complementary information across layers and generation 67
68 steps of LLMs. 68

69 To address the above issues, we propose **MHAD** (Model 69
70 Hallucination Awareness for Hallucination Detection), a 70
71 novel internal-representation-based method to detect halluci- 71
72 nations in LLMs. We assume that the internal representations 72
73 of LLMs encompass their awareness of whether the responses 73
74 they generate are hallucinated or factual. Our basic idea is to 74
75 model the hallucination awareness in LLMs based on their 75
76 internal representations across layers during the generation 76
77 process for detecting hallucinations. 77

78 After LLMs process the query but before generating a re- 78
79 sponse, they only encode the query without encoding any hal- 79
80 lucination. When LLMs generate the termination token, the 80
81 factuality of the responses is determined, as the termination 81
82 token itself is hallucination-free, and the generation process 82
83 ceases once the termination token is generated. We focus on 83
84 the internal representation at the initial and final generation 84

steps. Moreover, some works suggest that different layers of transformer-based language models capture various aspects of the input, from basic lexical and grammatical features in lower layers to more abstract concepts in higher layers [Jawahar *et al.*, 2019; Sajjad *et al.*, 2022; Wang *et al.*, 2022; Voita *et al.*, 2024]. Therefore, we harness the complementary information across layers of LLMs to enhance the modeling of hallucination awareness. Specifically, we leverage linear probing [Alain and Bengio, 2017; Burns *et al.*, 2023; Park *et al.*, 2024] to select neurons and layers within LLMs that demonstrate significant awareness of hallucinations at the initial and final generation steps. By concatenating the outputs from the selected neurons of selected layers at the initial and final generation steps, a hallucination awareness vector is formed. The vector is then used to detect hallucinations via a multi-layer perceptron (MLP). MHAD eliminates the need for external knowledge sources or multiple sampled responses and demonstrates superior performance.

To evaluate MHAD thoroughly, we develop **SOQHD** (Sustainable Open-Domain QA Hallucination Detection), a novel benchmark for hallucination detection in ODQA. ODQA is a challenging knowledge-intensive task and relevant to practical use cases [Guu *et al.*, 2020; Lewis *et al.*, 2020; Friel and Sanyal, 2023]. We hence focuses on detecting hallucinations in ODQA. Previous benchmarks [Li *et al.*, 2023a; Manakul *et al.*, 2023; Azaria and Mitchell, 2023; Friel and Sanyal, 2023] have primarily focused on specific data types, such as responses generated by LLMs along with hallucination labels, thereby limiting their applicability to evaluate internal-representation-based methods. Moreover, these benchmarks do not consider temporal consistency during construction, which may result in outdated labels. For example, the label for question like—To the nearest million, what is the population of Australia?—need to be updated to reflect the latest population. SOQHD provides not only the LLM-generated responses, along with hallucination labels, but also the internal representations across layers during the generation process of multiple LLMs, such as LLaMA3-Instruction-8B [Meta, 2024]. Additionally, SOQHD exclude questions with answers that vary over time to ensure temporal consistency. Our contributions are threefold:

- We propose MHAD, a novel hallucination detection method, which utilizes the internal representations across layers during the generation process of LLMs to detect hallucinations.
- We develop SOQHD, a novel hallucination detection benchmark for ODQA, which provides the LLM-generated responses along with hallucination labels and the internal representations of LLMs, while ensuring temporal consistency.
- We conduct extensive experiments on multiple datasets, demonstrating that MHAD outperforms existing hallucination detection methods across multiple LLMs in terms of effectiveness.

2 MHAD

In this section, we introduce **MHAD** (Model Hallucination Awareness for Hallucination Detection), a novel internal-

representation-based hallucination detection method. Three types of internal representations are primarily utilized: attention output, feed-forward network output, and layer output. MHAD consists of several key steps: internal representation collection, linear probing, neuron selection, layer selection, and hallucination awareness vector construction. The overview of MHAD is provided in Figure 1.

2.1 Internal Representation Collection

We feed the question into LLMs and gather the internal representations from each layer at the initial and final generation steps.

To comprehend the internal representations, the mechanism of Transformer is formalized. The essential computations involve query, key, and value vectors derived from the hidden state, concatenation of attention heads output, and feed-forward network transformations. Specifically, the attention output ($\mathbb{A}\mathbb{O}$), feed-forward network output ($\mathbb{F}\mathbb{O}$), and layer output ($\mathbb{L}\mathbb{O}$) are extracted as follows:

$$Q_h = XW_h^Q, \quad (1)$$

$$K_h = XW_h^K, \quad (2)$$

$$V_h = XW_h^V, \quad (3)$$

$$\mathbb{A}\mathbb{O} = \text{concat}_h(\text{softmax}(Q_h K_h^T) V_h), \quad (4)$$

$$H = X + \mathbb{A}\mathbb{O}, \quad (5)$$

$$\mathbb{F}\mathbb{O} = f_{\text{act}}(HW_1 + b_1)W_2 + b_2, \quad (6)$$

$$\mathbb{L}\mathbb{O} = H + \mathbb{F}\mathbb{O}, \quad (7)$$

where X represents the hidden state of Transformer, W_h^Q , W_h^K , W_h^V are the linear transformation matrices for computing the query vector Q_h , key vector K_h , and value vector V_h of the h -th attention head, concat denotes the concatenation operation, W_1, b_1, W_2, b_2 are the weight parameters of the feed-forward network (FFN), and f_{act} denotes the activation function in the FFN.

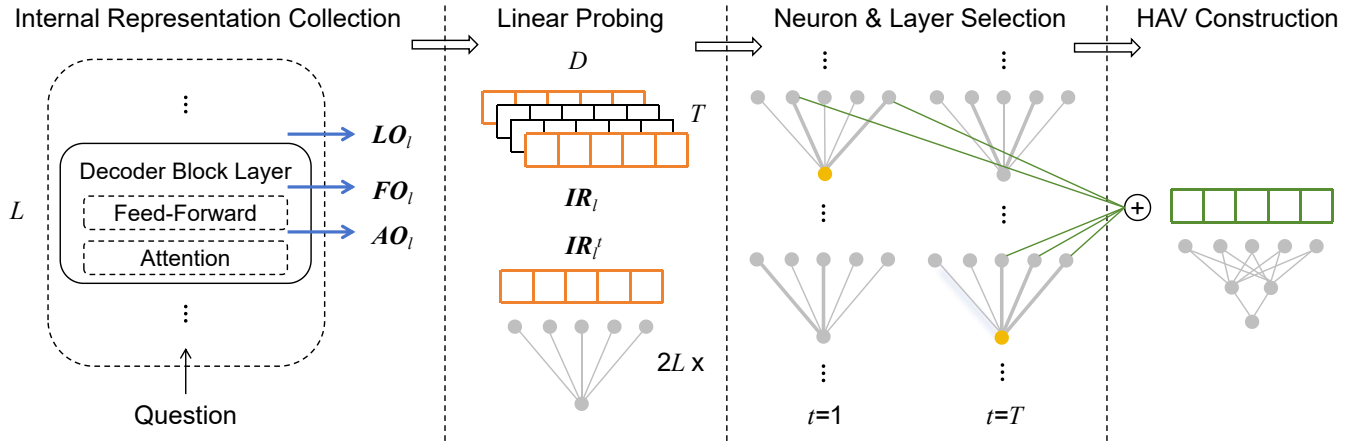
2.2 Linear Probing

Linear probing [Alain and Bengio, 2017] initially trains an auxiliary classifier based on the internal activation layer outputs of neural networks to detect certain attributes of the input. We use linear probing to model the hallucination awareness within the internal representations of LLMs.

Specifically, we construct a linear probe classifier for each layer at the initial and final generation steps. Each linear probe classifier is a two-layer FFN, with its input corresponding to the dimension of the internal representation. The output of the linear probe classifier is a binary label, indicating whether the LLMs generate hallucination. This step offers guidance for subsequent neuron and layer selection steps. The training process of the linear probe classifier is formulated as:

$$\hat{W}_l^t, \hat{b}_l^t \leftarrow \arg \min_{W_l^t, b_l^t} \text{BCE}(Y, \sigma(I_l^t W_l^t + b_l^t)), \quad (8)$$

where I_l^t represents the internal representations of the l -th layer at the t -th generation step, σ is the sigmoid activation function, mapping the output of the linear probe classifier to the $(0, 1)$ range, BCE is the binary cross-entropy loss function, W_l^t and b_l^t are the trainable weight parameters of the linear probe classifier, and Y is the ground-truth label.



L : layer number of LLMs LO : layer output FO : feed-forward network output AO : attention output IR : any internal representation
 D : dimension of IR T : total number of generation steps l : any layer t : any generation step HAV : hallucination awareness vector

Figure 1: Overview of MHAD. We focus on the internal representations at the initial and final generation steps (orange table). Neurons (gray bold line) and layers (yellow node) within LLMs that demonstrate significant awareness of hallucinations are selected by linear probing. The outputs from these selected neurons of selected layers at the initial and final generation steps are concatenated to form the hallucination awareness vector (green table), which enables precise hallucination detection via a multi-layer perceptron (MLP).

2.3 Neuron Selection

194 Han et al. [2016] assessed the importance of neural network
 195 weight parameters based on their absolute values, setting in-
 196 significant weight parameters to 0 for weight pruning. We se-
 197 lect neurons that demonstrate significant awareness of hallu-
 198 cinations based on the absolute value of the linear prob clas-
 199 sifier’s weight parameters. Neurons with larger absolute
 200 values are considered more informative for hallucination de-
 201 tection.

202 Specifically, we first sort the linear prob classifier’s weight
 203 parameters in descending order based on their absolute val-
 204 ues. We then go through these weight parameters from largest
 205 to smallest. When the ratio of the cumulative sum of squared
 206 weight values to the total sum of squared weight values ex-
 207 ceeds a predefined threshold, we select the neurons corre-
 208 sponding to the traversed weight parameters. Squaring the
 209 weight values is intended to reduce the impact of weight pa-
 210 rameters with small absolute values. Since the internal rep-
 211 resentations are high-dimensional vectors, this step reduces
 212 the introduction of noise irrelevant to hallucination. The process
 213 of neuron selection is formulated as:

$$A = \text{argsort}(\text{abs}(\hat{W}_l^t)), \quad (9)$$

$$\sum_{k=1}^{i-1} (\hat{W}_{l,A_k}^t)^2 < \alpha \cdot \|\hat{W}_l^t\|_2^2 \leq \sum_{k=1}^i (\hat{W}_{l,A_k}^t)^2, \quad (10)$$

$$1 \leq i \leq |A|, \quad 0 < \alpha < 1,$$

$$\hat{N}_l^t = \{A_k \mid k = 1, 2, \dots, i\}, \quad (11)$$

217 where A is the index set of the weight parameters sorted in de-
 218 scending order of their absolute values, A_k denotes the index
 219 of the weight parameter with the k -th largest absolute value,
 220 \hat{W}_{l,A_k}^t represents the value of the weight parameter, which
 221 ranks as the k -th largest in absolute value, α is the predefined
 222 ratio threshold, and \hat{N}_l^t represents the index set of the neurons

223 selected in the l -th layer at the t -th generation step of LLMs.
 224 Note that the smaller the α , the less likely it is to select neu-
 225 rons corresponding to weight parameters with small absolute
 226 values.

2.4 Layer Selection

227 We select layers that demonstrate significant awareness of
 228 hallucinations based on the performance of the linear prob clas-
 229 sifier. The better the performance of the linear probe clas-
 230 sifier, the more significant the hallucination awareness within
 231 the internal representations of the corresponding layer.

232 Specifically, we employ two heuristic rules: (i) using the
 233 top- k method to select the top-performing layers, and (ii) set-
 234 ting a threshold to select layers with AUROC above the pre-
 235 defined threshold on the validation set. Given the variance
 236 in hallucination awareness across different layers, this step
 237 helps mitigate the interference from the layers demonstrat-
 238 ing weak awareness of hallucinations.
 239

2.5 Hallucination Awareness Vector Construction

240 The last step first constructs a hallucination awareness vector
 241 through concatenating the outputs from the selected neurons
 242 of selected layers at the initial and final generation steps. The
 243 hallucination awareness vector encapsulates the critical infor-
 244 mation necessary for hallucination detection. The process is
 245 formulated as:
 246

$$\hat{I} = \text{concat}_{t,\hat{l}^t}(I_{\hat{l}^t,\hat{N}_{\hat{l}^t}^t}^t), \quad (12)$$

247 where t denotes the t -th generation step, \hat{l}^t represents the in-
 248 dex of a selected layer at the t -th generation step, $I_{\hat{l}^t,\hat{N}_{\hat{l}^t}^t}^t$ rep-
 249 represents the outputs from the selected neurons of the selected
 250 layer at the t -th generation step, and \hat{I} denotes the **hallucina-
 251 tion awareness vector**. An MLP is then trained based on the
 252 hallucination awareness vector to detect hallucinations.

3 The SOQHD Benchmark

In this section, we introduce **SOQHD** (Sustainable Open-Domain QA Hallucination Detection), a novel hallucination detection benchmark for ODQA. The construction process includes three steps: filtering, sampling, and reasoning.

3.1 Filtering

The filtering step aims to exclude questions with answers that vary over time to ensure temporal consistency. For example, the labels for questions like “To the nearest million, what is the population of Australia?” have become outdated as they fail to reflect current data.

This step begins with the manual annotation of a small sample of questions from the development sets of TriviaQA [Joshi *et al.*, 2017] and NQ [Kwiatkowski *et al.*, 2019], which are widely used ODQA benchmarks. gpt-3.5-turbo is also used to annotate these samples. Statistics indicate that the annotation results of gpt-3.5-turbo achieve a consistency rate of up to 96% with human annotations. Therefore, we annotate the remaining questions using gpt-3.5-turbo. Inspired by self-consistency [Wang *et al.*, 2023], we have gpt-3.5-turbo annotate each question five times and obtain the final annotation result via majority voting, ensuring high accuracy. Questions in the development sets of TriviaQA and NQ with time-varying answers are then filtered out.

3.2 Sampling

The sampling step aims to construct the question set of SOQHD while maintaining the original datasets’ response length distribution.

To ensure that the final token generated by LLMs is the termination token, questions that LLaMA2-Chat-13B [Touvron *et al.*, 2023] cannot answer within a maximum generation length of 300 are excluded. The remaining questions are stratified into three levels based on the length of response generated by LLaMA2-Chat-13B. Stratified sampling is then used to form the question set of SOQHD. The training set of SOQHD contains a total of 2000 questions, and the test set comprises 500 questions.

3.3 Reasoning

The reasoning step aims to obtain the responses generated by LLMs, the hallucination labels, and the internal representations across layers during the generation process of LLMs.

We select five widely used open-source fine-tuned LLMs, including LLaMA3-Instruction-8B [Meta, 2024], LLaMA2-Chat-13B [Touvron *et al.*, 2023], LLaMA2-Chat-7B [Touvron *et al.*, 2023], Vicuna-7B [Chiang *et al.*, 2023], and Alpaca-7B [Taori *et al.*, 2023], for reasoning. Compared to base LLMs, fine-tuned LLMs are better equipped to generate concise and user-aligned responses necessary for real-world applications. To tackle the challenge of hallucination in real-world applications, we select fine-tuned LLMs as our focus.

In this step, questions from SOQHD are inputted into multiple LLMs to generate responses. Normalization operations, including removing punctuation and converting to lowercase, are applied before inputting the questions. The greedy decoding strategy is employed. Since LLMs tend to

Model	H=0 Rate	Complete Rate
Alpaca-7B	0.476	1.000
Vicuna-7B	0.464	0.998
LLaMA2-Chat-7B	0.545	0.999
LLaMA2-Chat-13B	0.573	1.000
LLaMA3-Instruction-8B	0.634	1.000

Table 1: Hallucination-free rate and complete rate of LLMs on the training set of SOQHD, where H=0 Rate indicates the hallucination-free rate. We consider a response complete if its final token is the termination token.

be wordy, which makes the Exact Match (EM) score, the traditional evaluation metric for ODQA tasks [Chen *et al.*, 2017; Izacard and Grave, 2021], not applicable. Thus, we consider the response to have **hallucination** if the ground-truth answer does not appear in the response; otherwise, it is considered to not have hallucination. The generated responses, hallucination labels, and internal representations across layers during the generation process are stored to form the final SOQHD.

3.4 Analysis

The hallucination-free rate and complete rate of five LLMs on the training set of SOQHD are shown in Table 1.

Findings indicate that larger LLMs generally perform better, with LLaMA3-Instruction-8B outperforming others despite not being the largest. Moreover, LLMs can typically answer the questions from SOQHD within the maximum generation length.

4 Experiments

4.1 Experiment setting

Dataset and Metrics. We evaluate MHAD and other baselines on our proposed SOQHD dataset. Consistent with previous studies [Chen *et al.*, 2024; Du *et al.*, 2024], we use AUROC as the evaluation metric. AUROC is a popular metric to evaluate the quality of a binary classifier. We also evaluate on existing datasets: HaluEval [Li *et al.*, 2023a].

Baselines. We choose the following nine competitive hallucination detection methods as baselines. (i) Probability Assessment [Manakul *et al.*, 2023]: This method detects hallucinations based on the probabilities of generated tokens by LLMs. It uses average and max pooling to aggregate the negative log probabilities of generated tokens, denoted as $\text{Avg}(-\log p)$ and $\text{Max}(-\log p)$, respectively. (ii) Entropy Assessment [Manakul *et al.*, 2023]: This method detects hallucinations based on the entropy of generated tokens by LLMs. It uses average and max pooling to aggregate the entropy of generated tokens, denoted as $\text{Avg}(\mathcal{H})$ and $\text{Max}(\mathcal{H})$, respectively. (iii) SelfCheckGPT [Manakul *et al.*, 2023]: This method assesses the consistency among multiple sampled responses from LLMs. Four methods are employed to assess the consistency, denoted as SCG-BS, SCG-QA, SCG-NG, and SCG-NLI. (iv) EUBHD [Zhang *et al.*, 2023]: This method detects hallucinations based on the uncertainty of generated keywords by LLMs. (v) SAPLMA [Azaria and Mitchell, 2023]: This method detects the truthfulness of a

Baselines	LLC-13B	LLC-7B	LLI3-8B	Vicuna-7B	Alpaca-7B
Avg($-\log p$)	0.6336	0.5933	0.6498	0.6153	0.7009
Avg(\mathcal{H})	0.6122	0.5706	0.6842	0.4639	0.7313
Max($-\log p$)	0.5546	0.5211	0.6409	0.4627	0.7040
Max(\mathcal{H})	0.5496	0.5097	0.6793	0.5079	0.7215
SCG-BS	0.5552	0.5775	0.6195	0.5958	0.6792
SCG-QA	0.5431	0.5620	0.5888	0.5889	0.6969
SCG-NG	0.5364	0.5525	0.6018	0.6409	0.7109
SCG-NLI	0.5538	0.6073	0.7060	0.7019	0.7548
EUBHD	0.5728	0.5798	0.6431	0.6242	0.5764
SAPLMA	0.4384	0.4773	0.4052	0.4663	0.4310
MIND	0.5099	0.5138	0.5424	0.5065	0.5190
INSIDE	0.5398	0.5752	0.5972	0.6895	0.6532
HaloScope	0.6517	0.5959	0.5071	0.5581	0.5866
GPT4-HR	0.7092	0.6705	0.6684	0.7069	0.7942
MHAD-$\mathbb{A}\mathbb{O}$	0.7768	0.7336	0.7843	0.7771	0.7875
MHAD-$\mathbb{F}\mathbb{O}$	0.7642	0.7337	0.7665	0.7566	0.7869
MHAD-$\mathbb{L}\mathbb{O}$	0.7728	0.7204	0.7539	0.7646	0.7961

Table 2: Comparison of MHAD with baseline methods in terms of AUROC on the test set of SOQHD. SCG stands for ‘‘SelfCheckGPT’’, LLC stands for ‘‘LLaMA2-Chat’’, and LLI3 stands for ‘‘LLaMA3-Instruction’’. The best results are in bold. MHAD outperforms other baselines significantly with p -value <0.05 .

statement based on the internal representation of LLMs. (vi) MIND [Su *et al.*, 2024]: This is a training framework that leverages the internal representations of LLMs for hallucination detection. (vii) INSIDE [Chen *et al.*, 2024]: This method explores the semantic information preserved within internal representations for hallucination detection. (viii) HaloScope [Du *et al.*, 2024]: This is a learning framework for hallucination detection, which exploits the LLM generations arising in the wild. (ix) GPT4-HR [Li *et al.*, 2023a]: This method prompts a LLM to recognize whether the responses generated by other LLMs have hallucination.

Implementation Details. The MHAD classifier employs a 4-layer MLP for hallucination detection, with its input corresponding to the dimension of the hallucination awareness vector. The hidden layers have dimensions of 1024 and 128, respectively. The ReLU activation function is used between layers, with a dropout rate of 0.5. The classifier is optimized using Adam with a learning rate of $1e-5$, a weight decay of $1e-2$, and a training batch size of 64. For the hyperparameters α and top-k used for neuron and layer selection, the settings are determined using the separate validation set, which is a random sampling 20% subset from the SOQHD training set. Baselines are implemented using official code and data while following the settings outlined in the respective papers. All experiments are conducted on a single RTX A6000.

4.2 Main Results

Table 2 presents the performance comparison of MHAD against baseline methods. Our key findings are as follows:

(i) **MHAD outperforms all baseline methods across all LLMs.** MHAD, leveraging the internal representations across layers during the generation process of LLMs, shows superior performance across all LLMs, highlighting its effectiveness in detecting hallucinations. This supports the hypothesis that LLMs’ internal representations encompass the

awareness of whether their responses are hallucinated or factual. It also demonstrates the potential of using complementary information within the internal representations of LLMs to detect hallucinations. Notably, MHAD surpasses Probability/Entropy Assessment baseline methods. We believe this suggests that LLMs are aware that they generate hallucinations with high confidence, making internal-representation-based methods demonstrate greater reliability than uncertainty-based methods, as aligned with the discoveries made by Azaria and Mitchell [2023]. Moreover, MHAD does not rely on external knowledge sources or multiple sampled responses, making it suitable for real-world applications.

(ii) **In most LLMs, the hallucination awareness in the attention output is comparable to, or stronger than, that in the other two types of internal representations.** As indicated in Table 2, the MHAD- $\mathbb{A}\mathbb{O}$ demonstrates superior performance on LLaMA2-Chat-13B, LLaMA3-Instruction-8B, and Vicuna-7B compared to both MHAD- $\mathbb{F}\mathbb{O}$ and MHAD- $\mathbb{L}\mathbb{O}$. When applied to LLaMA2-Chat-7B, MHAD- $\mathbb{A}\mathbb{O}$ performs only slightly below MHAD- $\mathbb{F}\mathbb{O}$. On Alpaca-7B, MHAD- $\mathbb{A}\mathbb{O}$ ranks second in performance among MHAD- $\mathbb{A}\mathbb{O}$, MHAD- $\mathbb{F}\mathbb{O}$, and MHAD- $\mathbb{L}\mathbb{O}$.

(iii) **Other findings.** (1) LLaMA3-Instruction-8B shows the lowest propensity for hallucinations among five LLMs, as indicated in the Table 1. However, the responses generated by it are the most challenging for GPT-4 to recognize whether they are hallucinated or factual, as shown in Table 2. We believe this finding as different LLMs have different hallucination patterns. LLaMA3-Instruction-8B is the least prone to hallucinations, making its hallucination patterns the most intricate and its responses the most challenging for GPT-4 to correctly recognize. Nevertheless, our method still demonstrates outstanding performance. (2) SAPLMA, though effective on its own training dataset, exhibits inferior performance when applied to detect genuine hallucinations. We be-

Method	LLC(13)	LLC(7)	LLI3	Vicuna	Alpaca
SIR	0.7547	0.7071	0.7354	0.7475	0.7740
+SN	0.7672	0.7214	0.7685	0.7450	0.7756
+SL	0.7697	0.7182	0.7815	0.7649	0.7988
+CGS	0.7768	0.7336	0.7843	0.7771	0.7875

Table 3: Ablation results for attention output.

lieve this as its training data is not generated by LLM itself. Although MIND and EUBHD are effective in wikipedia generation task, they show suboptimal performance in ODQA task. This could be attributed to the discrepancy between two tasks. We would like to note that ODQA is a challenging knowledge-intensive task and relevant to practical use cases, ensuring SOQHD can effectively assess the performance of hallucination detection methods in real-world applications.

4.3 Ablation Study

Table 3 presents the results of our ablation study for attention output. We denote the baseline, which detects hallucinations using the single-layer internal representation at the final generation step, as SIR. We then introduce the tricks of ‘‘Select Neurons’’ (SN), ‘‘Select Layers’’ (SL), and ‘‘Concatenate Generation Steps’’ (CGS) incrementally to evaluate their impact. **Select Neurons.** By selecting neurons, the outputs from the selected neurons of each layer at the final generation step are concatenated to detect hallucinations. The results show that the SN trick improves the AUROC for most LLMs, indicating the effectiveness of neuron selection. By harnessing the complementary information across layers of LLMs, more precise hallucination detection is achieved. However, the Vicuna-7B shows a decrease in performance with the SN trick, suggesting that its layers may provide a more uniform hallucination awareness.

Select Layers. When SL trick is incorporated on the basis of SN, the outputs from the selected neurons of the selected layers at the final generation step are concatenated to detect hallucinations. The SL trick further enhances the performance by mitigating the interference from the layers that demonstrate weak awareness of hallucinations. However, the LLaMA2-Chat-7B demonstrates a decrease in performance with the SL trick. We believe this as heuristic rules may fall into local optima, but we need to note that the complexity of exhaustive search is $O(2^L)$, where L is the number of layers in LLMs, making it impractical.

Concatenate Generation Steps. Through further leveraging the CGS trick, the outputs from the selected neurons of selected layers at the initial and final generation steps are concatenated to detect hallucinations. The CGS trick generally yields the highest AUROC, suggesting that the internal representation at the initial generation step can provide complementary information. We believe this is similar to the rethinking process humans engage in during problem-solving. However, Alpaca-7B exhibits a decline in performance with the CGS trick, possibly attributed to the significant gap in performance among the top-performing layers at the initial and final generation steps, as shown in Figure 2.

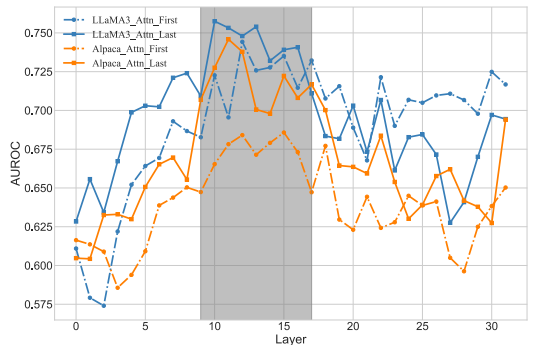


Figure 2: Comparison of performance on the validation set of SOQHD across different layers and generation steps.

Model	H=1 Rate
Alpaca-7B	0.732 (↑ 0.220)
Vicuna-7B	0.724 (↑ 0.172)
LLaMA2-Chat-7B	0.660 (↑ 0.202)
LLaMA2-Chat-13B	0.590 (↑ 0.176)
LLaMA3-Instruction-8B	0.648 (↑ 0.280)

Table 4: Hallucination rate on the test set of SOQHD when LLMs are presented with misleading information. H=1 Rate indicates the hallucination rate, and the red arrow shows the increase in hallucination rate compared to when no misleading information is provided.

Baselines	LLC(13)	LLC(7)	LLI3	Vicuna	Alpaca
Avg($-\log p$)	0.4731	0.4175	0.4256	0.4740	0.4591
Avg(\mathcal{H})	0.4675	0.4110	0.4093	0.4732	0.4612
Max($-\log p$)	0.4538	0.3966	0.3915	0.4297	0.4227
Max(\mathcal{H})	0.4381	0.4017	0.3736	0.4384	0.4439
MHAD-AO	0.6552	0.5369	0.4561	0.5701	0.5375
MHAD-FO	0.6169	0.5040	0.3948	0.6008	0.5120
MHAD-LO	0.6448	0.5276	0.4670	0.5690	0.5377

Table 5: Robustness study results.

4.4 Robustness Study Against Misleading Information

Retrieval Augmented Generation (RAG) enables LLMs to assess external knowledge sources, but the quality of these sources significantly affects the performance of LLMs. Misleading information can increase the likelihood of LLMs generating hallucinations [Pan *et al.*, 2023; Xu *et al.*, 2024]. We examine the robustness of MHAD and baseline methods when LLMs are presented with misleading information.

Specifically, we first have gpt-3.5-turbo generate misleading information for each question in the test set of SOQHD. Then, we input both the misleading information and the question into LLMs to obtain their responses.

Table 5 presents the robustness study results. The likelihood of LLMs generating hallucinations increases with misleading information, as shown in Table 4. However, MHAD maintains remarkable performance, demonstrating robustness against misleading information. Interestingly, LLaMA3-Instruction-8B is more susceptible to misleading information,

Baselines	LLC(13)	LLC(7)	LLI3	Vicuna	Alpaca
HaluEval-HR	–	49.60	–	60.34	6.68
MHAD- Δ O	64.25	55.04	59.63	53.69	67.52
MHAD-FO	68.23	50.43	59.10	50.79	73.25
MHAD-LO	60.18	51.85	52.10	68.53	71.85

Table 6: Performance on HaluEval-QA dataset.

likely due to its strong instruction-following ability. In general, larger LLMs are less affected by misleading information.

4.5 Other Results

We also evaluate our proposed MHAD on the existing HaluEval dataset [Li *et al.*, 2023a]. The HaluEval dataset includes 30,000 samples from HotpotQA [Yang *et al.*, 2018], OpenDialKG [Moon *et al.*, 2019] and CNN/Daily Mail [See *et al.*, 2017]. ChatGPT is used to generate hallucinated responses.

To maintain task format consistency, we focus on the 10,000 HotpotQA samples from the HaluEval dataset. Following the setting of [Li *et al.*, 2023a] and the proxy model strategy proposed by [Manakul *et al.*, 2023], we concatenate a question with the answer randomly selected from normal and hallucinated answers, and then input them into LLMs. The internal representations during processing the answer are stored for utilization by MHAD. The evaluation metric is Accuracy (ACC), aligned with [Li *et al.*, 2023a]. Although the proxy model strategy can adapt the HaluEval-QA dataset to our proposed method, it may not fully demonstrate the effectiveness of MHAD due to the exposure bias [Bengio *et al.*, 2015; Ranzato *et al.*, 2016] and the discrepancy between the synthetic hallucinations and the genuine hallucinations generated by LLM itself [Manakul *et al.*, 2023; Zhang *et al.*, 2024].

Table 6 presents the results of evaluating the hallucination detection classifier, which is trained on the training set of SOQHD, using 10,000 unseen samples from HaluEval-QA. Baseline results are referenced from [Li *et al.*, 2023a]. The baseline method prompts LLMs to assess whether the answer randomly selected from normal and hallucinated answers is factual. Although the flaws of the proxy model strategy limit the performance of the classifier, MHAD still outperforms the baseline across all LLMs. This further highlights MHAD’s effectiveness and generalization, suggesting that real-generation processes better reflect its superiority.

5 Related Work

5.1 Hallucination Detection

Hallucination detection in LLMs has garnered significant attention due to the increasing reliance on LLMs in various applications. Existing hallucination detection methods for LLMs can be broadly classified into four categories.

Retrieval-based. Li *et al.* [2023a] proposed an approach that prompts an LLM to evaluate whether the responses generated by other LLMs contradict objective facts, where the LLM is employed as an external knowledge source.

Sampling-based. Manakul *et al.* [2023] proposed SelfCheckGPT, a method to assess information consistency among multiple sampled responses from the same LLM. The

motivate idea of SelfCheckGPT is that when LLMs are uncertain about a given concept, the sampled responses are likely to be different and contain inconsistent facts.

Uncertainty-based. Manakul *et al.* [2023] proposed methods to detect hallucinations based on the probability or entropy of tokens in a given response. Factual responses are likely to contain tokens with higher probability and lower entropy. Inspired by human focus in factuality checking, Zhang *et al.* [2023] enhanced uncertainty-based hallucination detection with stronger focus.

Internal-representation-based. Azaria and Mitchell [2023] trained an MLP based on the single-layer internal representation of LLMs to predict the truthfulness of a sentence. Su *et al.* [2024] proposed a training framework that leverages the internal representation of LLMs for hallucination detection. Chen *et al.* [2024] explored the dense semantic information retained within LLMs’ internal representation for hallucination detection. Du *et al.* [2024] estimated the membership for samples based on an embedding factorization and trained a binary truthfulness classifier on top. However, these are limited by their single-layer focus and do not harness complementary information across layers and generation steps.

A more extensive discussion is in the Appendix. Note that the problem we study differs from the research on hallucination mitigation [Li *et al.*, 2023b; Chuang *et al.*, 2024], which aims to enhance the truthfulness of LLMs’ decoding process.

5.2 Hallucination Detection Benchmarks

Hallucination detection benchmarks are utilized to assess the effectiveness of hallucination detection methods. For instance, Manakul *et al.* [2023] developed a hallucination detection dataset by generating synthetic wikipedia articles with GPT-3, followed by manual annotation. Li *et al.* [2023a] constructed a challenging dataset of generated and human-annotated hallucinated samples to evaluate the capability of LLMs to recognize hallucination. Azaria and Mitchell [2023] introduced the True-False dataset of true and false statements. Su *et al.* [2024] constructed HELM based on the wikipedia articles generation task. Although Chen *et al.* [2024] and Du *et al.* [2024] proposed internal-representation-based methods, they neither open-sourced their utilized representations nor considered the temporal consistency of data. Friel and Sanyal [2023] concentrated on QA tasks, using ChatGPT to generate responses and assigning them hallucination labels. However, this benchmark only provides the responses generated by a single LLM, which cannot evaluate the internal-representation-based methods across multiple LLMs.

6 Conclusions

In this paper, we introduce MHAD, a novel hallucination detection method. MHAD leverages the internal representations across layers during the generation progress of LLMs to detect hallucination. Moreover, we propose SOQHD, a novel hallucination detection benchmark for ODQA, which provides the internal representations of LLMs and ensures temporal consistency. Experimental results demonstrate the effectiveness and generalization of MHAD. We aspire for our work to contribute to the field of LLM research, enhancing the reliability of LLMs in real-world applications.

591 References

- 592 [Alain and Bengio, 2017] Guillaume Alain and Yoshua Ben- 645
593 gio. Understanding intermediate layers using linear clas- 646
594 sifier probes. In *ICLR (Workshop)*, 2017. 647
- 595 [Azaria and Mitchell, 2023] Amos Azaria and Tom Mitchell. 648
596 The internal state of an LLM knows when it’s lying. In 649
597 *Findings of EMNLP*, 2023. 650
- 598 [Bang *et al.*, 2023] Yejin Bang, Samuel Cahyawijaya, 651
599 Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy 652
600 Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. 653
601 Do, Yan Xu, and Pascale Fung. A multitask, multilin- 654
602 gual, multimodal evaluation of ChatGPT on reasoning, 655
603 hallucination, and interactivity. In *IJCNLP*, 2023. 656
- 604 [Bengio *et al.*, 2015] Samy Bengio, Oriol Vinyals, Navdeep 657
605 Jaitly, and Noam Shazeer. Scheduled sampling for se- 658
606 quence prediction with recurrent neural networks. In 659
607 *NeurIPS*, 2015. 660
- 608 [Burns *et al.*, 2023] Collin Burns, Haotian Ye, Dan Klein, 661
609 and Jacob Steinhardt. Discovering latent knowledge in 662
610 language models without supervision. In *ICLR*, 2023. 663
- 611 [Chen *et al.*, 2017] Danqi Chen, Adam Fisch, Jason Weston, 664
612 and Antoine Bordes. Reading Wikipedia to answer open- 665
613 domain questions. In *ACL*, 2017. 666
- 614 [Chen *et al.*, 2024] Chao Chen, Kai Liu, Ze Chen, Yi Gu, 667
615 Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. IN- 668
616 SIDE: llms’ internal states retain the power of hallucina- 669
617 tion detection. In *ICLR*, 2024. 670
- 618 [Chiang *et al.*, 2023] W L Chiang, Z Li, Z Lin, et al. Vicuna: 671
619 An open-source chatbot impressing gpt-4 with 90%* chat- 672
620 gpt quality. See <https://vicuna.lmsys.org> (accessed 14 673
621 April 2023), 2(3):6, 2023. 674
- 622 [Chuang *et al.*, 2024] Yung-Sung Chuang, Yujia Xie, 675
623 Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng 676
624 He. Dola: Decoding by contrasting layers improves 677
625 factuality in large language models. In *ICLR*, 2024. 678
- 626 [Du *et al.*, 2024] Xuefeng Du, Chaowei Xiao, and Yixuan 679
627 Li. Haloscope: Harnessing unlabeled llm generations for 680
628 hallucination detection. In *NeurIPS*, 2024. 681
- 629 [Friel and Sanyal, 2023] Robert Friel and Atindriyo Sanyal. 682
630 Chainpoll: A high efficacy method for llm hallucination 683
631 detection. *arXiv preprint arXiv:2310.18344*, 2023. 684
- 632 [Guu *et al.*, 2020] Kelvin Guu, Kenton Lee, Zora Tung, 685
633 Panupong Pasupat, and Mingwei Chang. Retrieval aug- 686
634 mented language model pre-training. In *ICML*, 2020. 687
- 635 [Han *et al.*, 2016] Song Han, Huizi Mao, and William J. 688
636 Dally. Deep compression: Compressing deep neural net- 689
637 work with pruning, trained quantization and huffman cod- 690
638 ing. In *ICLR*, 2016. 691
- 639 [Hedderich *et al.*, 2024] Michael A. Hedderich, Natalie N. 692
640 Bazarova, Wenting Zou, Ryun Shim, Xinda Ma, and Qian 693
641 Yang. A piece of theatre: Investigating how teachers de- 694
642 sign LLM chatbots to assist adolescent cyberbullying edu- 695
643 cation. In *CHI*, 2024. 696
- [Huang *et al.*, 2023] Lei Huang, Weijiang Yu, Weitao Ma, 644
Weihong Zhong, Zhangyin Feng, Haotian Wang, Qiang- 645
long Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 646
A survey on hallucination in large language models: Prin- 647
ciples, taxonomy, challenges, and open questions. *ACM 648
Transactions on Information Systems*, 2023. 649
- [Izacard and Grave, 2021] Gautier Izacard and Edouard 650
Grave. Leveraging passage retrieval with generative mod- 651
els for open domain question answering. In *EACL*, 2021. 652
- [Jawahar *et al.*, 2019] Ganesh Jawahar, Benoît Sagot, and 653
Djamé Seddah. What does BERT learn about the struc- 654
ture of language? In *ACL*, 2019. 655
- [Ji *et al.*, 2023] Ziwei Ji, Nayeon Lee, Rita Frieske, 656
Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, 657
Andrea Madotto, and Pascale Fung. Survey of hallucina- 658
tion in natural language generation. *ACM Comput. Surv.*, 659
2023. 660
- [Joshi *et al.*, 2017] Mandar Joshi, Eunsol Choi, Daniel Weld, 661
and Luke Zettlemoyer. TriviaQA: A large scale distantly 662
supervised challenge dataset for reading comprehension. 663
In *ACL*, 2017. 664
- [Kwiatkowski *et al.*, 2019] Tom Kwiatkowski, Jennimaria 665
Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, 666
Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob 667
Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, 668
Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob 669
Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A 670
benchmark for question answering research. *TACL*, 2019. 671
- [Lewis *et al.*, 2020] Patrick S. H. Lewis, Ethan Perez, Alek- 672
sandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman 673
Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, 674
Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 675
Retrieval-augmented generation for knowledge-intensive 676
NLP tasks. In *NeurIPS*, 2020. 677
- [Li *et al.*, 2023a] Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian- 678
Yun Nie, and Ji-Rong Wen. HaluEval: A large-scale hallu- 679
cination evaluation benchmark for large language models. 680
In *EMNLP*, 2023. 681
- [Li *et al.*, 2023b] Kenneth Li, Oam Patel, Fernanda B. 682
Viégas, Hanspeter Pfister, and Martin Wattenberg. 683
Inference-time intervention: Eliciting truthful answers 684
from a language model. In *NeurIPS*, 2023. 685
- [Manakul *et al.*, 2023] Potsawee Manakul, Adian Liusie, 686
and Mark Gales. SelfCheckGPT: Zero-resource black-box 687
hallucination detection for generative large language mod- 688
els. In *EMNLP*, 2023. 689
- [Meta, 2024] AI Meta. Introducing meta llama 3: The most 690
capable openly available llm to date. *Meta AI.*, 2024. 691
- [Min *et al.*, 2023] Sewon Min, Kalpesh Krishna, Xinxin Lyu, 692
Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, 693
Luke Zettlemoyer, and Hannaneh Hajishirzi. FACTScore: 694
Fine-grained atomic evaluation of factual precision in long 695
form text generation. In *EMNLP*, 2023. 696
- [Moon *et al.*, 2019] Seungwhan Moon, Pararth Shah, Anuj 697
Kumar, and Rajen Subba. OpenDialKG: Explainable 698

- 699 conversational reasoning with attention-based walks over
700 knowledge graphs. In *ACL*, 2019.
- 701 [Mündler *et al.*, 2024] Niels Mündler, Jingxuan He, Slobodan
702 Jenko, and Martin T. Vechev. Self-contradictory hallucinations of large language models: Evaluation, detection
703 and mitigation. In *ICLR*, 2024.
- 704
- 705 [Pan *et al.*, 2023] Yikang Pan, Liangming Pan, Wenhui Chen,
706 Preslav Nakov, Min-Yen Kan, and William Wang. On the
707 risk of misinformation pollution with large language models. In *Findings of EMNLP*, 2023.
- 708
- 709 [Park *et al.*, 2024] Kiho Park, Yo Joong Choe, and Victor
710 Veitch. The linear representation hypothesis and the geometry of large language models. In *ICML*, 2024.
- 711
- 712 [Ranzato *et al.*, 2016] Marc’Aurelio Ranzato, Sumit Chopra,
713 Michael Auli, and Wojciech Zaremba. Sequence level
714 training with recurrent neural networks. In *ICLR*, 2016.
- 715 [Sajjad *et al.*, 2022] Hassan Sajjad, Nadir Durrani, Fahim
716 Dalvi, Firoj Alam, Abdul Khan, and Jia Xu. Analyzing
717 encoded concepts in transformer language models. In
718 *NAACL-HLT*, 2022.
- 719 [Schulman, 2023] John Schulman. Reinforcement learning
720 from human feedback: Progress and challenges. In *Berkeley EECS Colloquium*. *YouTube* www.youtube.com/watch,
721 2023.
- 722
- 723 [See *et al.*, 2017] Abigail See, Peter J. Liu, and Christopher
724 D. Manning. Get to the point: Summarization with
725 pointer-generator networks. In *ACL*, 2017.
- 726 [Shen *et al.*, 2023] Yiqiu Shen, Laura Heacock, Jonathan
727 Elias, Keith D Hentel, Beatriu Reig, George Shih, and
728 Linda Moy. Chatgpt and other large language models are
729 double-edged swords, 2023.
- 730 [Su *et al.*, 2024] Weihang Su, Changyue Wang, Qingyao Ai,
731 Yiran Hu, Zhijing Wu, Yujia Zhou, and Yiqun Liu. Un-
732 supervised real-time hallucination detection based on the
733 internal states of large language models. In *Findings of
734 ACL*, 2024.
- 735 [Taori *et al.*, 2023] Rohan Taori, Ishaan Gulrajani, Tianyi
736 Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy
737 Liang, and Tatsunori B Hashimoto. Stanford alpaca: An
738 instruction-following llama model, 2023.
- 739 [Thirunavukarasu *et al.*, 2023] Arun James
740 Thirunavukarasu, Darren Shu Jeng Ting, Kabilan
741 Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel
742 Shu Wei Ting. Large language models in medicine.
743 *Nature medicine*, 2023.
- 744 [Touvron *et al.*, 2023] Hugo Touvron, Louis Martin, Kevin
745 Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei,
746 Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava,
747 Shruti Bhosale, et al. Llama 2: Open foundation and
748 fine-tuned chat models. *arXiv preprint arXiv:2307.09288*,
749 2023.
- 750 [Voita *et al.*, 2024] Elena Voita, Javier Ferrando, and
751 Christoforos Nalmpantis. Neurons in large language
752 models: Dead, n-gram, positional. In *Findings of ACL*,
753 2024.
- [Wang *et al.*, 2022] Xiaozhi Wang, Kaiyue Wen, Zhengyan
754 Zhang, Lei Hou, Zhiyuan Liu, and Juanzi Li. Finding skill
755 neurons in pre-trained transformer-based language mod-
756 els. In *EMNLP*, 2022.
- [Wang *et al.*, 2023] Xuezhi Wang, Jason Wei, Dale Schuur-
758 mans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha
759 Chowdhery, and Denny Zhou. Self-consistency improves
760 chain of thought reasoning in language models. In *ICLR*,
761 2023.
- [Wang *et al.*, 2024] Xu Wang, Cheng Li, Yi Chang, Jindong
763 Wang, and Yuan Wu. Negativeprompt: Leveraging psy-
764 chology for large language models enhancement via nega-
765 tive emotional stimuli. In *IJCAI*, 2024.
- [Wu *et al.*, 2023] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim
767 Dabrovolski, Mark Dredze, Sebastian Gehrmann, Prab-
768 hanjan Kambadur, David Rosenberg, and Gideon Mann.
769 Bloomberggpt: A large language model for finance. *arXiv
770 preprint arXiv:2303.17564*, 2023.
- [Xu *et al.*, 2024] Rongwu Xu, Zehan Qi, Zhijiang Guo,
772 Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu.
773 Knowledge conflicts for LLMs: A survey. In *EMNLP*,
774 2024.
- [Yang *et al.*, 2018] Zhilin Yang, Peng Qi, Saizheng Zhang,
776 Yoshua Bengio, William Cohen, Ruslan Salakhutdinov,
777 and Christopher D. Manning. HotpotQA: A dataset for
778 diverse, explainable multi-hop question answering. In
779 *EMNLP*, 2018.
- [Zhang *et al.*, 2023] Tianhang Zhang, Lin Qiu, Qipeng Guo,
781 Cheng Deng, Yue Zhang, Zheng Zhang, Chenghu Zhou,
782 Xinbing Wang, and Luoyi Fu. Enhancing uncertainty-
783 based hallucination detection with stronger focus. In
784 *EMNLP*, 2023.
- [Zhang *et al.*, 2024] Dongxu Zhang, Varun Gangal, Barrett
786 Lattimer, and Yi Yang. Enhancing hallucination detec-
787 tion through perturbation-based synthetic data generation
788 in system responses. In *Findings of ACL*, 2024.
- 789