LLMs Know More Than They Show: On the Intrinsic Representation of LLM Hallucinations

Anonymous authors
Paper under double-blind review

Abstract

Large language models (LLMs) often produce errors, including factual inaccuracies, biases, and reasoning failures, collectively referred to as "hallucinations". Recent studies have demonstrated that LLMs' internal states encode information regarding the truthfulness of their outputs, and that this information can be utilized to detect errors. In this work, we show that the internal representations of LLMs encode much more information about truthfulness than previously recognized. We first discover that the truthfulness information is concentrated in specific tokens, and leveraging this property significantly enhances error detection performance. Yet, we show that such error detectors fail to generalize across datasets, implying that—contrary to prior claims—truthfulness encoding is not universal but rather multifaceted. Next, we show that internal representations can also be used for predicting the types of errors the model is likely to make, facilitating the development of tailored mitigation strategies. Lastly, we reveal a discrepancy between LLMs' internal encoding and external behavior: they may encode the correct answer, yet consistently generate an incorrect one. Taken together, these insights deepen our understanding of LLM errors from the model's internal perspective, which can guide future research on enhancing error analysis and mitigation.[1]

1 Introduction

The ever-growing popularity of large language models (LLM) across many domains has brought a significant limitation to center stage: their tendency to "hallucinate" – which is often used to describe the generation of inaccurate information. But what are hallucinations, and what causes them? A considerable body of research has sought to define, taxonomize, and understand hallucinations through extrinsic, behavioral analysis, primarily examining how users perceive such errors (Bang et al., 2023; Ji et al., 2023; Huang et al., 2023a; Rawte et al., 2023). However, this approach does not adequately address how these errors are encoded within the LLMs. Alternatively, another line of work has explored the internal representations of LLMs, suggesting that LLMs encode signals of truthfulness (Kadavath et al., 2022; Li et al., 2024; Chen et al., 2024, *inter alia*). However, these analyses were typically restricted to detecting errors—determining whether a generated output contains inaccuracies—without delving deeper into how such signals are represented and could be leveraged to understand or mitigate hallucinations.

In this work, we reveal that the internal representations of LLMs encode much more information about truthfulness than previously recognized. Through a series of experiments, we train classifiers on these internal representations to predict various features related to the truthfulness of generated outputs. Our findings reveal the patterns and types of information encoded in model representations, linking this intrinsic data to extrinsic LLM behavior. This enhances our ability to detect errors (while understanding the limitations of error detection), and may guide the development of more nuanced strategies based on error types and mitigation methods that make use of the model's internal knowledge. Our experiments are designed to be general, covering a broad array of LLM limitations. While the term "hallucinations" is widely used, it lacks a universally accepted definition (Venkit et al., 2024). Our framework adopts a broad interpretation, considering hallucinations to encompass all errors produced by an LLM, including factual inaccuracies, biases, common-sense reasoning

[1]Our code is available in `https://github.com/technion-cs-nlp/LLMsKnow`.

failures, and other real-world errors. This approach enables us to draw general conclusions about model errors from a broad perspective.

Our first step is identifying where truthfulness signals are encoded in LLMs. Previous studies have suggested methods for detecting errors in LLM outputs using intermediate representations, logits, or probabilities, implying that LLMs may encode signals of truthfulness (Kadavath et al., 2022; Li et al., 2024; Chen et al., 2024). Focusing on long-form generations, which reflect real-world usage of LLMs, our analysis uncovers a key oversight: the choice of token used to extract these signals (Section 3). We find that **truthfulness information is concentrated in the exact answer tokens** – e.g., "Hartford" in "The capital of Connecticut is Hartford, an iconic city...". Recognizing this nuance significantly improves error detection strategies across the board, revealing that truthfulness encoding is stronger than previously observed.

From this point forward, we concentrate on our most effective strategy: a classifier trained on intermediate LLM representations within the exact answer tokens, referred to as 'probing classifiers' (Belinkov, 2021). This approach helps us explore what these representations reveal about LLMs. Our demonstration that a trained probing classifier can predict errors suggests that LLMs encode information related to their own truthfulness. However, we find that **probing classifiers do not generalize across different tasks** (Section 4). Generalization occurs only within tasks requiring similar skills (e.g., factual retrieval), indicating "skill-specific" truthfulness features. For tasks involving different skills, e.g., sentiment analysis, these classifiers are no better–or worse–than logit-based uncertainty predictors, challenging the idea of a "universal truthfulness" encoding proposed in previous work (Marks & Tegmark, 2023; Slobodkin et al., 2023). Instead, our results indicate that **LLMs encode multiple, distinct notions of truth**. Thus, deploying trainable error detectors in practical applications should be undertaken with caution.

After establishing the limitations of error detection and examining how error encoding differs across tasks, we delve deeper into errors within a single task, taxonomizing its errors based on the model's responses across repeated samples (Section 5). For example, the same error being consistently generated is different from an error that is generated occasionally among many other distinct errors. Using a different set of probing classifiers trained on the exact answer tokens, we find that **error types are predictable from the LLM representations**, drawing a connection between the models's internal representations and its external behavior. This raises an interesting possibility that LLMs encode information about the types of errors they may produce. This classification offers a more nuanced understanding of errors, enabling developers to predict error patterns and implement more targeted mitigation strategies.

Finally, we investigate the alignment between the LLM's internal representations and its external behavior (Section 6). Our findings highlight a significant disagreement in some cases, where the model's internal encoding contradicts its behavior. **The model's internal encoding may identify the correct answer–yet it frequently generates an incorrect response**. This discrepancy reveals that the LLM's external behavior may misrepresent its abilities, potentially pointing to new strategies for reducing errors by utilizing its existing strengths. Overall, our model-centric framework provides a deeper understanding of LLM errors, suggesting potential directions for improvements in error analysis and mitigation.

2 Background

Defining and characterizing LLM errors. The term "hallucinations" is widely used across various subfields such as conversational AI (Liu et al., 2022), abstractive summarization (Zhang et al., 2019), and machine translation (Wang & Sennrich, 2020), each interpreting the term differently. Yet, no consensus exists on defining hallucinations: Venkit et al. (2024) identified 31 distinct frameworks for conceptualizing hallucinations, revealing the diversity of perspectives. Research efforts aim to define and taxonomize hallucinations, distinguishing them from other error types (Liu et al., 2022; Ji et al., 2023; Huang et al., 2023a; Rawte et al., 2023). On the other hand, recent scholarly conversations introduce terms like "confabulations" (Millidge, 2023) and "fabrications" (McGowan et al., 2023), attributing a possible "intention" to LLMs, although the notions of LLM "intention" and other human-like traits are still debated (Salles et al., 2020; Serapio-García et al., 2023; Harnad, 2024). These categorizations, however, adopt a *human-centric* view by focusing on the subjective interpretations of LLM hallucinations, which does not necessarily reflect how these errors are

encoded within the models themselves. This gap limits our ability to address the root causes of hallucinations, or to reason about their nature. For example, it is unclear whether conclusions about hallucinations defined in one framework can be applied to another framework. Liang et al. (2024) defined hallucinations as inconsistencies with the training data. While this approach engage with the possible root causes of hallucinations, our study focuses on insights from the model itself, without requiring training data access. Instead, we adopt a broad interpretation of hallucinations. Here, we define hallucinations as any type of error generated by an LLM, including factual inaccuracies, biases, failures in common-sense reasoning, and others.

Another line of research suggests that LLMs either encode information about their own errors (Kadavath et al., 2022; Azaria & Mitchell, 2023) or exhibit discrepancies between their outputs and internal representations (Liu et al., 2023; Gottesman & Geva, 2024), indicating the presence of underlying mechanisms not reflected in their final outputs. Moreover, Yona et al. (2024) found that current LLMs fail to effectively convey their uncertainty through their generated outputs. Hence, we propose shifting the focus from human-centric interpretations of hallucinations to a *model-centric* perspective, examining the model's intermediate activations.

Error detection in LLMs. Error detection is a longstanding task in NLP, crucial for maintaining high standards in various practical applications and for constructing more reliable systems that ensure user trust (Bommasani et al., 2021). Over the years, many studies have proposed task-specific solutions (see Section A.1). However, the recent shift towards general-purpose LLMs necessitates a holistic approach capable of addressing any error type, rather than focusing on specific ones, making it suitable for the diverse errors generated by these models.

A line of work has addressed this challenge by leveraging external knowledge sources (Lewis et al., 2020; Gao et al., 2023) or an external LLM judge (Lin et al., 2021; Rawte et al., 2023) to identify erroneous outputs. On the other hand, our work focuses on detection methods that rely solely on the computations of the LLM—specifically, output logits, probabilities after softmax, and hidden states.

Error detection in LLMs is also closely linked to uncertainty estimation, where low certainty signals potential inaccuracies and possible errors. Popular methods to derive calibrated confidence from LLMs include inspecting the model logit output values (Varshney et al., 2023), examining agreement across multiple sampled answers (Kuhn et al., 2023; Manakul et al., 2023; Tian et al., 2023a) eliciting verbalized probability (Tian et al., 2023b), and direct prompting (Kadavath et al., 2022).

Another line of work trains probing classifiers to discover and utilize truthfulness features. This approach has shown some success by probing the final token of an answer–either generated (Kadavath et al., 2022; Snyder et al., 2023; Yuksekgonul et al., 2023; Zou et al., 2023; Yin et al., 2024; Chen et al., 2024; Simhi et al., 2024) or not (Li et al., 2024; Marks & Tegmark, 2023; Burns et al., 2022; Azaria & Mitchell, 2023; Rateike et al., 2023). Others probe the final token of the prompt before the response is generated (Slobodkin et al., 2023; Snyder et al., 2023; Simhi et al., 2024; Gottesman & Geva, 2024). Many previous studies simplify the analysis by generating answers in a few-shot setting or limiting generation to a single token. In contrast, we simulate real-world usage of LLMs by allowing unrestricted answer generation. By probing exact answer tokens, we achieve significant improvements in error detection.

3 Better Error Detection

This section presents our experiments on detecting LLM errors through their own computations, focusing on token selection's impact and introducing a method that outperforms other approaches.

3.1 Task Definition

Given an LLM M, an input prompt p and the LLM-generated response $\hat{y}$, the task is to predict whether $\hat{y}$ is correct or wrong. We assume that there is access to the LLM's internal states (i.e., white-box setting), but no access to any external resources (e.g., search engine or additional LLMs).

We use a dataset $D = \{(q_i, y_i)\}_{i=1}^{N}$, consisting of N question-label pairs, where $\{q_i\}_{i=1}^{N}$ represents a series of questions (e.g., *"What is the capital of Connecticut?"*) and $\{y_i\}_{i=1}^{N}$ the corresponding ground-truth answers (*"Hartford"*). For each question q_i, we prompt the model M to generate a

response y_i, resulting in the set of predicted answers $\{\hat{y}_i\}_{i=1}^N$ (*"The capital of Connecticut is Hartford..."*). Next, to build our error-detection dataset, we evaluate the correctness of each generated response $\hat{y}_i$ by comparing it to the ground-truth label y_i. This comparison yields a correctness label $z_i \in \{0, 1\}$ (1 correct, 0 wrong). The comparison can be done either via automatic heuristics or with the assistance of an instruct-LLM.[2] Our error detection dataset is: $\{(q_i, \hat{y}_i, z_i)\}_{i=1}^N$. Note that this dataset is defined based on the analyzed LLM and its generated answers. Any instances where the LLM refuses to answer are excluded, as these can easily be classified as incorrect.

3.2 Experimental Setup

Datasets and models. We perform all experiments on four LLMs: Mistral-7b (Jiang et al., 2023), Mistral-7b-instruct-v0.2 (denoted Mistral-7b-instruct), Llama3-8b (Touvron et al., 2023), and Llama3-8b-instruct. We consider 10 different datasets spanning various domains and tasks: TriviaQA (Joshi et al., 2017), HotpotQA with/without context (Yang et al., 2018), Natural Questions (Kwiatkowski et al., 2019), Winobias (Zhao et al., 2018), Winogrande (Sakaguchi et al., 2021), MNLI (Williams et al., 2018), Math (Sun et al., 2024), IMDB review sentiment analysis (Maas et al., 2011), and a dataset of movie roles (movies) that we curate. We allow unrestricted response generation to mimic real-world LLM usage, with answers decoded greedily. For more details on the datasets and the prompts used to generate answers, refer to Appendix A.3.

Performance metric. We measure the area under the ROC curve to evaluate error detectors, providing a single metric that reflects their ability to distinguish between positive and negative cases across many thresholds, balancing sensitivity (true positive rate) and specificity (false positive rate).

Error detection methods. We compare methods from both uncertainty and hallucinations literature.

- **Aggregated probabilities / logits:** Previous studies (Guerreiro et al., 2023; Kadavath et al., 2022; Varshney et al., 2023; Huang et al., 2023b) aggregate output token probabilities or logits to score LLM confidence for error detection. We implement several methods from the literature, calculating the minimum, maximum, or mean of these values. The main paper reports results for the most common approach, **Logits-mean**, and the best-performing one, **Logits-min**, with additional baselines in Appendix B.
- **P(True):** Kadavath et al. (2022) showed that LLMs are relatively calibrated when asked to evaluate the correctness of their generation via prompting. We implement this evaluation using the same prompt.
- **Probing:** Probing classifiers involve training a small classifier on a model's intermediate activations to predict features of processed text (Belinkov, 2021). Recent studies show their effectiveness for error detection in generated text (Kadavath et al., 2022, *inter alia*). An intermediate activation is a vector $h_{l,t}$ from a specific LLM layer l and (either read or generated) token t. Thus, each LLM generation produces multiple such activations. Following prior work, we use a linear probing classifier for error detection (Li et al., 2024, inter alia) on static tokens: the last generated token ($h_{l,-1}$), the one before it ($h_{l,-2}$), and the final prompt token ($h_{l,k}$). The layer l is selected per token based on validation set performance.

For further details on the implementation of each method, refer to Appendix A.4.

3.3 Exact Answer Tokens

Existing methods often overlook a critical nuance: the token selection for error detection, typically focusing on the last generated token or taking a mean. However, since LLMs typically generate long-form responses, this practice may miss crucial details (Brunner et al., 2020). Other approaches use the last token of the prompt (Slobodkin et al., 2023, *inter alia*), but this is inherently inaccurate due to LLMs' unidirectional nature, failing to account for the generated response and missing cases where different sampled answers from the same model vary in correctness. We investigate a previously unexamined token location: the *exact answer tokens*, which represent the most meaningful parts of the generated response. We define exact answer tokens as those whose modification alters the

[2]For most datasets, we use heuristics to predict correctness, except for one case. See Appendix A.2.

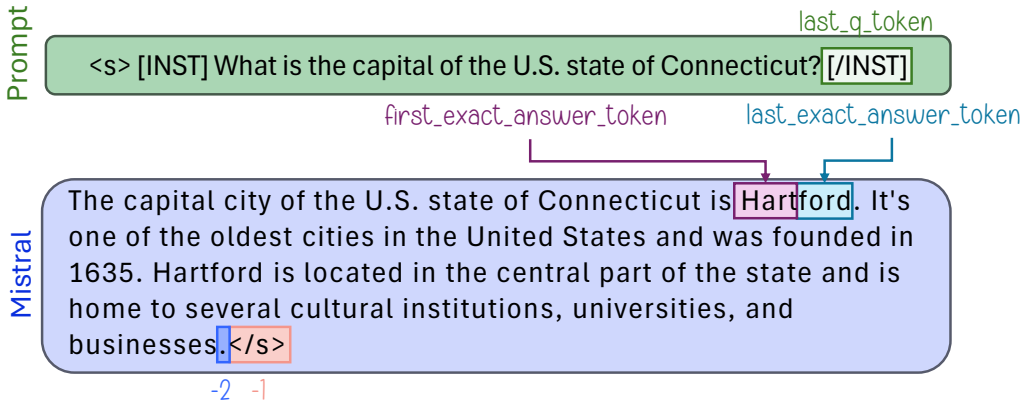
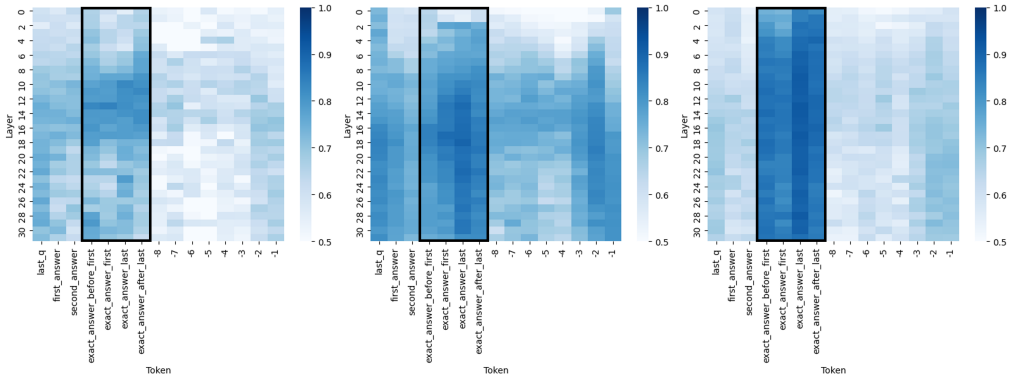


Figure 1: Example for the input and LLM output from the TriviaQA dataset, and the names of the tokens that can be probed.



(a) TriviaQA (b) Winobias (c) Math

Figure 2: AUC values of a probe error detector across layers and tokens, Mistral-7b-instruct. Generation proceeds from left to right, with detection performance peaking at the exact answer tokens.

answer's correctness, disregarding subsequent generated content.[3] Figure 1 illustrates the different token locations. In the following experiments, we implement each error detection method with an "exact answer" version, demonstrating that it often improves performance, especially in probing.

These exact answer is identified from a lengthy generated answer using an external algorithm, which processes the question and the LLM's response, $A(q_i, \hat{y}_i)$, to extract the exact answer. In our implementation, we use Mistral-7b-Instruct in a few-shot learning setup as A. However, all evaluated LLMs can extract exact answers from their own outputs (see Appendix A.2). After extraction, we identify the exact answer tokens via a simple search process, focusing on four key tokens: the one before the first exact answer token, the first and last exact answer tokens, and the one after the last.

3.4 Results

Patterns of truthfulness encoding. We first focus on probing classifiers to gain insights into the internal representations of LLMs. Specifically, we analyze the effects of layer and token selection on the error detection performance of these probing classifiers. By systematically probing all model layers, starting from the last question token to the final generated token, we observe consistent truthfulness encoding patterns. Figure 2 shows AUC metrics of probes across Mistral-7b-Instruct layers and tokens. Middle to later layers often yield the most effective probing results (see Appendix

[3]In practice, we do not use this definition for extracting the exact answer, but rather an instruct model in a few-shot setting. Still, the definition is useful to manually verify that automatic extractions work as expected.

Table 1: Comparison of error detection techniques using AUC metric, across different models and datasets. The best-performing method is **bolded**. Using exact answer tokens is useful for many cases, especially probing.

	Mistral-7b-Instruct			Llama 3-8b-Instruct		
	TriviaQA	Winobias	Math	TriviaQA	Winobias	Math
Logits-mean	0.60 $\pm$0.009	0.56 $\pm$0.017	0.55 $\pm$0.029	0.66 $\pm$0.005	0.60 $\pm$0.026	0.75 $\pm$0.018
Logits-mean-exact	0.68 $\pm$0.007	0.54 $\pm$0.012	0.51 $\pm$0.005	0.71 $\pm$0.006	0.55 $\pm$0.019	0.80 $\pm$0.021
Logits-min	0.63 $\pm$0.008	0.59 $\pm$0.012	0.51 $\pm$0.017	0.74 $\pm$0.007	0.61 $\pm$0.024	0.75 $\pm$0.016
Logits-min-exact	0.75 $\pm$0.006	0.53 $\pm$0.013	0.71 $\pm$0.009	0.79 $\pm$0.006	0.61 $\pm$0.019	0.89 $\pm$0.018
p(True)	0.66 $\pm$0.006	0.45 $\pm$0.021	0.48 $\pm$0.022	0.73 $\pm$0.008	0.59 $\pm$0.020	0.62 $\pm$0.017
p(True)-exact	0.74 $\pm$0.003	0.40 $\pm$0.021	0.60 $\pm$0.025	0.73 $\pm$0.005	0.63 $\pm$0.014	0.59 $\pm$0.018
Probe @ token						
Last generated [-1]	0.71 $\pm$0.006	0.82 $\pm$0.004	0.74 $\pm$0.008	0.81 $\pm$0.005	0.86 $\pm$0.007	0.82 $\pm$0.016
Before last generated [-2]	0.73 $\pm$0.004	0.85 $\pm$0.004	0.74 $\pm$0.007	0.75 $\pm$0.005	0.88 $\pm$0.005	0.79 $\pm$0.020
End of question	0.76 $\pm$0.008	0.82 $\pm$0.011	0.72 $\pm$0.007	0.77 $\pm$0.007	0.80 $\pm$0.018	0.72 $\pm$0.023
Exact	**0.85** $\pm$0.004	**0.92** $\pm$0.005	**0.92** $\pm$0.008	**0.83** $\pm$0.002	**0.93** $\pm$0.004	**0.95** $\pm$0.027

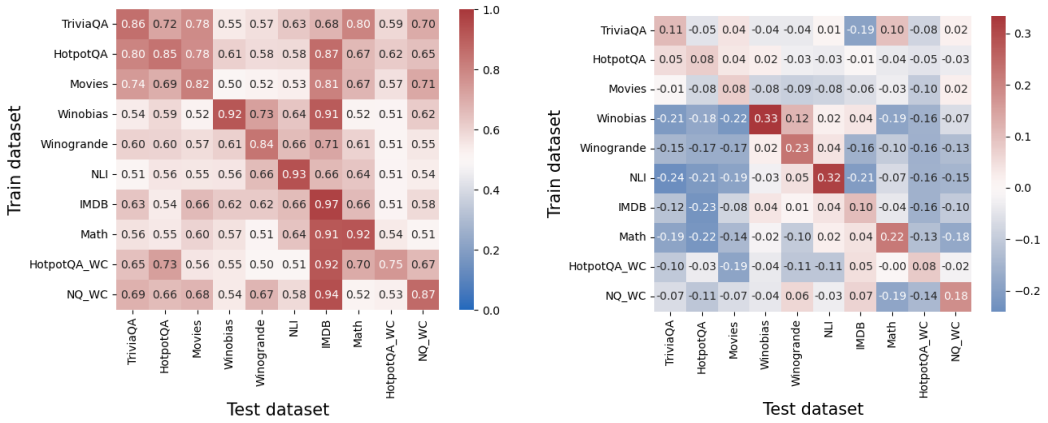
B for more datasets and models), aligning with previous studies on truthfulness encoding (Burns et al., 2022; CH-Wang et al., 2023) and transformer representations (nostalgebraist, 2020; Meng et al., 2022; Geva et al., 2023). Regarding tokens, a strong truthfulness signal appears immediately after the prompt, suggesting that this representation encodes information on the model's general ability to answer the question correctly. This signal weakens as text generation progresses but peaks again at the exact answer tokens. Towards the end of the generation process, signal strength rises again, though it remains weaker than at the exact answer tokens. These patterns are consistent across nearly all datasets and models (see Appendix B), suggesting a general mechanism by which LLMs encode and process truthfulness during text generation.

Error Detection Results. Next, we evaluate various error detection methods by comparing their performance with and without the use of exact answer tokens. Table 1 compares the AUC across three representative datasets (additional datasets and models in Appendix B, showing consistent patterns). Here we present results for the last exact answer token, which outperformed both the first exact answer token and the one preceding it, while the token following the last performed similarly. Incorporating the exact answer token improves the different error detection methods in almost all datasets. Notably, our probing technique (bottom line) consistently outperforms all other baselines across the board. While we did not compare all existing error detection methods, the primary conclusion is that information about truthfulness is highly localized in specific generated tokens, and that focusing on exact answer tokens leads to significant improvements in error detection.

4 Generalization Between Tasks

The effectiveness of a probing classifier in detecting errors suggests that LLMs encode information about the truthfulness of their outputs. This supports using probing classifiers for error detection in production, but their generalizability across tasks remains unclear. While some studies argue for a universal mechanism of truthfulness encoding in LLMs (Marks & Tegmark, 2023; Slobodkin et al., 2023), results on probe generalization across datasets are mixed (Kadavath et al., 2022; Marks & Tegmark, 2023; CH-Wang et al., 2023; Slobodkin et al., 2023; Levinstein & Herrmann, 2024). Understanding this is essential for real-world applications, where the error detector may encounter examples that significantly differ from those it was trained on. Therefore, we explore whether a probe trained on one dataset can detect errors in others.

Our generalization experiments are conducted between all of the datasets discussed in Section 3, covering a broader range of non-synthetic settings than previous work. We select the optimal token and layer combination for each dataset, train all probes using this combination on other datasets, and then test them on the original dataset. We evaluate generalization performance using the absolute AUC score, defined as $\max(\text{auc}, 1 - \text{auc})$, to also account for cases where the learned signal in one dataset is reversed in another.



(a) Raw AUC values. Values above 0.5 indicate some generalization.

(b) Performance (AUC) difference of the probe and the logit-based method. Values above 0 indicate generalization beyond the logit-based method.

Figure 3: Generalization between datasets, Mistral-7b-instruct. After subtracting the logit-based method's performance, we observe that most datasets show limited or no meaningful generalization.

Results. Figure 3a shows the generalization results for Mistral-7b-instruct, with similar patterns observed for other LLMs in Appendix C. In this context, values above 0.5 indicate successful generalization. At first glance, the results appear consistent with previous research: most heatmap values exceed 0.5, implying some degree of generalization across tasks. This observation supports the existence of a universal mechanism for decoding truthfulness, since the same linear directions—captured by the probe—encode truthfulness information across many datasets. However, upon closer inspection, it turns out that most of this performance can be achieved by logit-based truthfulness detection, which only observes the output logits. Figure 3b presents the same heatmap after subtracting results from our strongest logit-based baseline (Logit-min-exact). This adjusted heatmap reveals the probe's generalization rarely exceeds what can be achieved by examining logits alone. This implies that the apparent generalization does not stem from a universal internal encoding of truthfulness but rather reflects information already accessible through external features like logits.

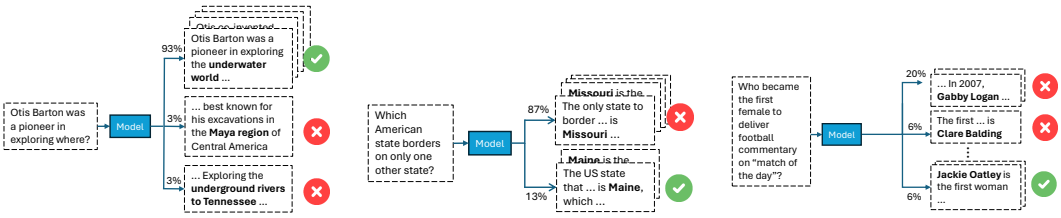
Nonetheless, we do observe some successful generalization in tasks requiring similar skills, such as parametric factual retrieval (TriviaQA, HotpotQA, Movies) and common-sense reasoning (Winobias, Wingrande, NLI). This suggests that, although the overall pattern of truthfulness signals across tokens appeared consistent across tasks (as observed in Section 3.4), LLMs have many "skill-specific" truthfulness mechanisms rather than universal ones. However, some patterns remain unexplained, such as the asymmetric generalization from TriviaQA to Math tasks. Overall, our findings indicate that models have a multifaceted representation of truthfulness. The internal mechanisms responsible for solving distinct problem are implemented as different mechanisms (e.g., circuits) within models (Elhage et al., 2021; Olah et al., 2023). Similarly, LLMs do not encode truthfulness through a single unified mechanism but rather through multiple mechanisms, each corresponding to different notions of truth. Further investigation is required to disentangle these mechanisms.

5 Investigating Error Types

Having established the limitations of error detection, we now shift to error analysis. Previously, we explored types of LLM limitations across *different* tasks, noting both commonalities and distinctions in their error representations. In this section, we focus on the types of errors LLMs make in a *specific* task—TriviaQA—which represents factual errors, a commonly studied issue in LLMs (Kadavath et al., 2022; Snyder et al., 2023; Li et al., 2024; Chen et al., 2024; Simhi et al., 2024).

5.1 Taxonomy of Errors

Intuitively, not all mistakes are identical. In one case, an LLM may consistently generate an incorrect answer, considering it correct, while in another case, it could issue a best guess. To analyze errors



(a) The LLM mostly answers correctly, but sometimes hallucinates.

(b) The LLM mostly answers incorrectly, but seems to have some knowledge on the correct answer.

(c) The LLM generates many different answers, one of them is the correct one which is generated a small fraction of the resamples.

Figure 4: Different error types in free-form generation, exposed when resampled many times.

from the LLM's perspective, we sample $K = 30$ responses at a temperature setting of $T = 1$[4] for each example in the dataset and then analyze the resulting distribution of answers.

Figure 4 illustrates three representative error types. In one (Figure 4a), the model usually gives the correct answer but occasionally make an error, implying correct information is present but sampling may lead to mistakes. In another (Figure 4b, the model often responds incorrectly, though it is capable of providing the right answer, indicating some retained knowledge despite consistently making the same error. In a third type (Figure 4c), the model generates a wide array of mostly incorrect answers, reflecting low confidence in any generated answer.

More generally, we categorize the errors by logging three specific features for each example: (a) the number of different answers generated; (b) the frequency of the correct answer; and (c) the frequency of the most common incorrect answer. These features reveal the following error patterns:

- **(A) Refuses to answer:** The model responds that it cannot answer the question in at least half the cases.
- **(B) Consistently correct:** Answers correctly in at least half of the cases. This category is divided into: (B1) always correct; and (B2) mostly correct with occasional errors.
- **(C) Consistently incorrect:** Consistently generates the same incorrect response in at least half of the cases. Similarly to type B, we subdivide this type into (C1) correct answer is never produced; and (C2) correct answer appears at least once.
- **(D) Two competing:** Generates both correct and incorrect responses at similar rates– difference in rates is 5 or less, and each response is generated at least 5 times.
- **(E) Many answers:** Generates over 10 distinct answers. Like types C and D, Subtypes include (E1) correct answer is never generated; and (E2) correct answer is generated at least once.

This taxonomy covers 96% of the errors in TriviaQA for Mistral-7b-instruct. Although some overlap exists between types, our goal is to identify general patterns and explore their connection to the models's internal representations. For a discussion on the design choices of this taxonomy, refer to Appendix G. This taxonomy classifies LLM errors based on an extrinsic, behavior-based analysis. Similarly, previous work analyzed repeated samples to assess an LLM's knowledge of the correct answer (Simhi et al., 2024; Gekhman et al., 2024). Our approach is distinct because it also examines the nature of errors that the LLM makes. Furthermore, as we discuss next, we analyze the connection between these behavioral patterns and the model's internal encoding.

5.2 Predicting Error Types

Our taxonomy offers an external, behavioral analysis of LLMs, which we complement by an intrinsic evaluation. We explore whether LLMs encode information on potential error types within their intermediate activations, offering a deeper insight into the underlying mechanisms. To investigate

[4]We chose $K = 30$ as the overall correctness seemed to plateau around this point; see Appendix D. We found that lower temperatures generally produced less truthful answers across repeated trials.

Table 2: AUC scores for error type classification. Error types are predictable from the inner model representations, indicating the encoding of fine-grained information on errors.

Error type	Mistral-7b	Mistral-Instr-7b	Llama3-8b	Llama3-Instr-8b
(A) Refuses to answer	$0.86_{\pm 0.002}$	$0.85_{\pm 0.011}$	$0.87_{\pm 0.002}$	$0.88_{\pm 0.014}$
(B) Consistently correct	$0.88_{\pm 0.001}$	$0.82_{\pm 0.008}$	$0.86_{\pm 0.001}$	$0.81_{\pm 0.002}$
(C) Consistently incorrect	$0.59_{\pm 0.002}$	$0.67_{\pm 0.002}$	$0.59_{\pm 0.002}$	$0.64_{\pm 0.003}$
(D) Two competing	$0.63_{\pm 0.002}$	$0.68_{\pm 0.006}$	$0.61_{\pm 0.001}$	$0.65_{\pm 0.004}$
(E) Many answers	$0.90_{\pm 0.001}$	$0.84_{\pm 0.003}$	$0.89_{\pm 0.001}$	$0.89_{\pm 0.001}$

this, we train a probe in a one-to-many setting, where a single probe identifies a specific error type from all others. We use representations extracted from the answers produced via greedy decoding.

Table 2 presents the test set results for all models. Our findings demonstrate that the error type can be predicted from the intermediate representations of the greedy decoding generations, suggesting that they may encode not just output correctness but also features that are correlative with fine-grained information about potential errors. This predictability opens up possibilities for targeted interventions on specific error types.

6 Detecting the Correct Answer

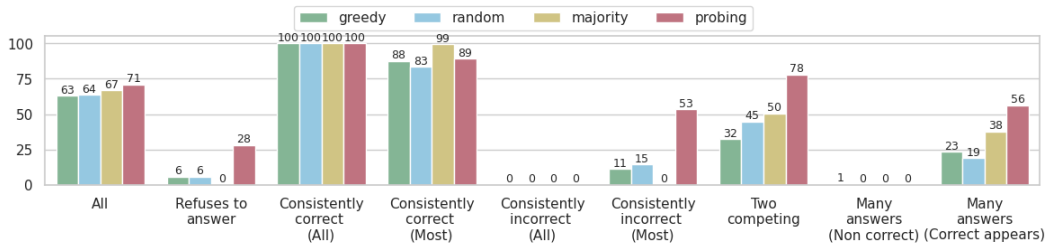
After identifying that models encode diverse truthfulness-related information, we examine how this *internal* truthfulness aligns with their *external* behavior during response generation. To this end, we use our probe,[5] trained on error detection, to select an answer from a pool of 30 generated responses to the same question. We then measure the model's accuracy based on the selected answers. A case where this accuracy does not significantly differ from traditional decoding methods (such as greedy decoding), suggests that the LLM's internal representation of truthfulness is consistent with its external behavior. In simpler terms, that the model is generating answers that it also internally considers as correct. Conversely, a case where using the probe alters performance either way, would suggest a misalignment between the LLM's internal representations and its actual behavior.

Experimental Setup The experiments were conducted on TriviaQA, Winobias, and Math. We resample each model answer in the same strategy described in Section 5.1. The final chosen answer is the one with the highest correctness probability, as assessed by the probe. We compare to three baselines: (1) greedy decoding, (2) random selection from the $K = 30$ answer candidates; and (3) majority vote wherein the most frequently generated answer is chosen.

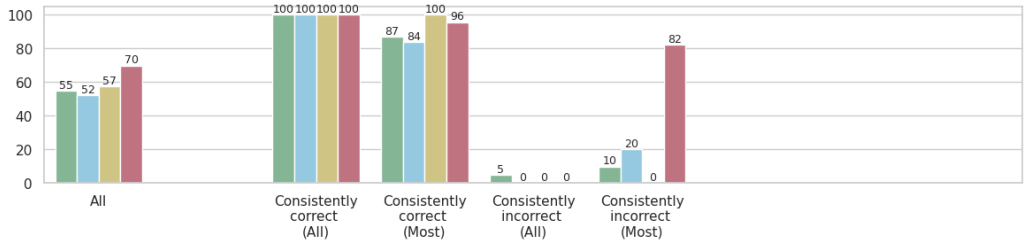
Results The results for Mistral-7b-instruct are summarized in Figure 5, with additional results for other LLMs and datasets as well as qualitative examples provided in Appendix E. We only present results on error types that appear 30 times or more in our test dataset. Overall, using the probe to select answers enhances the LLMs accuracy across all examined tasks. However, the extent of improvement varies by error type. For instance, in the TriviaQA dataset, there is minimal gain in the "mostly correct" category (B2). In contrast, substantial gains—ranging from 30 to 40 points in some cases—are observed in the "mostly incorrect" (C2), "two competing answers" (D), and "many answers" (E1) categories. Interestingly, and perhaps surprisingly, the probe is most effective in cases where the LLM lacks any (external) preference for the correct answer during generation. The fact that the probe can effectively identify the correct answer in these scenarios, points at a significant disconnect between the LLM's internal encoding and its external behavior. These results suggest that even when the model encodes information of which answer is correct, it can still generate an incorrect answer in practice.

While using the probe to select the answer proves effective, it is not proposed here as an error mitigation strategy but rather as a diagnostic tool. However, these findings indicate that further research in this area could leverage the existing knowledge within LLMs to significantly reduce errors. We recommend exploring this direction in future investigations.

[5]We choose the best-performing probe for each task, which is trained on the last exact answer token.



(a) TriviaQA



(b) Math

Figure 5: Different answer choice strategies, Mistral-7B-Instruct. A notable improvement in accuracy by using the error-detection probe is observed for error types where the LLM shows no preference for the correct answer across repeated generations.

7 Discussion and Conclusions

In this study, we examined LLM errors through their internal representations, finding that truthfulness-related information is localized in specific tokens. This enables practical improvements in error detection for production-level models. Our approach relies on access to internal representations, limiting its use to open-source models. This study focused on Q&A tasks with clear gold labels, leaving open-ended tasks like summarization for future research. We expect these findings to provide a foundation for extending the methodology to other settings.

Our results showed that truthfulness features generalize poorly across tasks and datasets. Some unexplained generalization patterns point to hidden links between unrelated tasks, warranting further research. Possible directions for improving generalization is to examine the effects of layer-token combinations on generalization and to explore training on diverse datasets, as was done by Bürger et al. (2024). Deciphering task-specific truthfulness features and their overlaps across tasks could also yield valuable insights for improving classifier design.

We also found that the intermediate representations of LLMs can be used to predict the types of errors they might make. As Simhi et al. (2024) noted, different error types may require distinct mitigation strategies, such as retrieval-augmented generation (Lewis et al., 2020) in some cases or fine-tuning in others. Identifying these error types is useful for customizing these strategies.

Lastly, we identified a significant discrepancy between the model's external behavior and internal states, where it repeatedly outputs incorrect responses despite internally encoding the correct answer. It is possible that mechanisms favoring likelihood override those promoting truthfulness, as LLMs are trained to predicting likely tokens, which does not necessarily align with factual accuracy. Our findings imply that these models already encode valuable information that could possibly be harnessed to reduce errors. Work by Chuang et al. (2024) shows promising results in this area.

The generalization results underscore the need for caution when using a trained error detector across varied settings. However, task-specific probes could be valuable in high-stakes domains like medicine or law, where reliability is crucial. These probes can detect wrong answers, predict error types, and guide the selection of the correct response from resampled outputs, offering significant practical benefits. Practical guidelines on applying these probes are in Appendix F. In conclusion, our findings highlight the value of LLM internal representations in understanding errors and provide pathways for improving error detection and mitigation in future work.

8 REPRODUCIBILITY STATEMENT

To ensure reproducibility of our work, we provide detailed instructions and necessary code. The source code, including scripts for generating model answers, probing, resampling, and error type analysis, is available in the supplementary material, where we also provide command examples and specific seeds used for experiment reproducibility. This repository includes documentation on how to set up the environment, download and preprocess datasets, and execute the experiments outlined in Sections 3–6 of the paper. Additionally, all datasets, models, and results generation steps are described in the Appendix A.

REFERENCES

Alexandre Allauzen. Error detection in confusion network. In *8th Annual Conference of the International Speech Communication Association, INTERSPEECH 2007, Antwerp, Belgium, August 27-31, 2007*, pp. 1749–1752. ISCA, 2007. doi: 10.21437/INTERSPEECH.2007-490. URL https://doi.org/10.21437/Interspeech.2007-490.

Amos Azaria and Tom Mitchell. The internal state of an llm knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, 2023.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*, 2023.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances, 2021. URL https://arxiv.org/abs/2102.12452.

Samuel J. Bell, Helen Yannakoudakis, and Marek Rei. Context is key: Grammatical error detection with contextual word representations. In Helen Yannakoudakis, Ekaterina Kochmar, Claudia Leacock, Nitin Madnani, Ildikó Pilán, and Torsten Zesch (eds.), *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@ACL 2019, Florence, Italy, August 2, 2019*, pp. 103–115. Association for Computational Linguistics, 2019. doi: 10.18653/V1/W19-4410. URL https://doi.org/10.18653/v1/w19-4410.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. On identifiability in transformers. In *8th International Conference on Learning Representations (ICLR 2020)(virtual)*. International Conference on Learning Representations, 2020.

Lennart Bürger, Fred A Hamprecht, and Boaz Nadler. Truth is universal: Robust detection of lies in llms. *arXiv preprint arXiv:2407.12831*, 2024.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2022.

Andrew Caines, Christian Bentz, Kate M. Knill, Marek Rei, and Paula Buttery. Grammatical error detection in transcriptions of spoken english. In Donia Scott, Núria Bel, and Chengqing Zong (eds.), *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pp. 2144–2162. International Committee on Computational Linguistics, 2020. doi: 10.18653/V1/2020.COLING-MAIN.195. URL https://doi.org/10.18653/v1/2020.coling-main.195.

Sky CH-Wang, Benjamin Van Durme, Jason Eisner, and Chris Kedzie. Do androids know they're only dreaming of electric sheep?, 2023.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. INSIDE: LLMs' internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Zj12nzlQbz.

Wei Chen, Sankaranarayanan Ananthakrishnan, Rohit Kumar, Rohit Prasad, and Prem Natarajan. ASR error detection in a conversational spoken language translation system. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, pp. 7418–7422. IEEE, 2013. doi: 10.1109/ICASSP.2013.6639104. URL https://doi.org/10.1109/ICASSP.2013.6639104.

Yong Cheng and Mofan Duan. Chinese grammatical error detection based on BERT model. In Erhong YANG, Endong XUN, Baolin ZHANG, and Gaoqi RAO (eds.), *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pp. 108–113, Suzhou, China, December 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.nlptea-1.15.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng 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.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

Rahhal Errattahi, Asmaa El Hannani, and Hassan Ouahmane. Automatic speech recognition errors detection and correction: A review. In Mourad Abbas and Ahmed Abdelali (eds.), *1st International Conference on Natural Language and Speech Processing, ICNLSP 2015, Algiers, Algeria, October 18-19, 2015*, volume 128 of *Procedia Computer Science*, pp. 32–37. Elsevier, 2015. doi: 10.1016/J.PROCS.2018.03.005. URL https://doi.org/10.1016/j.procs.2018.03.005.

Dan Flickinger, Michael Wayne Goodman, and Woodley Packard. Uw-stanford system description for AESW 2016 shared task on grammatical error detection. In Joel R. Tetreault, Jill Burstein, Claudia Leacock, and Helen Yannakoudakis (eds.), *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications, BEA@NAACL-HLT 2016, June 16, 2016, San Diego, California, USA*, pp. 105–111. The Association for Computer Linguistics, 2016. doi: 10.18653/V1/W16-0511. URL https://doi.org/10.18653/v1/w16-0511.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, et al. Rarr: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16477–16508, 2023.

Zorik Gekhman, Roee Aharoni, Genady Beryozkin, Markus Freitag, and Wolfgang Macherey. KoBE: Knowledge-based machine translation evaluation. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3200–3207, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.287. URL https://aclanthology.org/2020.findings-emnlp.287.

Zorik Gekhman, Dina Zverinski, Jonathan Mallinson, and Genady Beryozkin. RED-ACE: Robust error detection for ASR using confidence embeddings. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 2800–2808, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.180. URL https://aclanthology.org/2022.emnlp-main.180.

Zorik Gekhman, Jonathan Herzig, Roee Aharoni, Chen Elkind, and Idan Szpektor. TrueTeacher: Learning factual consistency evaluation with large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2053–2070, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.127. URL https://aclanthology.org/2023.emnlp-main.127.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. Does fine-tuning llms on new knowledge encourage hallucinations?, 2024.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. *arXiv preprint arXiv:2304.14767*, 2023.

Daniela Gottesman and Mor Geva. Estimating knowledge in large language models without generating a single token. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP 2024)*, Miami, Florida, 2024. Association for Computational Linguistics.

Nuno M Guerreiro, Elena Voita, and André FT Martins. Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 1059–1075, 2023.

Stevan Harnad. Language writ large: Llms, chatgpt, grounding, meaning and understanding. *arXiv preprint arXiv:2402.02243*, 2024.

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. q^2: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7856–7870, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.619. URL https://aclanthology.org/2021.emnlp-main.619.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. TRUE: Re-evaluating factual consistency evaluation. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3905–3920, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.287. URL https://aclanthology.org/2022.naacl-main.287.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*, 2023a.

Yuheng Huang, Jiayang Song, Zhijie Wang, Huaming Chen, and Lei Ma. Look before you leap: An exploratory study of uncertainty measurement for large language models. *arXiv preprint arXiv:2307.10236*, 2023b.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 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)*, pp. 1601–1611, 2017.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Sudhanshu Kasewa, Pontus Stenetorp, and Sebastian Riedel. Wronging a right: Generating better errors to improve grammatical error detection. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 4977–4983. Association for Computational Linguistics, 2018. URL https://aclanthology.org/D18-1541/.

Hadas Kotek, Rikker Dockum, and David Sun. Gender bias and stereotypes in large language models. In *Proceedings of the ACM collective intelligence conference*, pp. 12–24, 2023.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. Evaluating the factual consistency of abstractive text summarization. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9332–9346, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.750. URL https://aclanthology.org/2020.emnlp-main.750.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=VD-AYtP0dve.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*, 2019.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177, 2022. doi: 10.1162/tacl_a_00453. URL https://aclanthology.org/2022.tacl-1.10.

Benjamin A Levinstein and Daniel A Herrmann. Still no lie detector for language models: Probing empirical and conceptual roadblocks. *Philosophical Studies*, pp. 1–27, 2024.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33: 9459–9474, 2020.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36, 2024.

Wei Li and Houfeng Wang. Detection-correction structure via general language model for grammatical error correction. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 1748–1763. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.acl-long.96.

Xun Liang, Shichao Song, Zifan Zheng, Hanyu Wang, Qingchen Yu, Xunkai Li, Rong-Hua Li, Yi Wang, Zhonghao Wang, Feiyu Xiong, et al. Internal consistency and self-feedback in large language models: A survey. *arXiv preprint arXiv:2407.14507*, 2024.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

Kevin Liu, Stephen Casper, Dylan Hadfield-Menell, and Jacob Andreas. Cognitive dissonance: Why do language model outputs disagree with internal representations of truthfulness? In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4791–4797, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.291. URL https://aclanthology.org/2023.emnlp-main.291.

Tianyu Liu, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, and Bill Dolan. A token-level reference-free hallucination detection benchmark for free-form text generation. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6723–6737, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.464. URL https://aclanthology.org/2022.acl-long.464.

Chi-kiu Lo. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Marco Turchi, and Karin Verspoor (eds.), *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pp. 507–513, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5358. URL https://aclanthology.org/W19-5358.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

Potsawee Manakul, Adian Liusie, and Mark Gales. 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*, pp. 9004–9017, 2023.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*, 2023.

Alessia McGowan, Yunlai Gui, Matthew Dobbs, Sophia Shuster, Matthew Cotter, Alexandria Selloni, Marianne Goodman, Agrima Srivastava, Guillermo A Cecchi, and Cheryl M Corcoran. Chatgpt and bard exhibit spontaneous citation fabrication during psychiatry literature search. *Psychiatry Research*, 326:115334, 2023.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 36, 2022. arXiv:2202.05262.

Beren Millidge. LLMs confabulate not hallucinate. *Beren's Blog*, March 2023. URL https://www.beren.io/2023-03-19-LLMs-confabulate-not-hallucinate/.

Ritika Mishra and Navjot Kaur. A survey of spelling error detection and correction techniques. *International Journal of Computer Trends and Technology*, 4(3):372–374, 2013.

nostalgebraist. Interpreting gpt: The logit lens. LessWrong blog post, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens. Accessed: 2024-11-18.

Chris Olah, Nelson Elhage, Neel Nanda, Catherine Schubert, Daniel Filan, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

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. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Thomas Pellegrini and Isabel Trancoso. Error detection in broadcast news ASR using markov chains. In Zygmunt Vetulani (ed.), *Human Language Technology. Challenges for Computer Science and Linguistics - 4th Language and Technology Conference, LTC 2009, Poznan, Poland, November 6-8, 2009, Revised Selected Papers*, volume 6562 of *Lecture Notes in Computer Science*, pp. 59–69. Springer, 2009. doi: 10.1007/978-3-642-20095-3_6. URL https://doi.org/10.1007/978-3-642-20095-3_6.

Amy Pu, Hyung Won Chung, Ankur Parikh, Sebastian Gehrmann, and Thibault Sellam. Learning compact metrics for MT. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 751–762, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.58. URL https://aclanthology.org/2021.emnlp-main.58.

Gaoqi Rao, Erhong Yang, and Baolin Zhang. Overview of NLPTEA-2020 shared task for Chinese grammatical error diagnosis. In Erhong YANG, Endong XUN, Baolin ZHANG, and Gaoqi RAO (eds.), *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pp. 25–35, Suzhou, China, December 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.nlptea-1.4.

Miriam Rateike, Celia Cintas, John Wamburu, Tanya Akumu, and Skyler Speakman. Weakly supervised detection of hallucinations in llm activations. *arXiv preprint arXiv:2312.02798*, 2023.

Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar, SM Tonmoy, Aman Chadha, Amit P Sheth, and Amitava Das. The troubling emergence of hallucination in large language models–an extensive definition, quantification, and prescriptive remediations. *arXiv preprint arXiv:2310.04988*, 2023.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. COMET: A neural framework for MT evaluation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2685–2702, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.213. URL https://aclanthology.org/2020.emnlp-main.213.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In Philipp Koehn, Loïc Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Tom Kocmi, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Marco Turchi, and Marcos Zampieri (eds.), *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pp. 578–585, Abu Dhabi, United Arab Emirates (Hybrid), December 2022a. Association for Computational Linguistics. URL https://aclanthology.org/2022.wmt-1.52.

Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and André F. T. Martins. CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task. In Philipp Koehn, Loïc Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Tom Kocmi, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Marco Turchi, and Marcos Zampieri (eds.), *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pp. 634–645, Abu Dhabi, United Arab Emirates (Hybrid), December 2022b. Association for Computational Linguistics. URL https://aclanthology.org/2022.wmt-1.60.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Arleen Salles, Kathinka Evers, and Michele Farisco. Anthropomorphism in ai. *AJOB neuroscience*, 11(2):88–95, 2020.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. QuestEval: Summarization asks for fact-based evaluation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6594–6604, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational

Linguistics. doi: 10.18653/v1/2021.emnlp-main.529. URL https://aclanthology.org/2021.emnlp-main.529.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. BLEURT: Learning robust metrics for text generation. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7881–7892, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.704. URL https://aclanthology.org/2020.acl-main.704.

Greg Serapio-García, Mustafa Safdari, Clément Crepy, Luning Sun, Stephen Fitz, Peter Romero, Marwa Abdulhai, Aleksandra Faust, and Maja Matarić. Personality traits in large language models. *arXiv preprint arXiv:2307.00184*, 2023.

Adi Simhi, Jonathan Herzig, Idan Szpektor, and Yonatan Belinkov. Constructing benchmarks and interventions for combating hallucinations in llms, 2024.

Aviv Slobodkin, Omer Goldman, Avi Caciularu, Ido Dagan, and Shauli Ravfogel. The curious case of hallucinatory (un)answerability: Finding truths in the hidden states of over-confident large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3607–3625, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.220. URL https://aclanthology.org/2023.emnlp-main.220.

Ben Snyder, Marius Moisescu, and Muhammad Bilal Zafar. On early detection of hallucinations in factual question answering, 2023. URL https://arxiv.org/abs/2312.14183.

Yuhong Sun, Zhangyue Yin, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Hui Zhao. Benchmarking hallucination in large language models based on unanswerable math word problem. *CoRR*, 2024.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. Fine-tuning language models for factuality. *arXiv preprint arXiv:2311.08401*, 2023a.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *arXiv preprint arXiv:2305.14975*, 2023b.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation, 2023.

Pranav Narayanan Venkit, Tatiana Chakravorti, Vipul Gupta, Heidi Biggs, Mukund Srinath, Koustava Goswami, Sarah Rajtmajer, and Shomir Wilson. " confidently nonsensical?": A critical survey on the perspectives and challenges of'hallucinations' in nlp. *arXiv preprint arXiv:2404.07461*, 2024.

Chaojun Wang and Rico Sennrich. On exposure bias, hallucination and domain shift in neural machine translation. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3544–3552, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.326. URL https://aclanthology.org/2020.acl-main.326.

Quanbin Wang and Ying Tan. Grammatical error detection with self attention by pairwise training. In *2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020*, pp. 1–7. IEEE, 2020. doi: 10.1109/IJCNN48605.2020.9206715. URL https://doi.org/10.1109/IJCNN48605.2020.9206715.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/N18-1101.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.

Fan Yin, Jayanth Srinivasa, and Kai-Wei Chang. Characterizing truthfulness in large language model generations with local intrinsic dimension. In *Proceedings of the 41st International Conference on Machine Learning*, Vienna, Austria, 2024.

Gal Yona, Roee Aharoni, and Mor Geva. Can large language models faithfully express their intrinsic uncertainty in words?, 2024. URL https://arxiv.org/abs/2405.16908.

Mert Yuksekgonul, Varun Chandrasekaran, Erik Jones, Suriya Gunasekar, Ranjita Naik, Hamid Palangi, Ece Kamar, and Besmira Nushi. Attention satisfies: A constraint-satisfaction lens on factual errors of language models. In *The Twelfth International Conference on Learning Representations*, 2023.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced language representation with informative entities. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1441–1451, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1139. URL https://aclanthology.org/P19-1139.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Gender bias in coreference resolution: Evaluation and debiasing methods. *arXiv preprint arXiv:1804.06876*, 2018.

Lina Zhou, Yongmei Shi, Jinjuan Feng, and Andrew Sears. Data mining for detecting errors in dictation speech recognition. *IEEE Trans. Speech Audio Process.*, 13(5-1):681–688, 2005. doi: 10.1109/TSA.2005.851874. URL https://doi.org/10.1109/TSA.2005.851874.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to ai transparency, 2023. URL https://arxiv.org/abs/2310.01405.

A IMPLEMENTATION DETAILS

A.1 TASK SPECIFIC ERROR DETECTION

In this work, we specifically address errors produced by modern large language models (LLMs). Given the diverse range of tasks these models are applied to, our focus is on general error detection across all categories, rather than isolating specific types. Prior to the emergence of LLMs, much research targeted error detection for specific tasks, with common examples including grammatical errors (Kasewa et al., 2018; Bell et al., 2019; Cheng & Duan, 2020; Wang & Tan, 2020; Flickinger et al., 2016), spelling mistakes (Mishra & Kaur, 2013), machine translation inaccuracies (Lo, 2019; Pu et al., 2021; Sellam et al., 2020; Gekhman et al., 2020; Rei et al., 2020; 2022a;b), speech recognition faults (Caines et al., 2020; Rao et al., 2020; Li & Wang, 2024; Zhou et al., 2005; Allauzen, 2007; Gekhman et al., 2022; Errattahi et al., 2015; Pellegrini & Trancoso, 2009; Chen et al., 2013), and factual consistency failures (Honovich et al., 2022; Laban et al., 2022; Honovich et al., 2021; Gekhman et al., 2023; Scialom et al., 2021; Kryscinski et al., 2020).

A.2 PROBING: IMPLEMENTATION DETAILS

We examine the intermediate representations of the exact answer tokens generated by a large language model (LLM) during the answer generation process. The intermediate representation selected for this analysis is derived from the output of the final multi-layer perceptron (MLP). This choice is based on preliminary experiments comparing the MLP output, the residual stream, and the attention heads, which showed no significant differences. We leave the in-depth analysis for future work.

For the probing classifier, we employ a logistic regression model from the scikit-learn library (Pedregosa et al., 2011). We used the default hyperparameters, which include a norm penalty of L2 and an LBFGS solver. We initially experimented with other hyper-parameters and did not find a singnificant difference. For each random seed, the dataset was split to training and validation in a 80-20 ratio, and the test dataset was bootstrap sampled.

Obtaining correctness label for the probing dataset. An answer is generally considered correct if it includes the correct answer label and appears before any alternative incorrect labels. We manually analyzed the results of this heuristic to confirm that it is accurate in almost all cases. However, one exception is the Natural Questions with Context (NQ_WC) dataset, where we identified false negatives and thus deployed a more precise validation using an instruct LLM, as demonstrated below:

> Evaluate the following answers to questions. For each question you would be given an LLM answer and the correct answer. You would have to determine if the LLM answer is correct or not. If the LLM answer is correct, write '1' and if it is not correct, write '0'. For example:
>
> Question: `[Question 1]`
>
> Ground Truth: `[Gold label 1]`
>
> LLM Answer: `[LLM long answer 1]`
>
> Correctness: `0`
>
> Question: `[Question 2]`
>
> Ground Truth: `[Gold label 2]`
>
> LLM Answer: `[LLM long answer 2]`
>
> Correctness: `1`
>
> Question: `[Question]`
>
> Ground Truth: `[Label]`
>
> LLM Answer: `[LLM long answer]`
>
> Correctness:

Table 3: Success rate of extracting exact answer from a long model answer. Each model is used to extract answers from its own output.

Mistral-7b	Mistral-Instruct-7b	Llama3-8b	Llama3-Instruct-8b
0.99	0.96	0.99	0.95

Detecting and using exact answer tokens. To detect the exact locations of answer tokens, we use a combination of heuristic methods and an instruction-tuned LLM. Specifically, when the set of possible answers is finite, we rely on heuristics. For more open-ended scenarios, such as factual questions, we automatically locate the answer if it matches the gold label. Otherwise, we prompt an instruction-tuned LLM, specifically Mistral-7b-Instruct (Jiang et al., 2023), to identify and extract the exact answer substring using the following prompt:

> Extract from the following long answer the short answer, only the relevant tokens. If the long answer does not answer the question, output NO ANSWER.
>
> Q: `[Question 1]`
>
> A: `[LLM long answer 1]`
>
> Exact answer: [Short exact answer 1]
>
> Q: `[Question 2]`
>
> A: `[LLM long answer that does not answer the question]`
>
> Exact answer: NO ANSWER
>
> Q: `[Question]`
>
> A: `[LLM long answer]` Exact answer:

To extract a valid exact answer from a long response, we prompt the instruct LLM up to five times. This process involves verifying that the exact answer is a substring of the long answer unless the instruct LLM indicates that there is no answer. To avoid bias in our probing task, we only retain questions for which a valid exact answer was successfully extracted. This ensures there is no unfair correlation between invalid answers and incorrect answers in the experiments.

We note the following: (a) While it is possible to use an instruct LLM to extract every answer regardless of its correctness, we chose the aforementioned strategy to improve the efficiency of our experiments; (b) This is just one possible implementation. For each LLM, one could use the same LLM to extract its own exact answer token, as demonstrated in a proof-of-concept over 1000 samples of TriviaQA in Table 3. Alternatively, it may be more efficient to train a smaller system specifically designed for detecting exact answer tokens, which would be more suitable for real-world scenarios. We choose to keep the extraction process as abstract as possible, as our primary focus is not on the specific implementation, but on analyzing the potential gains from probing these locations.

Additionally, if the exact answer token is not among the first generated tokens, we examine the token immediately preceding it ("before exact answer token"). If the exact answer token is not the last one, we also examine the following token. When the exact answer spans multiple tokens, the first and last exact answer tokens are probed separately.

A.3 Datasets

We outline here all ten datasets that we investigate in our work. In our analysis, we aimed at covering a wide range of tasks, skills required to solve the tasks, diversity of datasets and as a result also different LLM limitations such as factual inaccuracies (often referred to as "hallucinations"), biases, arithmetic mistakes, and more. For each dataset, we explain how it covers something different from all the previous datasets. For all datasets, we present the LLM with non or a short instruct, a context (if exists for the task), and let it generate a free text. We follow this paradigm as it better mimics real-world usage of LLMs by humans, as opposed to using few-shot to force a short answer that is generated on the first token (Yuksekgonul et al., 2023; Chen et al., 2024; Simhi et al., 2024). One exception to this is a the sentiment analysis (IMDB) for which we apply 1-shot for the LLM

to use the allowed labels, as it did not follow the instruction alone and we could not identify if the answer is correct or not even with manual analysis. Additionally, we implemented a different prompting strategy to the instruct and non-instruct LLMs. To see the exact formats we used to prompt each dataset and LLM, refer to our code implementation at `https://github.com/technion-cs-nlp/LLMsKnow`.

For each dataset we used a split of 10K training samples and 10K test samples, unless the dataset is too small, in which case we mention the size.

- **TriviaQA** (Joshi et al., 2017): a collection of trivia question-answer pairs. The questions are presented to the LLM without any context, allowing it to generate responses based solely on its internal, parametric knowledge. The dataset includes various acceptable variations of the correct answer, which are used to automatically evaluate the accuracy of the generated res.
- **HotpotQA** (Yang et al., 2018): a dataset designed for diverse multi-hop question answering. Each entry includes Wikipedia documents that help answering the questions. We use two different settings: (1) without context, where questions are asked directly, which covers slightly different skills from TriviaQA as it requires reasoning in addition to factual knowledge; and (2) with context (**HotpotQA_WC**), where the additional context is provided, emphasizing the ability to adhere to and utilize contextual information to solve the task.
- **Movies:** to further investigate generalization, we focused on a case of classic "hallucinations", involving factual knowledge, within a non-diverse dataset. This approach allowed us to test whether generalization to other types of errors is influenced by the type of error (factual versus others) or by the dataset's diversity. For this purpose, we created the movies dataset consisting of prompts in the form: "Who acted as [figure name] in the movie [movie name]?" The figures, movies, and correct answers were sourced from "The Movies Dataset" in Kaggle: `https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset`, which is based on the MovieLens website.
- **Winogrande** (Sakaguchi et al., 2021): we use this dataset to explore errors in common-sense reasoning. It consists of Winograd-style coreference challenges, where each example presents a sentence containing two entities and a pronoun. The objective is to determine which entity the pronoun refers to, relying on common-sense reasoning. For example, in the sentence: "The trophy doesn't fit into the suitcase because it's too **large**," the pronoun "it" refers to the trophy, not the suitcase.
- **Winobias** (Zhao et al., 2018): this benchmark focuses on coreference resolution in the context of gender bias, revealing a different type of limitation in LLMs. Each example consists of two professions: one stereotypically male and one stereotypically female, along with a gendered pronoun. The task requires the LLM to determine which profession the pronoun refers to. The sentences are unambiguous, with one correct answer. In some cases, the correct answer aligns with the stereotype, while in others, it is anti-stereotypical. For example, in the sentence "The developer argued with the designer because she did not like the design," "she" refers to the developer, which is an anti-stereotypical case since "developer" is considered a stereotypically male profession. Research has shown that LLMs often perform poorly on anti-stereotypical sentences (Zhao et al., 2018) and tend to base their decisions on stereotypes rather than on common-sense reasoning or linguistic rules (Kotek et al., 2023). Each split contains around 1500 samples.
- **NLI** (Natural Language Inference): NLI involves determining whether a given "hypothesis" is true (entailment), false (contradiction), or undetermined (neutral) based on a provided "premise." For this purpose, we use the MNLI dataset (Williams et al., 2018). NLI tasks address a distinct aspect of common-sense reasoning and are generally considered complex. This complexity allows us to investigate whether a model's generalization ability is related to the difficulty of the task it was trained on, or to other factors, such as the limited diversity of labels (NLI has only three valid labels) or the type of task.
- **Math** (Sun et al., 2024): this dataset includes both unanswerable and answerable math problems. In our study, we focus exclusively on the answerable problems, as our aim is to assess the correctness of the LLM's outputs, which requires a known correct answer (gold

standard). This task introduces an additional, previously unexplored skill of arithmetic reasoning. The train-test split consists of approximately 2,000 and 650 samples, respectively.

- **IMDB** (Maas et al., 2011): contains movie reviews used for the task of sentiment classification.
- **Natural Questions With Context** (Kwiatkowski et al., 2019): the Natural Questions (NQ) dataset is designed to evaluate and train automatic question-answering systems. It consists of real, anonymized queries submitted by users to Google, with answers extracted from Wikipedia, as well as the relevant Wikipedia pages which can be given in context. We included this dataset to introduce an additional challenge that requires adherence to context, complementing the HotpotQA with context dataset.

A.4 Baselines: Implementation Details

Aggregated probabilities / logits. Inspired by prior work (Kadavath et al., 2022; Guerreiro et al., 2023), we compute an aggregated score using the log-probabilities or raw probabilities of the generated text tokens $y_1, y_2, \ldots, y_N$ produced by the generative large language model (LLM). For instance, the following formulation is used to compute the Logits-mean baseline on the entire generated answer:

$$\frac{1}{N}\sum_{i=1}^{N}\mathbb{P}(y_i|Q, y_1, ..., y_{i-1}) \tag{1}$$

We also explore aggregation strategies that focus solely on the exact answer tokens (PE-Exact). Following Varshney et al. (2023), we also experiment with aggregating the minimum and maximum values (PE-[Min—Max]-[Exact]), alongside the mean aggregation described in Equation 1.

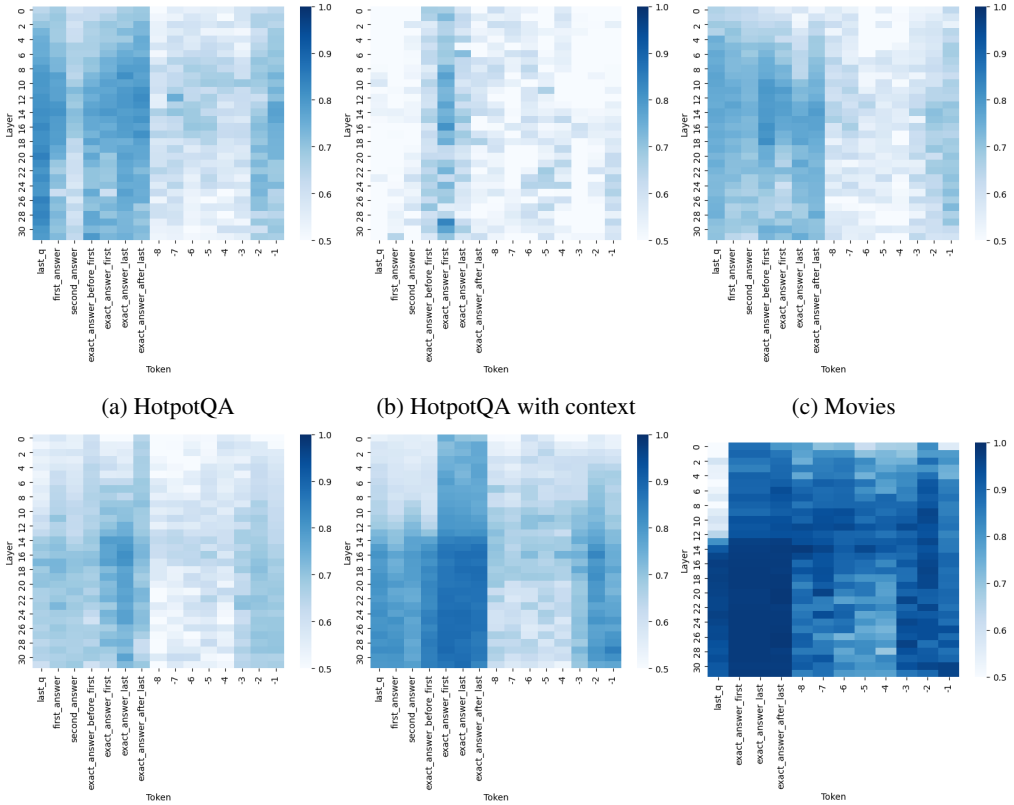
P(True): We follow Kadavath et al. (2022) and prompt the LLM to judge whether its answer is correct. Our prompt followed the following template, from Kadavath et al. (2022):

> Question: `[Question]`
>
> Proposed Answer: `[LLM long answer]`
>
> Is the proposed answer:
>
> (A) True
>
> (B) False
>
> The proposed answer is:

B Full Error Detection Results

Figure 6 presents the AUC values of a traind probe across layers and token for Mistral-7b-instruct, showing a similar pattern across all datasets. We also observe similar patterns across other models. See our repo `https://github.com/technion-cs-nlp/LLMsKnow`for the figures.

Tables 4, 5, 6, and 7 present the full error detection results across all baselines and datasets, which are consistent with the main paper results.



(a) HotpotQA (b) HotpotQA with context (c) Movies

(d) Winogrande (e) NLI (f) IMDB

Figure 6: AUC values of a probe error detector across layers and tokens, Mistral-7b-instruct. The detection performance spikes at the exact answer tokens.

Table 4: Comparison of error detection performance (AUC) on Mistral-7B.

	Mistral-7B				
	TriviaQA	Winobias	Math	Movies	IMDB
Logits-mean	0.67 ±0.004	0.49 ±0.010	0.41 ±0.015	0.67 ±0.007	0.88 ±0.064
Logits-mean-exact	0.67 ±0.004	0.50 ±0.010	0.56 ±0.026	0.68 ±0.008	0.57 ±0.080
Logits-min	0.80 ±0.003	0.45 ±0.014	0.48 ±0.021	0.73 ±0.006	0.78 ±0.056
Logits-min-exact	0.80 ±0.005	0.53 ±0.014	0.78 ±0.032	0.72 ±0.005	0.57 ±0.080
Logits-max	0.53 ±0.008	0.49 ±0.010	0.42 ±0.023	0.54 ±0.005	0.83 ±0.076
Logits-max-exact	0.54 ±0.009	0.50 ±0.010	0.40 ±0.024	0.58 ±0.007	0.57 ±0.080
Probas-mean	0.76 ±0.003	0.53 ±0.018	0.66 ±0.016	0.72 ±0.007	0.87 ±0.041
Probas-mean-exact	0.78 ±0.002	0.55 ±0.014	0.62 ±0.016	0.74 ±0.007	0.83 ±0.057
Probas-min	0.82 ±0.003	0.52 ±0.013	0.82 ±0.020	0.73 ±0.006	0.86 ±0.032
Probas-min-exact	**0.85** ±0.003	0.58 ±0.011	0.84 ±0.015	0.74 ±0.006	0.83 ±0.057
Probas-max	0.53 ±0.008	0.50 ±0.016	0.43 ±0.025	0.55 ±0.008	0.80 ±0.074
Probas-max-exact	0.55 ±0.009	0.51 ±0.013	0.39 ±0.019	0.59 ±0.009	0.83 ±0.057
p(True)	0.57 ±0.007	0.53 ±0.019	0.56 ±0.027	0.51 ±0.003	0.65 ±0.004
p(True)-exact	0.56 ±0.006	0.55 ±0.026	0.57 ±0.036	0.52 ±0.003	0.65 ±0.003
Probe @ token					
Last generated [-1]	0.83 ±0.002	0.65 ±0.008	0.82 ±0.023	0.79 ±0.002	0.85 ±0.007
Before last generated [-2]	0.82 ±0.003	0.84 ±0.012	0.83 ±0.019	0.78 ±0.003	0.95 ±0.004
End of question	0.74 ±0.005	0.78 ±0.012	0.83 ±0.016	0.77 ±0.002	0.81 ±0.009
Exact answer last	0.84 ±0.005	**0.89** ±0.007	**0.96** ±0.008	0.78 ±0.003	**0.95** ±0.004
Exact answer last+1	0.84 ±0.004	0.84 ±0.012	0.95 ±0.010	**0.80** ±0.002	0.85 ±0.007
	HotpotQA	HotpotQA-WC	Winogrande	NLI	NQ-WC
Logits-mean	0.63 ±0.005	0.52 ±0.009	0.49 ±0.004	0.51 ±0.004	0.69 ±0.006
Logits-mean-exact	0.57 ±0.008	0.52 ±0.007	0.50 ±0.003	**0.93** ±0.004	0.72 ±0.005
Logits-min	0.72 ±0.008	0.59 ±0.006	0.50 ±0.007	0.53 ±0.005	0.65 ±0.009
Logits-min-exact	0.72 ±0.007	0.65 ±0.004	0.51 ±0.007	0.49 ±0.006	0.70 ±0.005
Logits-max	0.54 ±0.007	0.49 ±0.010	0.48 ±0.005	0.48 ±0.005	0.59 ±0.012
Logits-max-exact	0.48 ±0.010	0.44 ±0.007	0.50 ±0.003	0.48 ±0.005	0.58 ±0.009
Probas-mean	0.65 ±0.004	0.55 ±0.006	0.51 ±0.007	0.49 ±0.003	0.63 ±0.008
Probas-mean-exact	0.62 ±0.006	0.56 ±0.007	0.51 ±0.005	0.02 ±0.001	0.66 ±0.007
Probas-min	0.73 ±0.005	0.58 ±0.007	0.52 ±0.009	0.53 ±0.004	0.63 ±0.011
Probas-min-exact	0.78 ±0.005	0.66 ±0.004	0.52 ±0.008	0.49 ±0.005	0.69 ±0.006
Probas-max	0.54 ±0.008	0.49 ±0.007	0.50 ±0.005	0.47 ±0.004	0.52 ±0.004
Probas-max-exact	0.48 ±0.010	0.44 ±0.005	0.50 ±0.004	0.48 ±0.003	0.53 ±0.012
p(True)	0.55 ±0.007	0.54 ±0.006	0.51 ±0.005	0.51 ±0.003	0.52 ±0.008
p(True)-exact	0.61 ±0.005	0.54 ±0.006	0.61 ±0.006	0.51 ±0.006	0.53 ±0.014
Probe @ token					
Last generated [-1]	0.78 ±0.006	0.67 ±0.004	0.51 ±0.007	0.77 ±0.004	0.78 ±0.003
Before last generated [-2]	0.79 ±0.007	0.69 ±0.007	0.66 ±0.004	0.81 ±0.002	0.75 ±0.006
End of question	0.72 ±0.007	0.56 ±0.003	0.51 ±0.007	0.88 ±0.004	0.70 ±0.005
Exact answer last	0.80 ±0.008	**0.74** ±0.007	**0.69** ±0.006	0.84 ±0.004	0.81 ±0.009
Exact answer last+1	**0.81** ±0.008	0.72 ±0.005	0.59 ±0.005	0.75 ±0.006	**0.84** ±0.007

Table 5: Comparison of error detection performance (AUC) on Mistral-7B-Instruct.

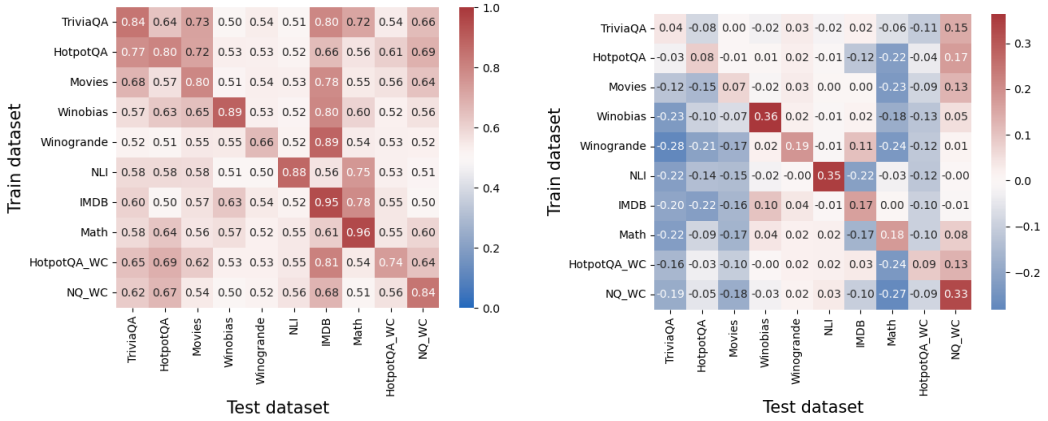
	Mistral-7B-Instruct				
	TriviaQA	Winobias	Math	Movies	IMDB
Logits-mean	0.60 ±0.009	0.56 ±0.017	0.55 ±0.029	0.63 ±0.005	0.57 ±0.006
Logits-mean-exact	0.68 ±0.007	0.54 ±0.012	0.51 ±0.005	0.70 ±0.004	0.87 ±0.007
Logits-min	0.63 ±0.008	0.59 ±0.012	0.51 ±0.017	0.66 ±0.008	0.52 ±0.007
Logits-min-exact	0.75 ±0.006	0.53 ±0.013	0.71 ±0.009	0.74 ±0.005	0.87 ±0.007
Logits-max	0.54 ±0.005	0.53 ±0.012	0.54 ±0.039	0.54 ±0.004	0.47 ±0.004
Logits-max-exact	0.55 ±0.004	0.54 ±0.011	0.32 ±0.015	0.61 ±0.006	0.87 ±0.007
Probas-mean	0.60 ±0.007	0.58 ±0.018	0.56 ±0.028	0.61 ±0.002	0.54 ±0.008
Probas-mean-exact	0.71 ±0.003	0.57 ±0.015	0.71 ±0.014	0.74 ±0.006	0.84 ±0.007
Probas-min	0.59 ±0.008	0.58 ±0.014	0.50 ±0.025	0.60 ±0.008	0.51 ±0.010
Probas-min-exact	0.74 ±0.004	0.57 ±0.016	0.75 ±0.011	0.73 ±0.006	0.84 ±0.007
Probas-max	0.50 ±0.006	0.41 ±0.010	0.53 ±0.009	0.51 ±0.005	0.48 ±0.004
Probas-max-exact	0.51 ±0.007	0.54 ±0.010	0.45 ±0.015	0.60 ±0.003	0.84 ±0.007
p(True)	0.68 ±0.005	0.45 ±0.021	0.48 ±0.026	0.62 ±0.005	0.62 ±0.009
p(True)-exact	0.74 ±0.003	0.40 ±0.021	0.60 ±0.025	0.69 ±0.008	0.60 ±0.009
Probe @ token					
Last generated [-1]	0.71 ±0.006	0.82 ±0.004	0.74 ±0.008	0.72 ±0.005	0.92 ±0.010
Before last generated [-2]	0.73 ±0.004	0.85 ±0.004	0.74 ±0.007	0.72 ±0.006	0.94 ±0.006
End of question	0.76 ±0.008	0.82 ±0.011	0.72 ±0.007	0.74 ±0.003	0.96 ±0.006
Exact answer last	0.85 ±0.004	**0.92** ±0.005	**0.92** ±0.008	0.81 ±0.003	**0.97** ±0.005
Exact answer last+1	**0.86** ±0.006	0.88 ±0.006	0.90 ±0.010	**0.82** ±0.003	0.96 ±0.006
	HotpotQA	HotpotQA-WC	Winogrande	NLI	NQ-WC
Logits-mean	0.61 ±0.002	0.55 ±0.009	0.59 ±0.004	0.64 ±0.006	0.71 ±0.008
Logits-mean-exact	0.66 ±0.009	0.55 ±0.004	0.49 ±0.004	0.57 ±0.004	0.69 ±0.009
Logits-min	0.61 ±0.003	0.53 ±0.013	0.61 ±0.003	0.62 ±0.002	0.67 ±0.008
Logits-min-exact	0.77 ±0.004	0.67 ±0.013	0.48 ±0.004	0.54 ±0.005	0.69 ±0.006
Logits-max	0.53 ±0.008	0.51 ±0.011	0.52 ±0.006	0.59 ±0.008	0.63 ±0.011
Logits-max-exact	0.51 ±0.011	0.41 ±0.010	0.49 ±0.007	0.64 ±0.003	0.63 ±0.013
Probas-mean	0.63 ±0.003	0.56 ±0.010	0.58 ±0.005	0.62 ±0.005	0.68 ±0.010
Probas-mean-exact	0.72 ±0.006	0.66 ±0.010	0.46 ±0.004	0.57 ±0.003	0.65 ±0.008
Probas-min	0.58 ±0.003	0.52 ±0.008	0.59 ±0.002	0.58 ±0.008	0.65 ±0.014
Probas-min-exact	0.76 ±0.004	0.68 ±0.010	0.46 ±0.005	0.57 ±0.003	0.66 ±0.008
Probas-max	0.50 ±0.005	0.53 ±0.003	0.48 ±0.007	0.52 ±0.007	0.51 ±0.005
Probas-max-exact	0.46 ±0.010	0.46 ±0.010	0.48 ±0.004	0.53 ±0.004	0.52 ±0.018
p(True)	0.54 ±0.006	0.54 ±0.004	0.53 ±0.003	0.58 ±0.003	0.57 ±0.006
p(True)-exact	0.60 ±0.008	0.48 ±0.005	0.57 ±0.011	0.65 ±0.004	0.57 ±0.009
Probe @ token					
Last generated [-1]	0.72 ±0.005	0.64 ±0.005	0.74 ±0.005	0.85 ±0.004	0.82 ±0.006
Before last generated [-2]	0.73 ±0.006	0.64 ±0.004	0.76 ±0.004	0.87 ±0.002	0.84 ±0.009
End of question	0.80 ±0.003	0.63 ±0.003	0.71 ±0.007	0.79 ±0.004	0.85 ±0.010
Exact answer last	0.85 ±0.003	0.75 ±0.006	**0.84** ±0.005	**0.93** ±0.003	0.86 ±0.003
Exact answer last+1	**0.85** ±0.002	**0.76** ±0.004	0.80 ±0.004	0.92 ±0.004	**0.87** ±0.006

Table 6: Comparison of error detection performance (AUC) on Llama-8b.

	Llama-8b				
	TriviaQA	Winobias	Math	Movies	IMDB
Logits-mean	0.58 ±0.006	0.44 ±0.015	0.43 ±0.026	0.64 ±0.008	0.77 ±0.007
Logits-mean-exact	0.63 ±0.007	0.50 ±0.015	0.50 ±0.028	0.64 ±0.008	0.77 ±0.007
Logits-min	0.75 ±0.007	0.50 ±0.022	0.45 ±0.042	0.73 ±0.005	0.73 ±0.007
Logits-min-exact	0.76 ±0.003	0.53 ±0.009	0.75 ±0.022	0.73 ±0.005	0.77 ±0.007
Logits-max	0.48 ±0.006	0.48 ±0.009	0.42 ±0.027	0.53 ±0.005	0.72 ±0.007
Logits-max-exact	0.52 ±0.007	0.49 ±0.014	0.35 ±0.026	0.53 ±0.005	0.77 ±0.007
Probas-mean	0.64 ±0.006	0.41 ±0.008	0.61 ±0.029	0.71 ±0.007	0.70 ±0.008
Probas-mean-exact	0.72 ±0.005	0.50 ±0.018	0.54 ±0.026	0.72 ±0.006	0.88 ±0.003
Probas-min	0.79 ±0.008	0.43 ±0.004	0.75 ±0.044	0.74 ±0.005	0.68 ±0.005
Probas-min-exact	0.82 ±0.003	0.53 ±0.014	0.78 ±0.022	0.74 ±0.005	0.88 ±0.003
Probas-max	0.49 ±0.006	0.50 ±0.009	0.46 ±0.032	0.53 ±0.007	0.60 ±0.009
Probas-max-exact	0.53 ±0.008	0.50 ±0.018	0.36 ±0.032	0.54 ±0.007	0.88 ±0.003
p(True)	0.62 ±0.005	0.48 ±0.011	0.53 ±0.027	0.61 ±0.005	0.51 ±0.010
p(True)-exact	0.67 ±0.002	0.53 ±0.017	0.63 ±0.028	0.58 ±0.005	0.52 ±0.008
Probe @ token					
Last generated [-1]	0.77 ±0.005	0.59 ±0.024	0.83 ±0.013	0.82 ±0.005	0.94 ±0.002
Before last generated [-2]	0.76 ±0.012	0.58 ±0.021	0.82 ±0.032	0.79 ±0.004	0.96 ±0.002
End of question	0.73 ±0.005	0.77 ±0.012	0.80 ±0.027	0.78 ±0.005	0.68 ±0.009
Exact answer last	**0.82** ±0.006	**0.91** ±0.007	**0.96** ±0.010	0.80 ±0.005	**0.97** ±0.001
Exact answer last+1	0.82 ±0.006	0.86 ±0.008	0.95 ±0.007	**0.82** ±0.006	0.95 ±0.003
	HotpotQA	HotpotQA-WC	Winogrande	NLI	NQ-WC
Logits-mean	0.65 ±0.004	0.62 ±0.006	0.48 ±0.003	0.47 ±0.002	0.53 ±0.010
Logits-mean-exact	0.55 ±0.003	0.54 ±0.006	0.49 ±0.004	0.48 ±0.002	0.58 ±0.009
Logits-min	0.57 ±0.004	0.49 ±0.003	0.48 ±0.003	0.48 ±0.007	0.58 ±0.009
Logits-min-exact	0.69 ±0.002	0.68 ±0.006	0.49 ±0.003	0.48 ±0.007	0.61 ±0.010
Logits-max	0.61 ±0.005	0.60 ±0.004	0.48 ±0.003	0.52 ±0.003	0.51 ±0.008
Logits-max-exact	0.47 ±0.003	0.46 ±0.005	0.49 ±0.004	0.51 ±0.002	0.54 ±0.005
Probas-mean	0.67 ±0.002	0.62 ±0.006	0.49 ±0.002	0.48 ±0.004	0.57 ±0.003
Probas-mean-exact	0.62 ±0.005	0.56 ±0.005	0.51 ±0.002	0.46 ±0.006	0.64 ±0.007
Probas-min	0.62 ±0.006	0.51 ±0.002	0.49 ±0.003	0.50 ±0.010	0.62 ±0.005
Probas-min-exact	0.76 ±0.005	0.67 ±0.004	0.51 ±0.002	0.50 ±0.010	0.69 ±0.008
Probas-max	0.61 ±0.004	0.58 ±0.004	0.48 ±0.002	0.48 ±0.003	0.51 ±0.012
Probas-max-exact	0.49 ±0.003	0.44 ±0.004	0.51 ±0.003	0.47 ±0.002	0.56 ±0.005
p(True)	0.52 ±0.007	0.45 ±0.005	0.54 ±0.004	0.54 ±0.007	0.56 ±0.006
p(True)-exact	0.58 ±0.005	0.50 ±0.007	0.64 ±0.004	0.62 ±0.005	0.61 ±0.002
Probe @ token					
Last generated [-1]	0.76 ±0.007	0.57 ±0.006	0.59 ±0.006	0.89 ±0.002	0.66 ±0.010
Before last generated [-2]	0.74 ±0.007	0.58 ±0.005	0.59 ±0.005	0.94 ±0.002	0.63 ±0.008
End of question	0.71 ±0.006	0.53 ±0.004	0.48 ±0.003	0.91 ±0.001	0.66 ±0.004
Exact answer last	0.81 ±0.006	0.77 ±0.004	**0.65** ±0.004	**0.94** ±0.002	**0.75** ±0.008
Exact answer last+1	**0.82** ±0.004	**0.79** ±0.001	0.57 ±0.004	0.90 ±0.002	0.75 ±0.007

Table 7: Comparison of error detection performance (AUC) on Llama-8b-Instruct.

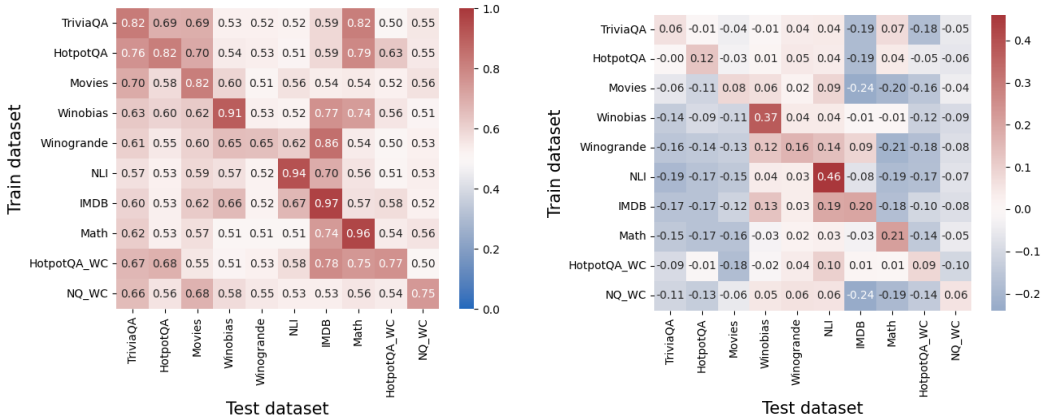
	Llama-8b-Instruct				
	TriviaQA	Winobias	Math	Movies	IMDB
Logits-mean	0.66 ±0.005	0.60 ±0.026	0.75 ±0.018	0.75 ±0.005	0.59 ±0.017
Logits-mean-exact	0.71 ±0.006	0.55 ±0.019	0.80 ±0.021	0.72 ±0.004	0.88 ±0.012
Logits-min	0.74 ±0.007	0.61 ±0.024	0.75 ±0.016	0.71 ±0.005	0.55 ±0.016
Logits-min-exact	0.79 ±0.006	0.61 ±0.019	0.89 ±0.018	0.77 ±0.006	0.88 ±0.012
Logits-max	0.54 ±0.007	0.55 ±0.013	0.73 ±0.027	0.67 ±0.003	0.51 ±0.009
Logits-max-exact	0.58 ±0.005	0.54 ±0.019	0.64 ±0.014	0.61 ±0.003	0.88 ±0.012
Probas-mean	0.67 ±0.006	0.63 ±0.024	0.66 ±0.033	0.73 ±0.006	0.73 ±0.015
Probas-mean-exact	0.75 ±0.009	0.61 ±0.014	0.83 ±0.022	0.74 ±0.005	0.74 ±0.021
Probas-min	0.67 ±0.009	0.65 ±0.019	0.64 ±0.036	0.65 ±0.004	0.57 ±0.016
Probas-min-exact	0.79 ±0.008	0.62 ±0.014	0.86 ±0.024	0.74 ±0.005	0.74 ±0.021
Probas-max	0.54 ±0.003	0.49 ±0.020	0.57 ±0.022	0.64 ±0.006	0.49 ±0.008
Probas-max-exact	0.56 ±0.007	0.55 ±0.016	0.57 ±0.018	0.61 ±0.003	0.74 ±0.021
p(True)	0.73 ±0.008	0.59 ±0.020	0.62 ±0.017	0.66 ±0.004	0.60 ±0.006
p(True)-exact	0.73 ±0.005	0.63 ±0.014	0.59 ±0.018	0.63 ±0.006	0.76 ±0.004
Probe @ token					
Last generated [-1]	0.81 ±0.005	0.86 ±0.007	0.82 ±0.016	0.78 ±0.004	0.81 ±0.014
Before last generated [-2]	0.75 ±0.005	0.88 ±0.005	0.79 ±0.020	0.82 ±0.005	0.83 ±0.006
End of question	0.77 ±0.007	0.80 ±0.018	0.72 ±0.023	0.76 ±0.005	0.87 ±0.006
Exact answer last	**0.83** ±0.002	**0.93** ±0.004	**0.95** ±0.027	0.85 ±0.005	**0.96** ±0.003
Exact answer last+1	0.83 ±0.006	0.90 ±0.005	0.94 ±0.023	**0.86** ±0.004	0.95 ±0.004
	HotpotQA	HotpotQA-WC	Winogrande	NLI	NQ-WC
Logits-mean	0.65 ±0.002	0.56 ±0.004	0.58 ±0.007	0.59 ±0.009	0.65 ±0.006
Logits-mean-exact	0.66 ±0.008	0.57 ±0.005	0.48 ±0.003	0.49 ±0.010	0.67 ±0.005
Logits-min	0.67 ±0.008	0.55 ±0.007	0.60 ±0.008	0.53 ±0.009	0.68 ±0.004
Logits-min-exact	0.76 ±0.010	0.65 ±0.010	0.48 ±0.004	0.50 ±0.009	0.68 ±0.004
Logits-max	0.59 ±0.005	0.56 ±0.005	0.46 ±0.004	0.55 ±0.013	0.56 ±0.006
Logits-max-exact	0.52 ±0.006	0.48 ±0.002	0.48 ±0.003	0.49 ±0.009	0.63 ±0.008
Probas-mean	0.61 ±0.002	0.56 ±0.010	0.57 ±0.007	0.58 ±0.007	0.65 ±0.007
Probas-mean-exact	0.68 ±0.008	0.65 ±0.006	0.51 ±0.006	0.57 ±0.009	0.67 ±0.003
Probas-min	0.60 ±0.004	0.51 ±0.007	0.59 ±0.007	0.55 ±0.005	0.64 ±0.008
Probas-min-exact	0.74 ±0.007	0.67 ±0.007	0.51 ±0.006	0.59 ±0.008	0.66 ±0.004
Probas-max	0.56 ±0.005	0.53 ±0.005	0.46 ±0.003	0.51 ±0.004	0.55 ±0.004
Probas-max-exact	0.49 ±0.007	0.47 ±0.002	0.51 ±0.005	0.50 ±0.009	0.62 ±0.006
p(True)	0.55 ±0.005	0.55 ±0.008	0.47 ±0.002	0.54 ±0.006	0.71 ±0.003
p(True)-exact	0.55 ±0.004	0.50 ±0.005	0.50 ±0.008	0.50 ±0.003	0.67 ±0.007
Probe @ token					
Last generated [-1]	0.77 ±0.005	0.68 ±0.006	0.69 ±0.006	0.78 ±0.005	0.77 ±0.009
Before last generated [-2]	0.76 ±0.002	0.69 ±0.005	0.67 ±0.008	0.79 ±0.004	0.75 ±0.007
End of question	0.78 ±0.004	0.60 ±0.003	0.65 ±0.004	0.74 ±0.002	0.75 ±0.011
Exact answer last	**0.83** ±0.005	**0.76** ±0.003	**0.78** ±0.007	**0.91** ±0.005	**0.78** ±0.006
Exact answer last+1	0.83 ±0.002	0.76 ±0.006	0.70 ±0.006	0.90 ±0.004	0.78 ±0.007



(a) Raw AUC values. Values above 0.5 indicate some generalization.

(b) Performance (AUC) difference of the probe and the logit-based method. Values above 0 indicate generalization beyond the logit-based method.

Figure 7: Generalization between datasets, Mistral-7b.



(a) Raw AUC values. Values above 0.5 indicate some generalization.

(b) Performance (AUC) difference of the probe and the logit-based method. Values above 0 indicate generalization beyond the logit-based method.

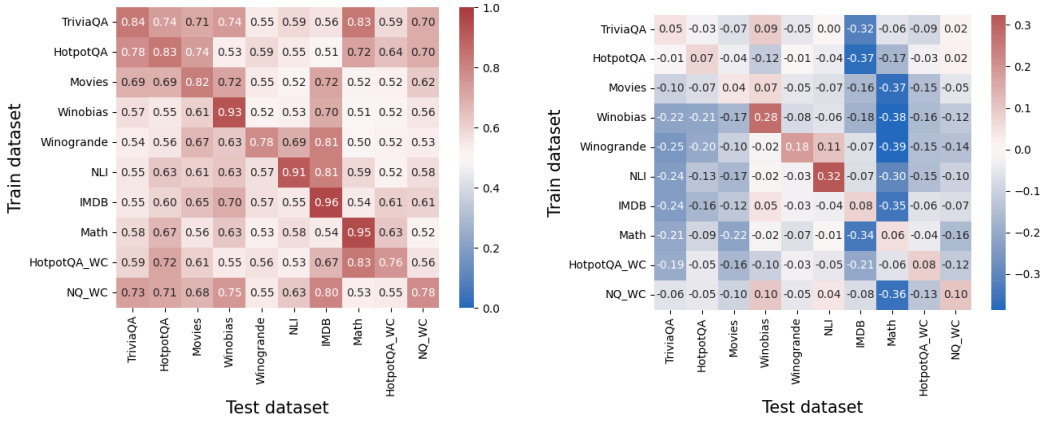
Figure 8: Generalization between datasets, Llama-3-8b.

C Full Generalization Results

Figures 7, 8 and 9 present the generalization results for the remaining models. While these results exhibit similar high-level patterns to those found in the main paper on Mistral-7b-instruct, notable differences suggest that these models may possess different mechanisms for encoding truthfulness.

D Taxonomy of Errors

Figure 10 presents, for each amount of resamples, the amount percentage of answers for which at least one generated answer was correct. The experiment was done on Mistral-7b-instruct with the TriviaQA dataset. For many answers that the greedy decoding fails to correctly provide an answer, the LLM is still able to generate the correct answer in at least one resample. The plot plateues around 30 resamples.



(a) Raw AUC values. Values above 0.5 indicate some generalization.

(b) Performance (AUC) difference of the probe and the logit-based method. Values above 0 indicate generalization beyond the logit-based method.

Figure 9: Generalization between datasets, Llama-3-8b-instruct.

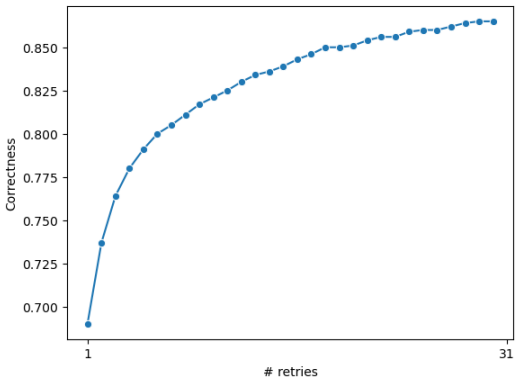


Figure 10: The percentage of answers for which at least one generated answer was correct. The first step is greedy decoding.

E DETECTING THE CORRECT ANSWER FULL RESULTS

In Table 8 we present some qualitative samples from Mistral-7b-instruct, for the phenomenon we observe at error type **(C2)** Consistently incorrect but generates the correct answer at least one time. The samples in the table represent cases where the probe chose the correct answer. Table 9 compares different decoding mechanisms, including the choice via probe, on non-instruct models, and Table 10 compares on the instruct models. For all datasets and models, we observe similar conclusions to those in the main paper: significant improvement is observed for error types where the LLM shows no preference to the correct answer.

Table 8: Examples of questions where Mistral-7b-Instruct consistently provided incorrect answers but occasionally generated the correct one. In these instances, the probe successfully identified the right answer. For each question, the model was samples 30 times.

Question	**Wrong Answer**	**Count**	**Correct Answer**	**Count**
Which town in southeast Wales became a UNESCO World Heritage Site in 2000?	Caerleon	29	Blaenavon	1
From her first US film musical "Down Argentina Way" (1940), who became famous for extravagant hats, jewellery and dresses?	Betty Grable	27	Carmen Miranda	1
Men Against the Sea and Pitcairn's Island were two sequels to what famous novel?	Robinson Crusoe	18	Mutiny on the Bounty	2
Which is the only property on a traditional UK Monopoly board which s south of the River Thames?	Coventry Street	17	Old Kent Road	3
Which French Canadian became Prime Minister of Canada in 1968?	Jean Chrétien	21	Pierre Elliott Trudeau	4

F PRACTICAL GUIDANCE ON INTEGRATING INSIGHTS FROM THIS PAPER INTO MODEL DEVELOPMENT WORKFLOWS

The findings of this study reveal critical insights into the internal mechanisms of Large Language Models (LLMs) and their implications for truthfulness and error handling. To effectively incorporate these insights into model development, consider the following strategies:

Error Detection. Focus on representations of exact answer tokens to train the error detection probe. These tokens encode significant truthfulness signals and improve the reliability of error detection mechanisms. The trained probe should be integrated as part of the pipeline for specific task, e.g., math calculations. The probe provides a confidence score which can be used to warn the user for unreliable outputs, to add a step of RAG for getting a better answer or to regenerate the answer.

Task-Specific Error Handling. The taxonomy of errors outlined in this study can be utilized to classify and analyze the types of errors that an LLMs may produce. The probes for detecting error types can be deployed as part of the LLM pipeline and create interventions based on their predictions. For example, RAG can help for consistently incorrect errors, as well as resampling and choosing the answer ranked highest by the error detection probe. For consistently correct,

Table 9: Various answer choice strategies, non-instruct models.

	Mistral-7b											
	TriviaQA				Math				Winobias			
Error type	Greedy	Random	Majority	Probing	Greedy	Random	Majority	Probing	Greedy	Random	Majority	Probing
All	0.63 ±0.003	0.54 ±0.004	0.65 ±0.002	0.62 ±0.003	0.25 ±0.018	0.36 ±0.022	0.49 ±0.019	0.60 ±0.017	0.69 ±0.016	0.58 ±0.009	0.62 ±0.009	0.83 ±0.006
(A) Refuses to answer	0.08 ±0.015	0.04 ±0.009	0.00 ±0.000	0.13 ±0.007	0.01 ±0.009	0.04 ±0.019	0.00 ±0.000	0.22 ±0.033	-	-	-	-
(B1) All	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000	-	-	-	-	-	-	-	-
(B2) Most	0.98 ±0.001	0.84 ±0.009	1.00 ±0.000	0.91 ±0.002	0.96 ±0.024	0.84 ±0.031	1.00 ±0.000	0.86 ±0.041	0.96 ±0.004	0.73 ±0.009	0.95 ±0.003	0.91 ±0.009
(C) Consistently incorrect												
(C1) All	0.00 ±0.003	0.00 ±0.000	0.00 ±0.000	0.00 ±0.000	-	-	-	-	-	-	-	-
(C2) Most	0.03 ±0.014	0.20 ±0.008	0.00 ±0.000	0.27 ±0.036	-	-	-	-	0.19 ±0.010	0.30 ±0.026	0.00 ±0.000	0.70 ±0.007
(D) Two competing	0.48 ±0.006	0.36 ±0.008	0.52 ±0.015	0.54 ±0.016	-	-	-	-	0.73 ±0.018	0.54 ±0.022	0.47 ±0.030	0.85 ±0.019
(E) Many answers												
(E1) Non correct	0.01 ±0.004	0.00 ±0.000	0.00 ±0.000	0.00 ±0.000	0.01 ±0.010	0.00 ±0.000	0.00 ±0.000	0.00 ±0.000	-	-	-	-
(E2) Correct appears	0.38 ±0.009	0.21 ±0.006	0.42 ±0.015	0.38 ±0.009	0.09 ±0.010	0.17 ±0.034	0.36 ±0.020	0.62 ±0.035	-	-	-	-
	Llama-8b											
	TriviaQA				**Math**				**Winobias**			
Error type	Greedy	Sampling	Majority	Probing	Greedy	Sampling	Majority	Probing	Greedy	Sampling	Majority	Probing
All	0.66 ±0.002	0.58 ±0.003	**0.68** ±0.003	**0.68** ±0.002	0.30 ±0.023	0.47 ±0.022	0.62 ±0.014	0.70 ±0.021	0.73 ±0.011	0.61 ±0.005	0.66 ±0.016	**0.84** ±0.006
(A) Refuses to answer	0.08 ±0.005	0.07 ±0.011	0.00 ±0.000	**0.16** ±0.011	0.00 ±0.007	0.04 ±0.015	0.00 ±0.000	0.25 ±0.025	-	-	-	-
(B) Consistently correct												
(B1) All	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000	-	-	-	-	-	-	-	-
(B2) Most	0.98 ±0.001	0.87 ±0.002	**1.00** ±0.000	0.95 ±0.002	0.77 ±0.024	0.88 ±0.025	1.00 ±0.000	0.97 ±0.014	0.98 ±0.005	0.75 ±0.004	**1.00** ±0.000	0.94 ±0.003
(C) Consistently incorrect												
(C1) All	0.00 ±0.000	0.00 ±0.000	0.00 ±0.000	0.00 ±0.000	-	-	-	-	-	-	-	-
(C2) Most	0.06 ±0.013	0.18 ±0.009	0.00 ±0.000	**0.35** ±0.043	-	-	-	-	0.25 ±0.026	0.29 ±0.023	0.00 ±0.000	**0.65** ±0.022
(D) Two competing	0.44 ±0.029	0.42 ±0.035	0.53 ±0.020	**0.66** ±0.030	-	-	-	-	0.73 ±0.025	0.47 ±0.019	0.41 ±0.037	**0.86** ±0.014
(E) Many answers												
(E1) Non correct	0.00 ±0.000	0.00 ±0.000	0.00 ±0.000	0.00 ±0.000	0.00 ±0.000	0.00 ±0.000	0.00 ±0.000	0.00 ±0.000	-	-	-	-
(E2) Correct appears	0.46 ±0.009	0.34 ±0.009	0.53 ±0.007	**0.54** ±0.005	0.14 ±0.015	0.17 ±0.025	0.44 ±0.047	0.65 ±0.031	-	-	-	-

Table 10: Various answer choice strategies, instruct models.

	Mistral-7b-Instruct											
	TriviaQA				Math				Winobias			
Error type	Greedy	Random	Majority	Probing	Greedy	Random	Majority	Probing	Greedy	Random	Majority	Probing
All	0.63 ±0.003	0.64 ±0.002	0.67 ±0.004	0.71 ±0.003	0.55 ±0.021	0.52 ±0.019	0.57 ±0.025	0.70 ±0.014	0.77 ±0.012	0.77 ±0.008	0.77 ±0.010	0.79 ±0.008
(A) Refuses to answer	0.06 ±0.005	0.06 ±0.011	0.00 ±0.000	0.28 ±0.009	-	-	-	-	-	-	-	-
(B1) All	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000
(B2) Most	0.88 ±0.007	0.83 ±0.009	0.99 ±0.002	0.89 ±0.010	0.87 ±0.013	0.84 ±0.024	1.00 ±0.000	0.96 ±0.007	0.91 ±0.031	0.87 ±0.029	0.96 ±0.017	0.89 ±0.032
(C) Consistently incorrect												
(C1) All	0.00 ±0.003	0.00 ±0.000	0.00 ±0.000	0.00 ±0.000	0.05 ±0.020	0.00 ±0.000	0.00 ±0.000	0.00 ±0.000	0.00 ±0.000	0.00 ±0.000	0.00 ±0.000	0.00 ±0.000
(C2) Most	0.11 ±0.009	0.15 ±0.012	0.00 ±0.000	0.53 ±0.005	0.10 ±0.040	0.20 ±0.050	0.00 ±0.000	0.82 ±0.037	0.18 ±0.057	0.20 ±0.039	0.00 ±0.000	0.54 ±0.067
(D) Two competing	0.32 ±0.010	0.45 ±0.023	0.50 ±0.024	0.78 ±0.017	-	-	-	-	-	-	-	-
(E) Many answers												
(E1) Non correct	0.01 ±0.003	0.00 ±0.000	0.00 ±0.000	0.00 ±0.000	-	-	-	-	-	-	-	-
(E2) Correct appears	0.23 ±0.020	0.19 ±0.022	0.38 ±0.009	0.56 ±0.025	-	-	-	-	-	-	-	-
	Llama-8b-Instruct											
	TriviaQA				**Math**				**Winobias**			
Error type	Greedy	Sampling	Majority	Probing	Greedy	Sampling	Majority	Probing	Greedy	Sampling	Majority	Probing
All	0.69 ±0.003	0.67 ±0.001	0.71 ±0.002	**0.73** ±0.004	0.89 ±0.010	0.87 ±0.012	**0.91** ±0.013	**0.91** ±0.010	0.75 ±0.009	0.74 ±0.009	0.76 ±0.012	**0.83** ±0.009
(A) Refuses to answer	0.06 ±0.011	0.05 ±0.011	0.00 ±0.000	**0.27** ±0.025	-	-	-	-	-	-	-	-
(B) Consistently correct												
(B1) All	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000	1.00 ±0.000
(B2) Most	0.93 ±0.002	0.86 ±0.009	**1.00** ±0.001	0.92 ±0.004	0.94 ±0.014	0.92 ±0.014	**1.00** ±0.000	0.95 ±0.013	0.94 ±0.006	0.88 ±0.010	1.00 ±0.000	0.93 ±0.011
(C) Consistently incorrect												
(C1) All	0.00 ±0.001	0.00 ±0.000	0.00 ±0.000	0.00 ±0.000	-	-	-	-	0.00 ±0.000	0.00 ±0.000	0.00 ±0.000	0.00 ±0.000
(C2) Most	0.12 ±0.018	0.22 ±0.010	0.00 ±0.000	**0.43** ±0.010	-	-	-	-	0.11 ±0.018	0.15 ±0.025	0.00 ±0.000	**0.67** ±0.016
(D) Two competing	0.43 ±0.017	0.42 ±0.014	0.46 ±0.016	**0.60** ±0.010	-	-	-	-	0.39 ±0.068	0.39 ±0.047	0.38 ±0.042	**0.83** ±0.050
(E) Many answers												
(E1) Non correct	0.00 ±0.002	0.00 ±0.000	0.00 ±0.000	0.00 ±0.000	-	-	-	-	-	-	-	-
(E2) Correct appears	0.28 ±0.006	0.28 ±0.008	0.40 ±0.009	**0.52** ±0.009	-	-	-	-	-	-	-	-

an intervention on the LLM's internal representations can increase the confidence in generating a correct answer Simhi et al. (2024).

Cross-Task Generalization. Universal generalization of probing classifiers across unrelated tasks should be approached with caution. The results in this work show that probes are mainly useful for task-specific error detection.

G Error Taxonomy Design Choices

The error taxonomy proposed in this paper is intentionally non-orthogonal, as some errors may simultaneously belong to multiple categories. For instance, an error might fall under both "consistently incorrect" (e.g., the same incorrect answer appears at least 15 times) and "many different answers" (e.g., the remaining answers show over 10 distinct variants).

Our taxonomy is designed to capture such nuanced cases, as restricting classification to a single category would hinder the generalizability of insights. Instead, we aim to learn general properties across different error types, providing LLM providers with actionable insights into questions exhibiting overlapping error patterns.

To support this non-orthogonal framework, our probes function as one-to-many classifiers, enabling precise error analysis and tailored solutions.