amazon | science

Internal Representations as Indicators of Hallucinations in Agent Tool Selection

Kait Healy & Bharathi Srinivasan & Visakh Madathil & Jing Wu

Amazon Web Services, Seattle, WA, USA | {kaitheal, bhrsrini, visakhm, jingwua}@amazon.com

arXiv:2601.05214

Introduction & Motivation

The Agent Tool-Selection Hallucination Problem

Large language models (LLMs) increasingly power AI agents that decide how and when to call tools — invoking APIs, querying data sources, and completing complex tasks (Brown et al. 2020; Schick et al. 2023; Qin et al. 2024). However, these models exhibit **tool-calling hallucinations**, distinct from textual hallucinations:

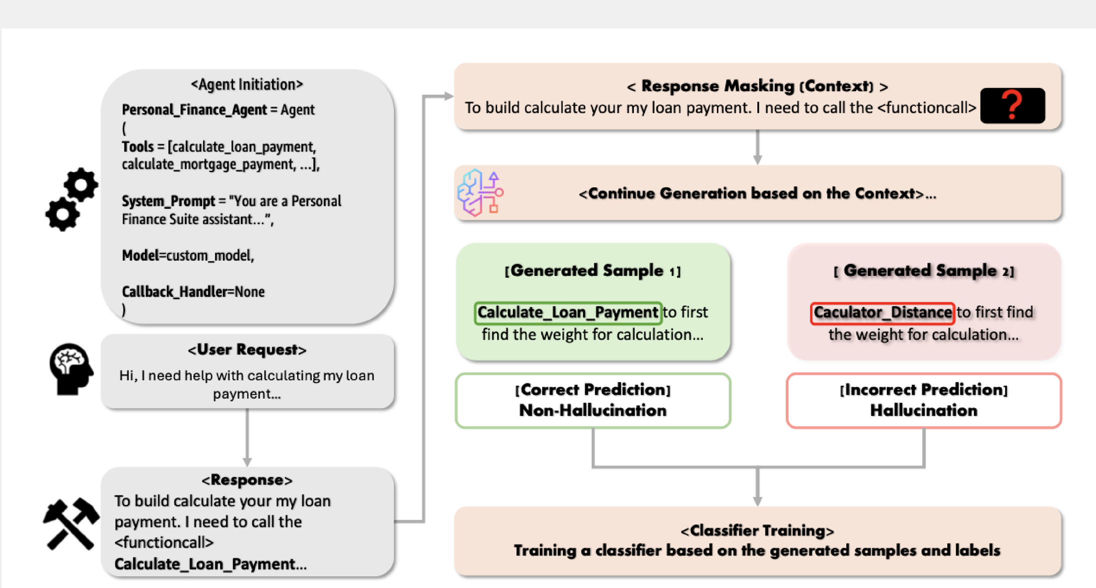
- **Function selection errors**: invoking non-existent functions
- **Appropriateness errors**: semantically inappropriate tool choice
- **Parameter errors**: missing required arguments or incorrect types
- **Tool bypass errors**: responding without using any tool

Research Question: How can we accurately detect hallucinations in AI agents in real-time with minimal additional computation?

Key Contributions

1. **Single-pass internal-state detection:** Final-layer internal representations enable real-time detection of tool-calling hallucinations without extra forward passes or external validators.
2. **Unsupervised hallucination labeling pipeline:** Masked-call regeneration plus function and argument canonicalization yields automatic labels without manual annotation.
3. **Lightweight, model-specific classifiers:** Simple MLPs over compact features achieve up to 86.4% accuracy across Qwen-7B, Llama-3.1-8B, and GPT-OSS-20B.

System Architecture



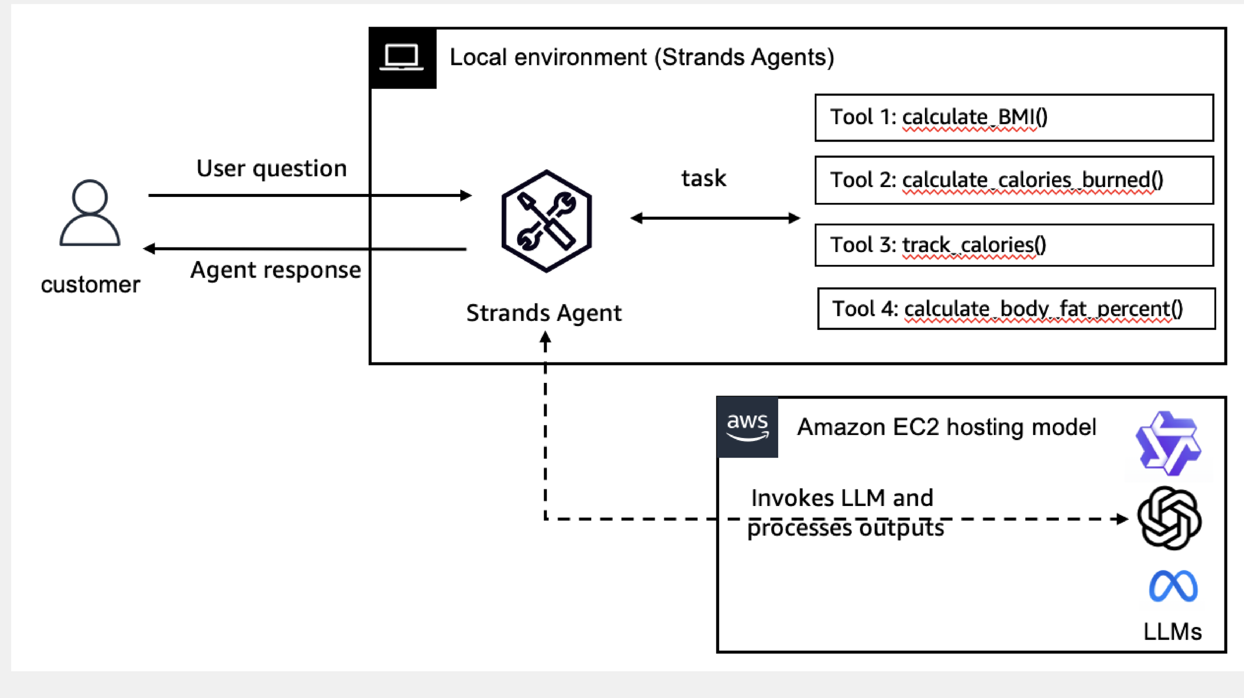
Key Innovation: Single-pass detection using internal states—no multiple samplings required

Algorithm Overview

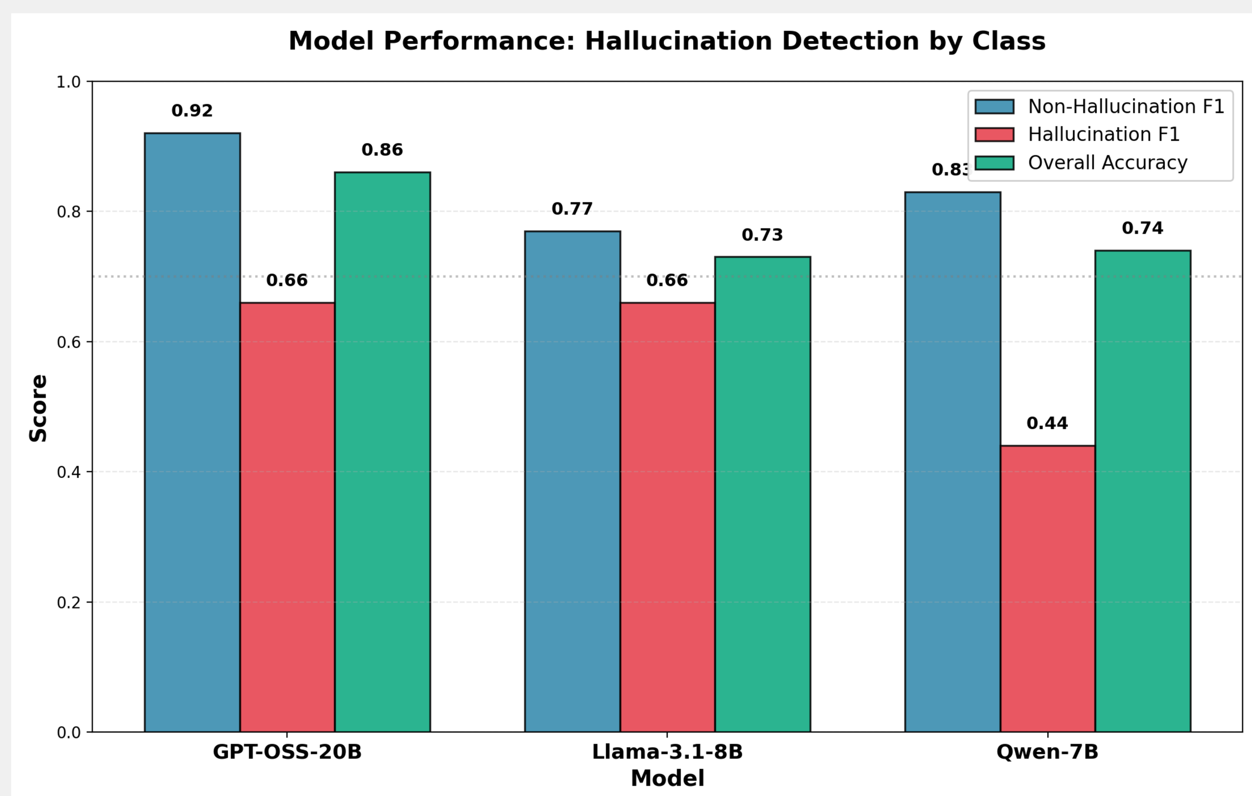
1. **Data Generation:** Mask ground-truth tool calls from agent responses. Label the sample by agreement with reference (correct vs hallucinated)
2. **Feature Extraction:** Extract 3 token positions from final transformer layer: t_func: Initial function name token
   T_args: All argument tokens (averaged)
   t_{end}: Closing delimiter token
   Concatenate: $z_i = [h_tfunc||mean(h_Targs)||h_tend]$
3. **Classifier Training:** Train Lightweight 2-layer MLP (512 hidden units)
4. **Use binary classifier to detect hallucinations:** Use MLP to predict hallucinations in real time

Experimental Setup

Dataset: Glaive Dataset (GlaiveAI 2024) used to create five specialized agents: **Quick Calculator, Personal Finance Suite, Health Assistant, Sustainability Assistant, and Digital Commerce Assistant**



Results

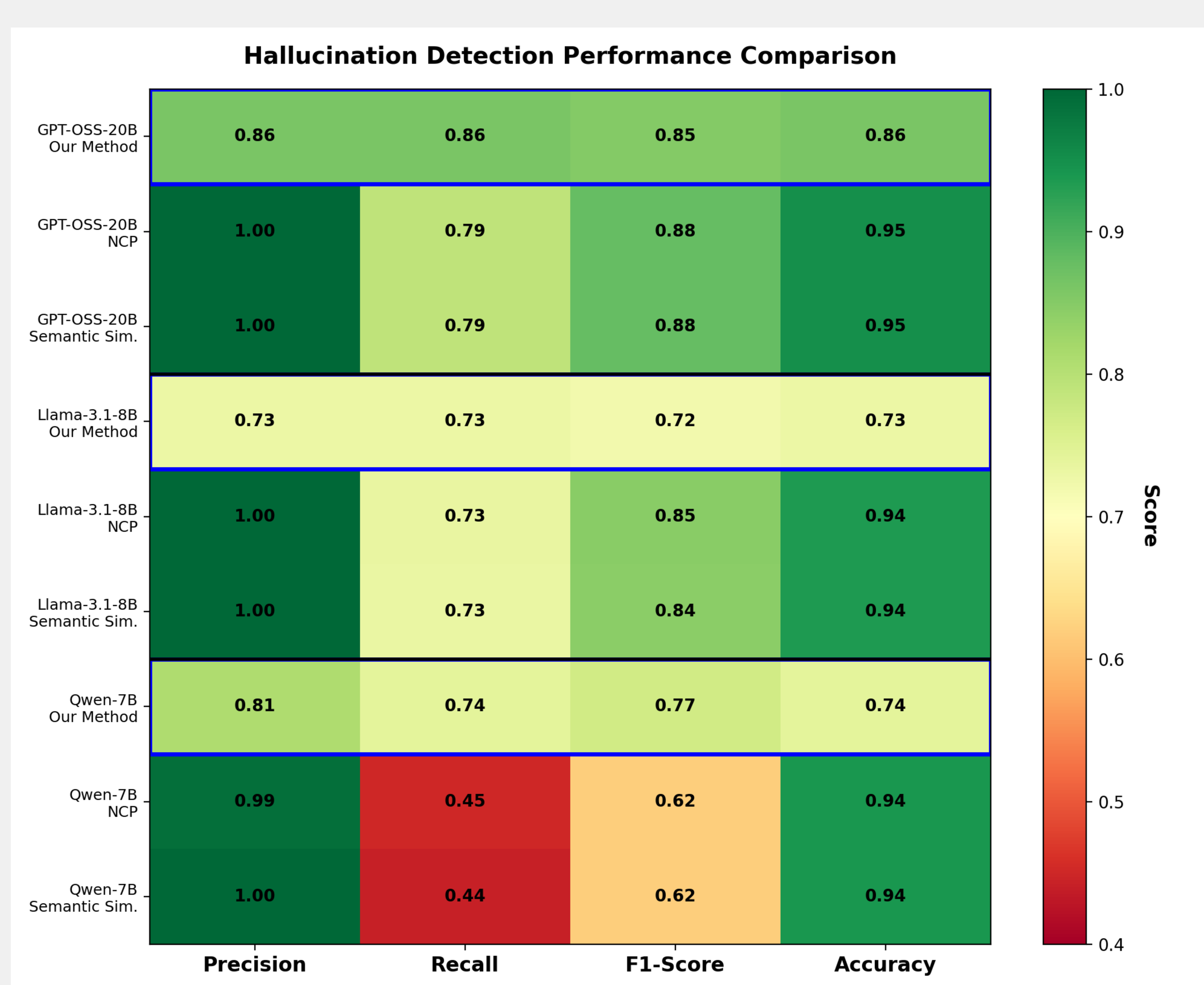


1. The model's **internal representations contain distributed hallucination signals**
2. The performance ceiling appears **inherent to the model's representations**

Experimental Results

Baselines:

1. **Non Contradiction Probability (NCP)** (Hou et al. 2025) is measured by prompting the agent multiple times (n=3) and measuring consistency using agreement.
2. **Semantic Similarity** (Kuhn, Arakelyan, and Percha 2023) is measured using cosine similarity of responses from the agent over multiple invocations (n=3).

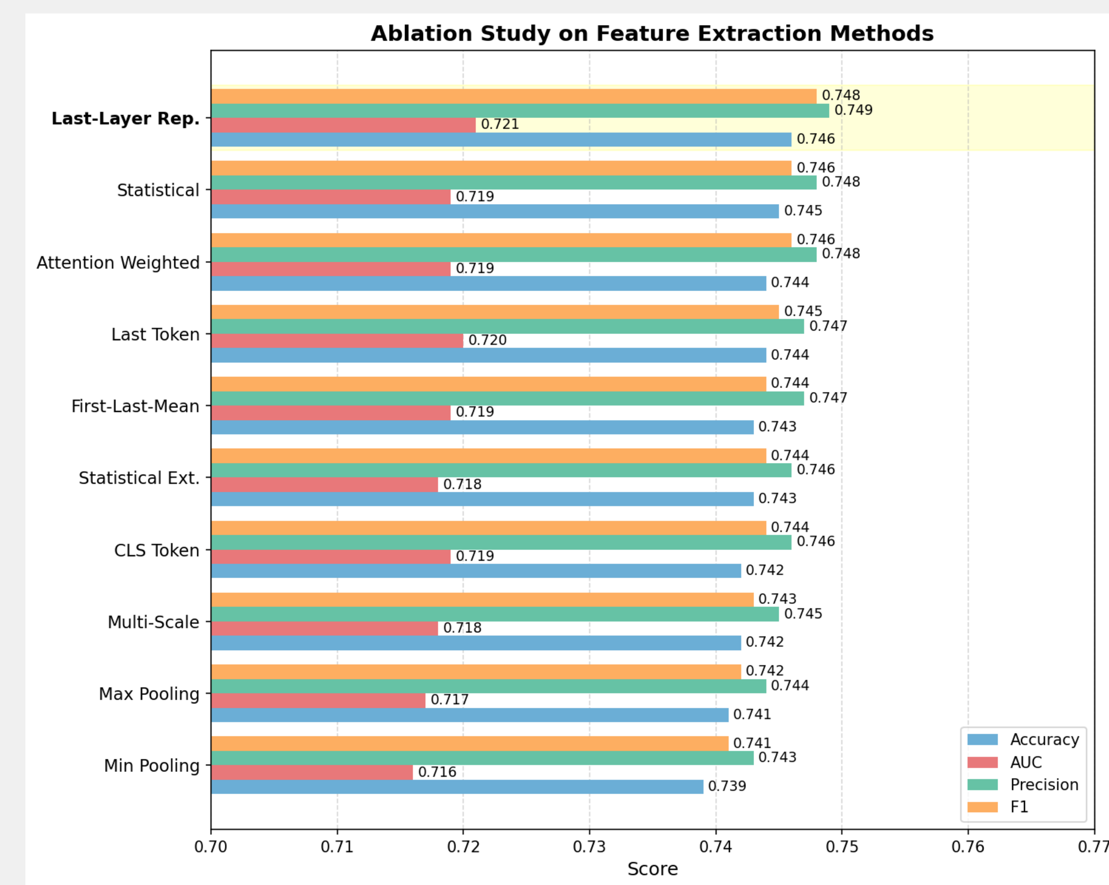


Achieves strong F1 scores while maintaining real-time performance

Analysis & Insights

Why Last Layer Representations Matter:

- **Simple aggregation methods are effective:** Mean pooling provides the best balance of performance and simplicity
- **Diminishing returns of complexity:** base transformer representations already capture the essential information
- **Sequence-level information is valuable:** Methods that aggregate information across the entire sequence consistently outperform single-token approaches
- **Robustness across methods:** transformer representations are robust and that multiple aggregation strategies can effectively capture hallucination patterns



Conclusion

This work shows that internal transformer representations provide a practical basis for detecting tool-calling hallucinations in LLM agents in real-time, using only a single forward pass and lightweight classifiers. By combining an unsupervised labeling pipeline with compact feature extraction and per-model MLPs, the approach achieves strong accuracy across diverse architectures while avoiding the additional computational overhead of multi-sample approaches or external verification methods.

References

2024. Glaive Function Calling Dataset v2. https://huggingface.co/datasets/glaiveai/glaive-function-calling-v2. Accessed: 2025-09-30.

Bigeard et al. (2025). Finance Agent Benchmark. *arXiv:2508.00828*

Islam et al. (2023). FinanceBench: A New Benchmark for Financial QA.

Bayless et al. (2025). Neurosymbolic Approach to NL Formalization.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. Advances in neural information processing systems, 33: 1877– 1901.

Chen et al. (2022). ConvFinQA: Conversational Finance QA. *EMNLP*.

Hou, B.; Zhang, Y.; Andreas, J.; and Chang, S. 2025. A Probabilistic Framework for LLM Hallucination Detection via Belief Tree Propagation. In Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), 3076–3099.

Kuhn, F.; Arakelyan, K.; and Percha, B. 2023. Semantic INconsistency Index for Hallucination Detection in LLMs. arXiv preprint arXiv:2503.05980.

Lewis et al. (2020). RAG for Knowledge-Intensive NLP Tasks. *NeurIPS*.

Schick, T.; Dwivedi-Yu, J.; Dess'ı, R.; Raileanu, R.; Lomeli, M.; Zettlemoyer, L.; Cancedda, N.; and Scialom, T. 2023. Toolformer: Language models can teach themselves to use tools. In Advances in Neural Information Processing Systems, volume 36.

LaTeX TikZposter