Too Consistent to Detect: A Study of Self-Consistent Errors in LLMs

Anonymous ACL submission

Abstract

As large language models (LLMs) often generate plausible but incorrect content, error detection has become increasingly critical to ensure truthfulness. However, existing detection methods often overlook a critical problem we term as **self-consistent error**, where LLMs repeatly generate the *same* incorrect response across multiple stochastic samples. This work formally defines self-consistent errors and evaluates mainstream detection methods on them. Our investigation reveals two key findings: (1) Unlike inconsistent errors, whose frequency diminishes significantly as LLM scale increases, the frequency of self-consistent errors remains stable or even increases. (2) All four types of detection methshods significantly struggle to detect self-consistent errors. These findings reveal critical limitations in current detection methods and underscore the need for improved methods. Motivated by the observation that self-consistent errors often differ across LLMs, we propose a simple but effective *cross-model probe* method that fuses hidden state evidence from an external verifier LLM. Our method significantly enhances performance on self-consistent errors across three LLM families[1].

1 Introduction

As large language models (LLMs) are increasingly deployed in high-stakes applications (Chen et al., 2024), their tendency to generate plausible yet incorrect content raises critical safety concerns. Therefore, error detection has become essential for ensuring the trustworthiness of LLMs (Manakul et al., 2023; Lin et al., 2024; Farquhar et al., 2024). Numerous error detection methods rely on measuring consistency across multiple samples (Manakul et al., 2023; Lin et al., 2024; Kuhn et al., 2023; Chen et al.; Xue et al., 2025) under the assumption that consistent outputs are more likely to be correct.

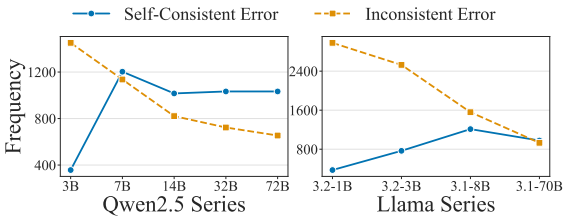

Figure 1: Frequency of self-consistent and inconsistent errors across different model scales on SciQ. Inconsistent errors decrease with model size while self-consistent errors remain stable or even slightly increase.

However, this assumption fails to account for a crucial phenomenon we define as "**self-consistent error**", where LLMs *consistently generate semantically equivalent errors across multiple stochastic samples for the same question*, in contrast to "**inconsistent error**" which vary between samples.

To demonstrate the importance of self-consistent errors, we analyze their frequency across the SciQ and TriviaQA datasets using nine model scales from the Qwen and Llama series. Figure 1 shows that the frequency of self-consistent errors remains stable or even increase with model scale, while inconsistent errors decrease significantly. This divergence highlights that self-consistent errors remain resistant to scaling, posing a persistent and long-term challenge. Therefore, detecting self-consistent errors becomes a critical research goal.

This paper systematically evaluates four types of mainstream error detectors on self-consistent errors, including probability methods (Duan et al., 2024), prompt-based (Kadavath et al., 2022; Tian et al.; Xiong et al.), supervised probe-based (Azaria and Mitchell, 2023; Beigi et al., 2024; Zhu et al., 2024), and consistency-based methods. We find that all methods suffer substantial performance drops on self-consistent errors, in contrast to their strong performance on inconsistent errors. Consistency-based detectors degrade the most, even falling below random guessing (AUROC

[1]Code and datasets will be released after review

≤ 0.5). Notably, even the strongest supervised probe that accesses the model's hidden states show significant performance drops, suggesting that the hidden states of an LLM alone cannot provide sufficient signal for detecting self-consistent error.

To improve detecting self-consistent errors, we propose a novel *cross-model probe* based on an observation: self-consistent errors tend to be model-specific and rarely overlap across different LLMs. Inspired by this, we feed the original model's response into an external verifier, extract its hidden states, and train a dedicated probe on them. This verifier-based probe are then integrated with the original probe to produce a unified detection score. This cross-model perspective compensates for the blind spots of the original model, enabling more reliable detection. Experiments across three LLM families and two datasets demonstrate that our method achieves substantial improvements in detecting self-consistent errors, offering a promising direction for future detection methods.

2 Self-Consistent Errors in LLMs

2.1 Task Definition

Error detection (Orgad et al., 2025; Farquhar et al., 2024), also called hallucination detection, seeks to decide whether an LLM's answer is factually correct. We use "error detection" due to the ambiguity of "hallucination" across domains (Wang Chaojun, 2020). Starting from a QA dataset $\mathcal{Q} = \{(q_i, a_i)\}_{i=1}^{N}$, where q_i is a question and a_i its reference answer, we obtain the model's greedy response $r_i^{\mathrm{g}} = \mathcal{M}(q_i; \theta, T = 0)$, with language model $\mathcal{M}$ (parameters θ) and temperature T. Current work primarily targets greedy responses as they reflect the model's best choice and facilitate reproducibility. We label each prediction by comparing it with a_i, yielding $z_i \in \{0, 1\}$ according to the procedure in Section 3.1. This produces the error detection datasets $\mathcal{D}_{\mathcal{M}} = \{(q_i, r_i^{\mathrm{g}}, z_i)\}_{i=1}^{N}$. At test time, the detector observes only (q_i, r_i^{g}) and predict the error score $s_i = f(q_i, r_i^{\mathrm{g}})$.

2.2 Definition of Self-Consistent Error

We categorize errors as *self-consistent* if the model repeatedly generates semantically equivalent incorrect responses across multiple stochastic samples for a given question, and as *inconsistent* otherwise.

Definition 1 (Self-Consistent Error). *For a question q_i, we draw k stochastic samples*

$$r_{i,j}^{\mathrm{s}} = \mathcal{M}(q_i; \theta, T > 0, j), \qquad j = 1, \ldots, k.$$

If all samples are semantically equivalent to the greedy response,

$$r_{i,1}^{\mathrm{s}} \equiv r_{i,2}^{\mathrm{s}} \equiv \cdots \equiv r_{i,k}^{\mathrm{s}} \equiv r_i^{\mathrm{g}},$$

and the greedy answer is judged incorrect ($z_i = 0$), then r_i^{g} is a self-consistent error *for model $\mathcal{M}$. The relation $\equiv$ denotes semantic equivalence.*

To operationalize Definition 1 and categorize errors in $\mathcal{D} = \{(q_i, r_i^{\mathrm{g}}, z_i)\}_{i=1}^{M}$, we proceed as follows. For every incorrect instance ($z_i = 0$), we generate $k = 15$ stochastic samples $r_{i,1}^{\mathrm{s}}, \ldots, r_{i,15}^{\mathrm{s}}$ in addition to the greedy answer r_i^{g}. Sampling is performed with temperature T=0.5, `top_p`=1 and `top_k`=-1, which is the commonly adopted settings in prior work (Kuhn et al., 2023). Next, we test pairwise semantic equivalence within $\{r_i^{\mathrm{g}}, r_{i,1}^{\mathrm{s}}, \ldots, r_{i,15}^{\mathrm{s}}\}$ with the NLI-based criterion of Kuhn et al. (2023), treating two responses as equivalent if they mutually entail each other. An error r_i^{g} is labeled *self-consistent* when all stochastic samples and greedy response are semantically equivalent; otherwise, r_i^{g} is labeled *inconsistent*.

2.3 Why Self-Consistent Errors Matter?

We investigate the prevalence of self-consistent errors across different model scales, including Qwen (Qwen2.5-3/7/14/32/72B-Instruct) and Llama (Llama3.2-1B/3B, 3.1-8/70B-Instruct)[2]. We use TriviaQA (TQA for short) (Joshi et al., 2017) and SciQ (Welbl et al., 2017) datasets, which represent trivia and scientific knowledge domains, respectively. Figure 1 shows how the frequency of errors changes with model scale on SciQ, with TQA shown in Appendix 2. Unlike inconsistent errors, which markedly decrease as models scale up, the number of self-consistent errors remains relatively stable, or even slightly increases. This suggests that self-consistent errors, being more resistant to model scaling, will likely remain a persistent challenge, potentially becoming more concerning as LLMs continue to scale. Therefore, analyzing and improving the capability to detect this class of errors becomes increasingly crucial.

Besides their prevalence, self-consistent errors are potentially more challenging to detect. The methods leveraging sample consistency implicitly equate consistency with correctness, thereby inherently failing to detect these self-consistent errors. The effectiveness of other methods on them

[2]As this work focuses on text-only models, we exclude vision LLMs (Llama3.2-11B/90B).

Method	Llama3.1-8b						Qwen2.5-7b					
	SciQ-CE	SciQ-IE	Δ ↓	TQA-CE	TQA-IE	Δ ↓	SciQ-CE	SciQ-IE	Δ ↓	TQA-CE	TQA-IE	Δ ↓
Probability	0.6325	0.8192	0.1867	0.6243	0.8455	0.2212	0.4571	0.6594	0.2023	0.5360	0.7148	0.1788
P(True)	0.6251	0.7625	0.1374	0.6836	0.8018	0.1182	0.6158	0.7589	0.1431	0.7478	0.8373	0.0895
SE	0.4608	0.8820	0.4212	0.5216	0.9226	0.4010	0.4782	0.8247	0.3465	0.4453	0.9119	0.4666
Probe (OOD)	0.7287	0.908	0.1793	0.7396	0.8989	0.1593	0.7487	0.8605	0.1118	0.7734	0.8911	0.1177
+ cross-model	0.8289	0.9385	0.1096	0.8024	0.9263	0.1239	0.8211	0.8893	0.0682	0.8691	0.9457	0.0766
Probe (ID)	0.7917	0.9249	0.1332	0.7922	0.9272	0.1350	0.8250	0.8891	**0.0641**	0.8626	0.9467	0.0841
+ cross-model	**0.8659**	**0.9408**	**0.0749**	**0.8470**	**0.9477**	**0.1007**	**0.8399**	**0.9078**	0.0679	**0.9088**	**0.9696**	**0.0608**

Method	Qwen2.5-14b						Mistral-12b					
	SciQ-CE	SciQ-IE	Δ ↓	TQA-CE	TQA-IE	Δ ↓	SciQ-CE	SciQ-IE	Δ ↓	TQA-CE	TQA-IE	Δ ↓
Probability	0.5480	0.7517	0.2037	0.4926	0.6477	0.1551	0.5858	0.7354	0.1496	0.6283	0.8605	0.2322
P(True)	0.5287	0.6744	0.1457	0.7052	0.8515	0.1463	0.6595	0.7625	0.1030	0.7502	0.8545	0.1043
SE	0.5427	0.8764	0.3337	0.4425	0.9074	0.4649	0.3633	0.8210	0.4677	0.4494	0.9093	0.4599
Probe (OOD)	0.7425	0.9025	0.1600	0.7871	0.9174	0.1303	0.7767	0.8553	0.0786	0.6927	0.8577	0.1650
+ cross-model	0.7927	0.9263	0.1336	0.8754	0.9115	**0.0361**	0.8458	0.9276	0.0818	0.7872	0.9069	0.1197
Probe (ID)	0.7473	0.8582	0.1109	0.8512	0.9570	0.1058	0.7726	0.8652	0.0926	0.8163	0.9063	0.0900
+ cross-model	**0.8118**	**0.8931**	**0.0813**	**0.9332**	**0.9776**	0.0444	**0.8548**	**0.9253**	**0.0705**	**0.8497**	**0.9359**	**0.0862**

Table 1: AUROC performance of error detection methods. Δ is the performance gap between CE and IE subsets.

may also be limited. For instance, probability-based methods assume that the errors have lower sequence probabilities, which may not hold for self-consistent errors, as such consistent responses intuitively exhibit higher probabilities. Therefore, we begin by systematically evaluating the performance of existing methods on self-consistent errors.

3 How Well Do We Detect Self-Consistent Errors?

This section evaluates the performance of current error detection methods on self-consistent errors.

3.1 Experiment Setup

To ensure a fair comparison between two types of errors for supervised probe methods, we controlled the distribution of the dataset. We created specialized subsets for the two types of errors: **(i) CE subset**, containing only self-consistent errors as negative (incorrect) examples, and **(ii) IE subset**, containing only inconsistent errors as negative examples. Both subsets contain an identical number of negative examples and are paired with the same number of positive examples for training. This setup controls for the influence of training data volume on supervised probe. The performance gap Δ between these two subsets reveals the different detection difficulty between two types of errors.

Evaluation Metric. Following prior works (Kuhn et al., 2023; Xiong et al.; Duan et al., 2024), we evaluate error detection using the area under the receiver operator characteristic curve (**AUROC**). We produce the correctness label z_i by employing an LLM to evaluate whether the response is semantically equivalent to the ground truth answer, following (Tian et al.; Wei et al., 2024). Details are provided in Appendix A.5.

Baseline & LLMs. We evaluate four types of mainstream error detection methods on commonly used LLMs: Qwen2.5-7b/14b (Yang et al., 2024), Llama3.1-8b, and Mistral-12b. Training-free baselines include: **(1) Probability** uses aggregated token probabilities (Orgad et al., 2025; Mahaut et al., 2024; Malinin and Gales, 2021). **(2) P(True)** prompts LLM to self-critique correctness and uses the probability of "True" as the confidence score (Kadavath et al., 2022). **(3) SE** (Kuhn et al., 2023; Farquhar et al., 2024) samples multiple responses and calculates the entropy of their semantic clusters. Supervised baselines include: **(4) Probe** which trains a simple feedforward neural network to detect error based on the hidden states of LLMs (Azaria and Mitchell, 2023). We use the hidden states of the last token at the layer with the best validation performance. We distinguish **Probe (ID)** (trained and evaluated on the same dataset) from **Probe (OOD)** (trained on one dataset, evaluated on another). For instance, Probe-OOD might be trained on the SciQ-CE before being evaluated on TQA-CE. OOD evaluation is critical to ensure the probe captures truthfulness features, rather than overfitting to a single dataset (Orgad et al., 2025). Further details are in Appendix A.3.

3.2 Failures in Self-Consistent Errors

As shown in Table 1, existing methods perform well on inconsistent errors (AUROC up to about 90%). However, all methods suffer a substantial

performance degradation on consistent errors. SE which performs best among training-free methods on IE subsets, exhibits the most dramatic decline on CE subsets, performing at or below random guessing. This challenges the assumption that self-consistency implies correctness, revealing critical limitations in consistency-based detection methods. Although supervised methods generally outperform training-free approaches on CE subsets, they still show significant performance degradation compared to IE subsets. This indicates that self-consistent errors are more challenging to distinguish from correct responses even at the hidden state level. Furthermore, Probe (OOD) shows larger performance gaps (Δ) compared to Probe (ID), suggesting that self-consistent errors are particularly difficult to detect when generalizing across different knowledge domains TQA and SciQ.

4 Cross-Model Probe

The poor performance of the evaluated methods on self-consistent errors suggests that features from the response-generating LLM alone may be insufficient for detecting such errors. Fortunately, we observe that self-consistent errors are often model-specific and rarely overlap across different LLMs. For instance, among questions where Qwen2.5-14B produces self-consistent errors, only 9.6% of them lead Llama3.1-70B to consistently make the same errors. This observation motivates the use of an external verifier to supplement the detection of self-consistent errors.

Given the high efficiency (Su et al., 2024) and strong performance of supervised probes, we build upon this approach. Standard probe methods train a classifier to detect errors using internal states of $\mathcal{M}$ which generate the response r_i^{g}:

$$s_i^{\mathcal{M}} = \mathrm{Probe}_{\mathcal{M}}(\mathbf{h}_i^{\mathcal{M}}), \quad \mathbf{h}_i^{\mathcal{M}} = \phi_{\mathcal{M}}^{(l,t)}\left(q_i, r_i^{\mathrm{g}}\right)$$

where $\phi_{\mathcal{M}}^{(l,t)}$ extracts internal states from layer l and token position t of model $\mathcal{M}$. We introduce a cross-model probe that leverages an external verifier LLM $\mathcal{V}$ to embed the responses generated by $\mathcal{M}$ and trains a separate $\mathrm{Probe}_{\mathcal{V}}$:

$$s_i^{\mathcal{V}} = \mathrm{Probe}_{\mathcal{V}}(\mathbf{h}_i^{\mathcal{V}}), \quad \mathbf{h}_i^{\mathcal{V}} = \phi_{\mathcal{V}}^{(l,t)}\left(q_i, r_i^{\mathrm{g}}\right)$$

The final error score combines both probes through an integration parameter λ:

$$\mathrm{score}_i = (1-\lambda) \cdot s_i^{\mathcal{M}} + \lambda \cdot s_i^{\mathcal{V}}$$

Verifier	Different Series	Scale	λ	AUROC		
				Res only	Ver only	Fused
Qwen2.5-3b	✗	smaller	0.25	0.8250	0.8129	0.8357
Llama3.2-3b	✓	smaller	0.50	0.8250	0.8125	0.8453
Llama3.1-70b	✓	larger	0.85	0.8250	0.8740	**0.8794**
Qwen2.5-72b	✗	larger	1.00	0.8250	0.8689	0.8689

Table 2: Effect of using different verifier LLMs against responses generated by qwen2.5-7b on the SciQ-CE.

In our implementation, we select Qwen2.5-14B as the verifier for all other models except itself, for which we use Llama3.1-70b. λ is selected from $\{0, 0.05, 0.1, \ldots, 1.0\}$ by choosing the value that yields the best validation performance. As shown in Table 1, cross-model probe demonstrates significant performance improvements on CE subsets, regardless of in-domain or out-of-domain settings.

We conduct an analysis of verifier selection across different model scale and series, detailed in Appendix A.4. As shown in Table 2, all tested verifiers, including the 3B-scale models, consistently achieve substantial performance gains, validating the effectiveness of our approach. Besides, our empirical results suggest that using a *larger* verifier from a *different series* could achieve the most substantial improvement.

5 Related work

Zhang et al. (2023); Chen et al. also mention the limitation of consistency-based methods regarding self-consistent errors. Beyond these studies, we demonstrate the importance of self-consistent errors by analyzing their frequency, systematically quantify performance degradation across four mainstream detection methods (not only consistency-based), and propose a simple yet effective improvement. Appendix A.2 provides a more detailed discussion of related works.

6 Conclusion

This work investigates self-consistent errors where the LLM repeats the same incorrect response across multiple stochastic samples. Our analysis shows that the frequency of self-consistent errors persist or even increase with increasing model scale, highlighting the importance of detecting them in ever-larger LLMs. Then, we evaluate four representative error detection methods and find all of them expose clear limitations in self-consistent errors. Finally, we introduce a simple but effective cross-model probe to improve detection performance on self-consistent errors.

7 Limitations

The underlying causes of consistent errors still require deeper investigation. These systematic failures may stem from prevalent misconceptions in training data, or biases introduced during the supervised training phase. Future works may construct controlled experiments to investigate the causes.

8 Ethics Statement

Data All data used in this study are publicly available and do not pose any privacy concerns.

AI Writing Assistance In our study, we only employed ChatGPT to polish our textual expressions rather than to generate new ideas or suggestions.

References

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

Mohammad Beigi, Ying Shen, Runing Yang, Zihao Lin, Qifan Wang, Ankith Mohan, Jianfeng He, Ming Jin, Chang-Tien Lu, and Lifu Huang. 2024. InternalInspector i^2: Robust confidence estimation in LLMs through internal states. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12847–12865, Miami, Florida, USA. Association for Computational Linguistics.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2023. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations*.

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*.

Zhiyu Chen, Jing Ma, Xinlu Zhang, Nan Hao, An Yan, Armineh Nourbakhsh, Xianjun Yang, Julian McAuley, Linda Ruth Petzold, and William Yang Wang. 2024. A survey on large language models for critical societal domains: Finance, healthcare, and law. *Transactions on Machine Learning Research*. Survey Certification.

Hanxing Ding, Liang Pang, Zihao Wei, Huawei Shen, and Xueqi Cheng. 2024. Retrieve only when it needs: Adaptive retrieval augmentation for hallucination mitigation in large language models. *arXiv preprint arXiv:2402.10612*.

Xuefeng Du, Chaowei Xiao, and Yixuan Li. 2024. Haloscope: Harnessing unlabeled LLM generations for hallucination detection. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2024. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5050–5063.

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

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2).

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, and 17 others. 2022. Language models (mostly) know what they know. *Preprint*, arXiv:2207.05221.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Stephanie C. Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *Trans. Mach. Learn. Res.*, 2022.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024. Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*.

Matéo Mahaut, Laura Aina, Paula Czarnowska, Momchil Hardalov, Thomas Müller, and Lluis Marquez.

2024. Factual confidence of LLMs: on reliability and robustness of current estimators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4554–4570, Bangkok, Thailand. Association for Computational Linguistics.

Andrey Malinin and Mark Gales. 2021. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*.

Potsawee Manakul, Adian Liusie, Mark JF Poon, Yun-Sung Chuang, and Philip HS Torr. 2023. Selfcheck-gpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1516–1540.

Samuel Marks and Max Tegmark. 2024. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. 2025. LLMs know more than they show: On the intrinsic representation of LLM hallucinations. In *The Thirteenth International Conference on Learning Representations*.

Weihang Su, Changyue Wang, Qingyao Ai, Yiran Hu, Zhijing Wu, Yujia Zhou, and Yiqun Liu. 2024. Unsupervised real-time hallucination detection based on the internal states of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14379–14391, Bangkok, Thailand. Association for Computational Linguistics.

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. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Sennrich Rico Wang Chaojun. 2020. On exposure bias, hallucination and domain shift in neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3544–3552, Online. Association for Computational Linguistics.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*.

Johannes Welbl, Nelson F Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *The Twelfth International Conference on Learning Representations*.

Yihao Xue, Kristjan Greenewald, Youssef Mroueh, and Baharan Mirzasoleiman. 2025. Verify when uncertain: Beyond self-consistency in black box hallucination detection. *arXiv preprint arXiv:2502.15845*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Jiaxin Zhang, Zhuohang Li, Kamalika Das, Bradley Malin, and Sricharan Kumar. 2023. SAC^3: Reliable hallucination detection in black-box language models via semantic-aware cross-check consistency. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

Derui Zhu, Dingfan Chen, Qing Li, Zongxiong Chen, Lei Ma, Jens Grossklags, and Mario Fritz. 2024. PoLLMgraph: Unraveling hallucinations in large language models via state transition dynamics. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4737–4751, Mexico City, Mexico. Association for Computational Linguistics.

A Appendix

A.1 The Number of Consistent and Inconsistent Errors

Figure 2 shows the number of consistent and inconsistent errors for different LLMs.

A.2 Related Work

Error Detection. Large language models (LLMs) often generate responses that appear plausible but contain factual inaccuracies. This challenge underscores the critical importance of accurately detecting errors in LLM-generated content for establishing trustworthiness. While this task is also referred to as "hallucination detection" (Chen et al.; Farquhar et al., 2024; Du et al., 2024), we adopt the

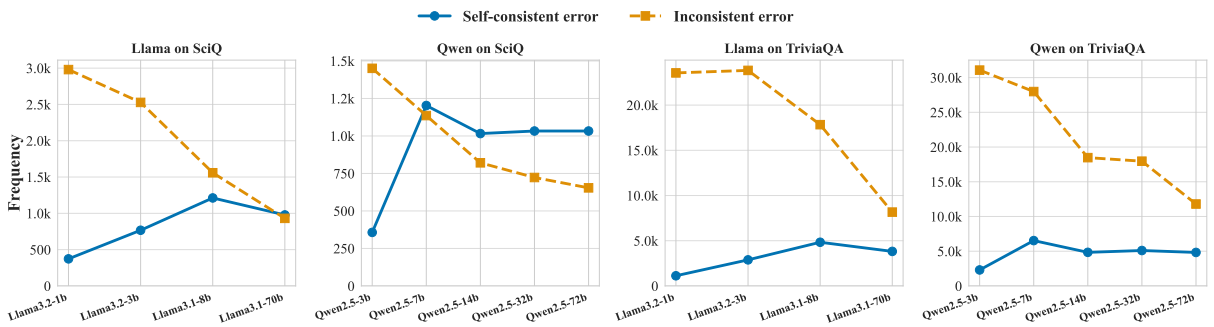

Figure 2: The number of self-consistent and inconsistent errors across different scales of LLMs.

term "error detection" to avoid ambiguity, as "hallucination" carries domain-specific meanings across different fields (Huang et al., 2025; Wang Chaojun, 2020; Zhang et al., 2019).

Training-Free Error Detection. A prominent approach to error detection involves estimating the uncertainty inherent in the model itself. Methods in this category include analyzing response probabilities (Malinin and Gales, 2021; Duan et al., 2024) and eliciting verbalized confidence scores directly from the model (Tian et al.; Lin et al., 2022; Xiong et al.). Among these methods, consistency-based uncertainty (Manakul et al., 2023; Kuhn et al., 2023; Lin et al., 2024; Xiong et al.; Chen et al.; Zhang et al., 2023; Chen et al.) has received considerable attention. Building on the assumption that consistent responses are more likely to be factually correct (Wang et al., 2023), consistency-based methods sample multiple responses and compute semantic consistency among them to detect hallucinations.

Supervised Probe. In contrast to the above methods, probe-based approaches employ supervised learning to identify truthfulness features embedded within LLMs' internal states. Several previous works (Marks and Tegmark, 2024; Azaria and Mitchell, 2023; Burns et al., 2023; Li et al., 2023; Chen et al.) have claimed that there existed truthfulness features in the internal states of LLMs. Based on the assumption, numerous studies have tried to detect hallucination using the features from LLMs' own internal states (Kadavath et al., 2022; Azaria and Mitchell, 2023; Beigi et al., 2024; Zhu et al., 2024). These works trained a probe, a simple classifier, to predict whether the response of LLMs is correct based on the internal states. As the probe is often a simple multi-layer perceptron, these methods need very low computation cost both during inference time and training process (Su et al., 2024). Moreover, recent comparative studies (Mahaut et al., 2024) have demonstrated their superior performance over other consistency-based, probability-based and verbalized methods.

Self-Consistent Error. Prior consistency-based error detectors (Farquhar et al., 2024; Zhang et al., 2023; Chen et al.) also acknowledged the limitations of consistency-based methods in handling self-consistent errors. However, they neither quantify the extent of performance degradation nor systematically examine the prevalence of such errors. Moreover, their analysis is limited to consistency-based paradigms, leaving open the question of whether other types of detectors are similarly affected. In contrast, our work provides a comprehensive evaluation across four mainstream categories of error detection methods and reveals that self-consistent errors pose a universal challenge, leading to significant performance drops across all methods, not just those relying on sample consistency.

Cross-Model Checking. Zhang et al. (2023); Ding et al. (2024) and concurrent work (Xue et al., 2025) propose to detect errors by sampling multiple responses from both the target model and an external model, followed by measuring their agreement. However, these approaches require 10–20 additional generations per query across both models, making them impractical for real-time usage. In contrast, our Cross-Model Probe offers a novel and efficient alternative that requires only a single forward pass through a verifier model. Furthermore, our empirical analysis provides practical insights for verifier selection. All tested verifiers, including the lightweight 3B models, consistently yield performance gains, demonstrating the robustness of our approach. Nonetheless, larger models from a different series than the response generator tend to perform best.

A.3 Baseline Method Implementation Details

Here we provide detailed implementation details for the baseline error detection methods evaluated in Section 3.1.

(i) Probability: Several studies have employed the aggregated token probabilities to detect errors (Orgad et al., 2025; Mahaut et al., 2024; Malinin and Gales, 2021). Following prior work (Orgad et al., 2025), we average the log-probabilities of all generated tokens in a response. This average log-probability serves as the error detection indicator, where lower values suggest a higher likelihood of error.

(ii) P(True): This method follows the prompting strategy introduced by Kadavath et al. (2022), where the LLM is directly queried to assess the correctness of its own output. Specifically, we construct the following prompt:

```
Question: {question}
Possible answer: {response}
Is the possible answer: A. True
B. False
The possible answer is:
```

The model's confidence is then quantified as the probability it assigns to the token sequence corresponding to "A". A higher probability indicates greater model confidence in the correctness of its response.

(iii) SE (Semantic Entropy): As proposed by Kuhn et al. (2023) and further explored by Farquhar et al. (2024), semantic entropy estimates uncertainty over the meaning conveyed by a response, rather than just the token sequence. Higher semantic entropy suggests greater uncertainty about the response's meaning and thus a higher likelihood of error. Following the implementation details recommended by Kuhn et al. (2023), we set the sampling parameters as follows: temperature 0.5, number of samples 10, top_p = 1.0, and top_k = -1.

(iv) Probe: Following Azaria and Mitchell (2023), we implement a probe using a three-layer feedforward neural network (FFN) with ReLU activations and hidden dimensions set to (256, 128, 64). The model is trained with cross-entropy loss. To select the most informative hidden layer, we train a separate probe on the output of each layer and choose the one that achieves the highest AUROC on the validation set. To mitigate overfitting, the probe is trained for a fixed number of epochs, and we select the checkpoint with the best validation performance for final evaluation.

A.4 Details about Cross-Model Probe

How to Select Verifier. We study the impact of different verifiers on cross-model probe performance, focusing on two factors: (1) whether the verifier is from the same model series as the response model, and (2) model scale. Using Qwen2.5-7B as the response model, we evaluate several verifiers: Qwen2.5-3B and LLaMA3.2-3B (smaller models); LLaMA3.1-70B and Qwen2.5-72B (larger models).

Table 2 shows that (1) for models of the same scale, using a verifier from a different series yields better results. (2) within the same series, larger verifiers perform better. Notably, all tested verifiers (even 3B models) significantly improve performance over the standard probe, validating the effectiveness of our approach.

A.5 Evaluation Metric

Following prior works (Kuhn et al., 2023; Xiong et al.; Duan et al., 2024), we evaluate error detection using the area under the receiver operator characteristic curve (**AUROC**), which reflects models' ability to distinguish incorrect and correct responses. We produce the correctness label z_i by employing an LLM to evaluate whether the response is semantically equivalent to the ground truth answer, following (Tian et al.; Wei et al., 2024). To ensure reproducibility, we employ the powerful open-source model, Llama-3.1-70b. Inspired by (Wei et al., 2024), we use the prompt in Appendix A.6 to check the correctness of the generated response. This prompt categorizes responses into correct, incorrect, and refusal. In our experiments, we filter out the refusal responses, as our focus is on effectively distinguishing between correct and incorrect responses. A manual review finds that only 1 out of 300 samples disagrees with human annotation, demonstrating the reliability of the correctness label.

A.6 Prompt

Evaluation Prompt

Your job is to look at a question, some gold targets, and a predicted answer, and then assign a grade of either ["CORRECT", "INCORRECT", "NOT_ATTEMPTED"].
First, I will give examples of each grade, and then you will grade a new example.

The following are examples of CORRECT predicted answers.

Question: What are the names of Barack Obama's children?
Gold target: ["Malia Obama and Sasha Obama", "malia and sasha"]
Predicted answer 1: sasha and malia obama
Predicted answer 2: most people would say Malia and Sasha, but I'm not sure and would have to double check
Predicted answer 3: Barack Obama has two daughters. Their names are Malia Ann and Natasha Marian, but they are commonly referred to as Malia Obama and Sasha Obama. Malia was born on July 4, 1998, and Sasha was born on June 10, 2001.

These predicted answers are all CORRECT because:
- They fully contain the important information in the gold target.
- They do not contain any information that contradicts the gold target.
- Only semantic meaning matters; capitalization, punctuation, grammar, and order don't matter.
- Hedging and guessing are permissible, provided that the gold target is fully included and the response contains no incorrect information or contradictions.

The following are examples of INCORRECT predicted answers.

Question: What are the names of Barack Obama's children?
Gold target: ["Malia and Sasha"]
Predicted answer 1: Malia.
Predicted answer 2: Malia, Sasha, and Susan.
Predicted answer 3: Barack Obama does not have any children.
Predicted answer 4: I think it's either Malia and Sasha. Or it could be Malia and Jackie. Or it could be Joey and Malia.
Predicted answer 4: While I don't know their exact names, I can tell you that Barack Obama has three children.
Predicted answer 5: It's possible you may mean Betsy and Olivia. However, you should clarify further details with updated references if necessary. Is that the correct answer?
Predicted answer 6: It may be the case that Obama's child is named James. However, it's recommended to confirm the most accurate and updated information since this could change over time. This model may not always reflect the most current information.

These predicted answers are all INCORRECT because:
- A factual statement in the answer contradicts the gold target. Incorrect statements that have some hedging (e.g., "it is possible that", "although i'm not sure, i think") are also considered incorrect.

The following are examples of NOT_ATTEMPTED predicted answers.

Question: What are the names of Barack Obama's children?
Gold target: ["Malia and Sasha"]
Predicted answer 1: I don't know.
Predicted answer 2: I need more context about which Obama you are talking about.
Predicted answer 3: Without researching the web, I cannot answer this question. However, I can tell you that Barack Obama has two children.
Predicted answer 4: Barack Obama has two children. I know that one of them is Malia, but I'm not sure about the other one.

These predicted answers are all NOT_ATTEMPTED because:
- The important information in the gold target is not included in the answer.
- No statements in the answer contradict the gold target.

Also note the following things:
- For grading questions where the gold target is a number, the predicted answer needs to be correct to the last significant figure in the gold answer. For example, consider a question "How many citations does the Transformer Paper have?" with gold target "120k".

687

- Predicted answers "120k", "124k", and 115k" are all CORRECT.
- Predicted answers "100k" and "113k" are INCORRECT.
- Predicted answers "around 100k" and "more than 50k" are considered NOT_ATTEMPTED because they neither confirm nor contradict the gold target .

- The gold target may contain more information than the question . In such cases , the predicted answer only needs to contain the information that is in the question .
  - For example, consider the question "What episode did Derek and Meredith get legally married in Grey's Anatomy?" with gold target "Season 7, Episode 20: White Wedding". Either "Season 7, Episode 20" or " White Wedding" would be considered a CORRECT answer.
- Do not punish predicted answers if they omit information that would be clearly inferred from the question .
  - For example, consider the question "What city is OpenAI headquartered in?" and the gold target "San Francisco , California ". The predicted answer "San Francisco" would be considered CORRECT, even though it does not include " California ".
  - Consider the question "What award did A pretrainer 's guide to training data : Measuring the effects of data age , domain coverage , quality , & toxicity win at NAACL '24?", the gold target is "Outstanding Paper Award". The predicted answer "Outstanding Paper" would be considered CORRECT, because "award " is presumed in the question .
  - For the question "What is the height of Jason Wei in meters ?", the gold target is "1.73 m". The predicted answer "1.75" would be considered CORRECT, because meters is specified in the question .
  - For the question "What is the name of Barack Obama's wife ?", the gold target is "Michelle Obama". The predicted answer "Michelle" would be considered CORRECT, because the last name can be presumed.
- Do not punish for typos in people 's name if it 's clearly the same name.
  - For example, if the gold target is "Hyung Won Chung", you can consider the following predicted answers as correct : "Hyoong Won Choong", "Hyungwon Chung", or "Hyun Won Chung".

Here is a new example. Simply reply with either CORRECT, INCORRECT, NOT_ATTEMPTED. Don't apologize or correct yourself if there was a mistake; we are just trying to grade the answer.

Question : { question }
Gold target : { target }
Predicted answer: { predicted_answer }

Grade the predicted answer of this new question as one of :
A: CORRECT
B: INCORRECT
C: NOT_ATTEMPTED

Just return the letters "A", "B", or "C", with no text around it .