HALT: Hallucination Assessment via Latent Testing

Rohan Bhatnagar [*1] **Youran Sun** [*2] **Chi Andrew Zhang** [3] **Yixin Wen** [†4] **Haizhao Yang** [†1 2]

Abstract

Hallucination in large language models (LLMs) can be understood as a failure of faithful readout: although internal representations may encode uncertainty about a query, decoding pressures still yield a fluent answer. We propose lightweight residual probes that read hallucination risk directly from intermediate hidden states of question tokens, motivated by the hypothesis that these layers retain epistemic signals that are attenuated in the final decoding stage. The probe is a small auxiliary network whose computation is orders of magnitude cheaper than token generation and can be evaluated fully in parallel with inference, enabling near-instantaneous hallucination risk estimation with effectively zero added latency in low-risk cases. We deploy the probe as an agentic critic for fast selective generation and routing, allowing LLMs to immediately answer confident queries while delegating uncertain ones to stronger verification pipelines. Across four QA benchmarks and multiple LLM families, the method achieves strong AUROC and AURAC, generalizes under dataset shift, and reveals interpretable structure in intermediate representations, positioning fast internal uncertainty readout as a principled foundation for reliable agentic AI.

1. Introduction

Large language models (LLMs) have made remarkable progress recently (Yang et al., 2025; Comanici et al., 2025) and are being applied in an increasing number of real-world scenarios. However, hallucination, which refers to situations where a model produces information that appears plausible but is actually false or not supported by facts, undermines users' trust in LLMs and limits their use in critical settings. Therefore, developing effective methods for hallucination detection is critical. Ideally, such methods should maintain real-time responsiveness without increasing response latency or significantly increasing generation costs.

We propose a lightweight hallucination-detection method based on question intermediate representations that predicts hallucination risk efficiently and accurately. The additional computation introduced by our method is less than 1‰ of that required to generate a single token, so our approach adds negligible cost to generation. Moreover, the detector can be evaluated in parallel with inference, introducing no extra latency to the user. In essence, we aim to build a "lie detector" or "mind reader" for LLMs.

With an effective hallucination detector, we can build an *LLM router* that adaptively allocates computation. Given a user query, the default LLM begins generating a response while our detector estimates the hallucination risk in parallel. Because the detector only requires the question's intermediate representations, it can run concurrently with generation. If the predicted risk is low, the system returns the generated response as usual; otherwise, it routes the query to a slower but more reliable pipeline (e.g., a stronger model, reasoning-augmented generation, cross-model verification, or retrieval-augmented generation (RAG)). In the low-risk case, this design introduces zero additional latency; in the fallback case, the extra delay is less than the time to generate a single token.

Our motivation follows naturally from recent evidence that LLMs perform substantial hidden consideration during forward propagation. Prior work (Lindsey et al., 2025; Chen et al., 2025) suggests that intermediate representations encode reasoning, planning, and control signals that guide generation, yet these signals may not be faithfully expressed in the final text. This motivates directly reading out hallucination-related signals from intermediate representations, rather than relying on generated results.

Based on the above observations, we hypothesize that **intermediate representations encode uncertainty signals**. This is analogous to how students taking an exam can sense "I don't know this" for a question, yet would never write that on the answer sheet. LLMs exhibit a similar pattern because they are trained to provide an answer whenever

[*] Equal contribution [1] Department of Computer Science, University of Maryland, College Park, MD, USA [2] Department of Mathematics, University of Maryland, College Park, MD, USA [3] Department of Statistics, University of Chicago, Chicago, IL, USA [4] Department of Geography, University of Florida, Gainesville, FL, USA. Correspondence to: Yixin Wen <yixin.wen@ufl.edu>, Haizhao Yang <hzyang@umd.edu>.

Preprint. January 19, 2026.

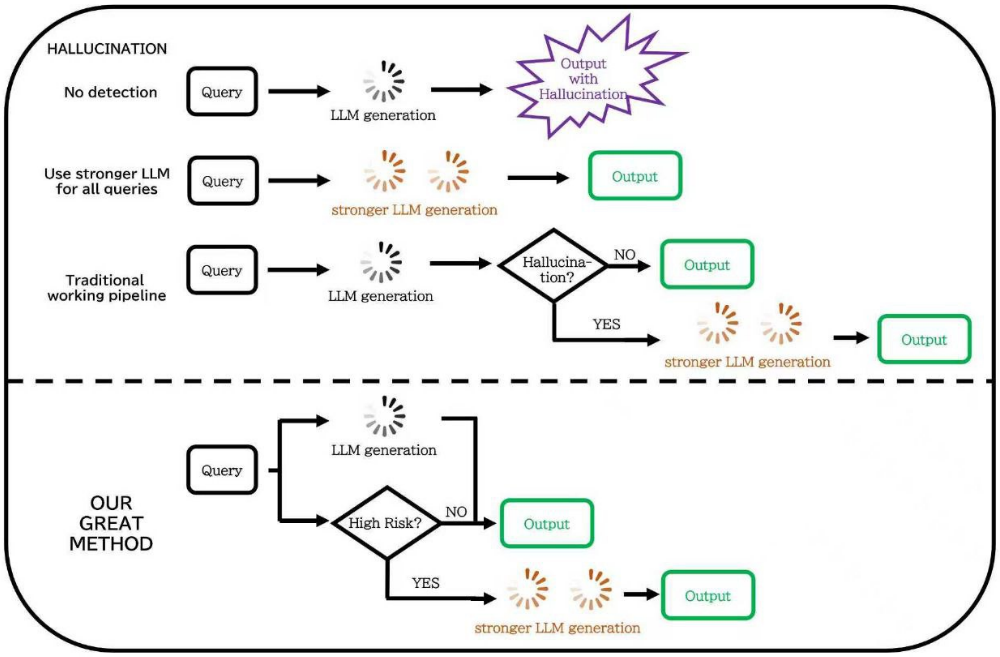


Figure 1. Comparison of hallucination handling strategies. Traditional detection pipelines wait for generation to complete before checking for hallucinations, doubling latency in the fallback case. Our method evaluates hallucination risk in parallel with generation, enabling zero-latency responses for confident queries while routing uncertain ones to stronger models.

possible. This motivates us to build a small detector that captures this internal sense of confusion. Moreover, this analogy suggests that intermediate layers should be more informative than the final output, just as the internal feeling differs from what appears on the answer sheet.

In our experiments, we indeed find that using representations from intermediate layers is often more effective than using the final-layer representations. We attribute this to the fact that the final layer has already been decoded into the token space, leading to significant information loss. In other words, the model contains internal features related to confidence or hallucination, but these features are unnecessary for output generation, so they may be discarded in the last few layers.

Furthermore, we only use the representations aligned to the question as input to the detector. This choice is driven by the need to balance detection quality and response latency. In our experiments, we find that question-only representations are already sufficient for accurate detection. Including answer representations yields a marginal improvement but incurs substantial latency overhead.

Our main contributions are as follows.

- We propose a lightweight hallucination detection method that incurs negligible computational overhead and can be evaluated in parallel with inference.
- Building on this detector, we develop a pipeline that routes queries with high hallucination risk to stronger models to improve generation quality without increasing response latency or token generation cost. Preliminary experiments show that this pipeline improves answer correctness by xxx.
- We conduct extensive ablations and identify several properties of intermediate representations. (a) Intermediate-layer representations are more effective than the final layer for hallucination detection. (b) While question-aligned representations are often sufficient for accurate risk estimation, including answer tokens improves our method.

2. Related Work

The perplexity of an LLM's answer can itself serve as an indicator for hallucination detection, but as shown in (Ren et al., 2023), it is unreliable. (Kuhn et al., 2023) and (Farquhar et al., 2024) propose semantic entropy. They require the LLM to generate multiple answers to a given question, then cluster them, and determine the hallucination likelihood based on the entropy of the clusters. Higher semantic entropy indicates a higher probability of hallucination. Similarly, (Lin et al., 2024), (Manakul et al., 2023), and (Chen et al., 2024) detect hallucinations based on consistency over multiple sampled answers.

Table 1. Summary of hallucination detection methods.

Method	Sampling	Features Used	Q/A Features Used
Perplexity	No Need	Output Logits	Answer
Semantic Entropy (Farquhar et al., 2024)	Need	Output	Answer
Lexical Similarity (Lin et al., 2024)	Need	Output	Answer
SelfcheckGPT (Manakul et al., 2023)	Need	Output	Answer
EigenScore (Chen et al., 2024)	Need	Middle Hidden States	Answer
P(I Know) (Kadavath et al., 2022)	No Need	Last Hidden State	Question
True Direction (Bürger et al., 2024)	No Need	Last Hidden State	Answer
HaloScope (Du et al., 2024)	No Need	Middle Hidden States	Answer
HARP (Hu et al., 2025)	No Need	Last Hidden State	Answer
Ours	No Need	Middle Hidden States	Question

However, a practical hallucination detector should require much less computation, at least smaller than the cost of generating an answer with the LLM. (Kadavath et al., 2022) trained a classifier that uses the last hidden state of the question to predict whether the model knows the answer, i.e., $P(\text{I know})$. (Bürger et al., 2024) used the last hidden state of the answer to learn, via linear regression, two directions, the True direction and the False direction, and performed hallucination detection within this two-dimensional subspace.

Based on the above methods, HARP (Hu et al., 2025) and HaloScope (Du et al., 2024) employ SVD to project the last or intermediate hidden states corresponding to the answer tokens into a low-dimensional subspace, after which a two-layer MLP is used to regress the hallucination score. As a follow-up to HaloScope, (Park et al., 2025) attempts to identify hallucinations by modifying the intermediate representations of the LLM and then classifying the last hidden state.

3. Methodology

Problem Definition We formulate hallucination detection as a *knowledge prediction* task. Given a user query q, our goal is to estimate whether the LLM contains the correct answer before or during generation. During the forward process, the LLM produces hidden representations $h_l \in \mathbb{R}^{N\times D}$ for each layer $l \in [0, L]$, where N is the sequence length, D is the hidden dimension, and L is the number of transformer layers. We use a lightweight auxiliary network g_θ that takes the hidden representation h as input and outputs a scalar score

$$p = g_\theta(h),$$

where $p \in [0, 1]$ represents the probability that the model can correctly answer the query q. A low value of p indicates that the model is uncertain or likely to hallucinate when generating the response.

Feature Extraction from LLM The detector relies on the LLM's hidden representations to infer whether the model is confident about a given query. In principle, representations from any transformer layer can be used as input. We choose to extract features from *intermediate layers* rather than the final one. As discussed earlier, the last layer is optimized for next-token prediction and may discard signals unrelated to generation. Intermediate layers, by contrast, retain richer representations that capture the model's internal uncertainty.

We also restrict the detector's input to hidden states corresponding only to the *question tokens*. Formally, the feature we use can be represented as

$$h_l = \text{Transformer}_l(q),$$

where h_l denotes the hidden representation of the query(question) q at layer l. This choice reduces computation and avoids the need to process the model's generated text. More importantly, it allows the hallucination likelihood to be estimated *before* any answer is produced. Methods that depend on analyzing the generated output must first wait for the model to finish responding, then, if the answer is found unreliable, invoke a stronger model to regenerate it, effectively doubling the latency. Our approach avoids this inefficiency by evaluating the question representations during the original forward pass, so the system can proactively switch to a more reliable model and introduce almost no additional delay. For factual benchmarks, we achieve performance comparable to or better than output-based methods, using only the question representations without waiting for the model's response.

Training Objective We construct the training data from question-answering benchmarks that include reference answers. For each question q with the standard answer a^*, the target LLM generates one answer a. We then obtain a correctness label by using an external judge to compare (q, a) with a^*. In particular, we use `gpt4o` for to produce more accurate labels than rule-based evaluation methods. In principle, multiple answers can be sampled for each question to estimate the probability of correctness. Still, in practice, we generate only one answer per question due to time and cost limitations. The resulting label is $y \in \{0,1\}$, where $y = 1$ means the generated answer is correct and $y = 0$ means it is hallucinated or incorrect.

Given the detector output $p \in [0,1]$, we employ a binary cross-entropy objective. For a dataset with M examples $\{(q_i, y_i)\}_{i=1}^{M}$ and detector scores p_i, we minimize

$$\mathcal{L} = -\frac{1}{M}\sum_{i=1}^{M}\big(y_i \log p_i + (1-y_i)\log(1-p_i)\big).$$

In closed-book question answering, the model generates answers solely from its parametric knowledge without access to external sources. An incorrect answer in this setting indicates that the model has produced content inconsistent with factual ground truth, which aligns with the definition of factual hallucination. We therefore treat correctness labels as a proxy for hallucination detection, following prior work in this area (Du et al., 2024; Farquhar et al., 2024).

Table 2. Detector architecture configurations. Input dimension varies by base LLM (e.g., 3584 for Qwen-2.5-7B).

Architecture	Hidden Dim	Layers	Params
MLP	128–1024	4	3M–37M
Transformer	256–512	4–8	4M–30M

Network Architecture (Hallucination Detector) We experiment with two architectures for the hallucination detector, MLP and Transformer, as summarized in Table 2. Both networks take the hidden representations of the question h_l as input and output a single confidence score p.

For the MLP architecture, the input must have a fixed length. Therefore the hidden representation $h_l \in \mathbb{R}^{N\times D}$ is first compressed into a single vector in $\mathbb{R}^D$. We consider several simple aggregation methods, including mean pooling, max pooling, and using the hidden state of the last token, which often corresponds to the question mark. We also consider applying principal component analysis (PCA) to the hidden representations and retain the top n principal components. The resulting pooled vector or concatenated principal components are then fed into an MLP that predict the confidence score p.

The transformer architecture is more flexible since it naturally processes sequence representations. In this case, the hidden states h_l are directly used as input to a lightweight transformer encoder. This model can capture token-level interactions within the question without requiring input-stage pooling. At the output stage, we apply attention pooling to aggregate the sequence into a single vector. Specifically, a small MLP computes a scalar score for each token, and the final representation is a softmax-weighted sum of all token representations.

4. Experiments

Table 3. Summary of four QA benchmarks used for hallucination detection evaluation.

Dataset	Size	Type	Topic
TriviaQA	99K	Open QA	Trivia
NQ-Open	92K	Open QA	Wikipedia
MMLU-Pro	12K	MCQ	Professional
WebQuestions	6K	Open QA	Entity

Datasets and models Our experiments use four open-domain question answering benchmarks, namely TriviaQA (Joshi et al., 2017), NQ-Open (Kwiatkowski et al., 2019), MMLU-Pro (Wang et al., 2024), and WebQuestions (Berant et al., 2013), summarized in Table 3. TriviaQA contains general knowledge trivia questions in an unfiltered no-context setting. NQ-Open consists of naturally occurring queries from Google Search paired with Wikipedia-based answers. MMLU-Pro provides multiple-choice questions spanning 14 professional and academic subjects, often requiring multi-step reasoning. WebQuestions includes entity-centric questions originally collected from the Google Suggest API. We evaluate our approach on three model families, namely LLaMA-2 (Touvron et al., 2023), Qwen2.5 (Yang et al., 2024), and Gemma-3 (Team, 2025). Ablation studies on model scale are conducted using the 7B, 14B, and 32B instruct variants of Qwen2.5. Detailed statistics on correctness labels for each model-dataset pair are provided in Table 5.

Evaluation Metrics AUROC (area under the ROC curve) is employed as the primary evaluation metric. The ROC (receiver operating characteristic) curve plots the true positive rate (TPR) against the false positive rate (FPR) across different classification thresholds. AUROC summarizes a binary classifier's ability to separate positives from negatives over all thresholds, ranging from 0 to 1, where higher values indicate stronger discriminative power.

In addition, accuracy and AURAC (area under the rejection–accuracy curve) are also reported. The RAC plots accuracy against coverage, where samples are ranked by confidence and coverage denotes the fraction of samples retained. AURAC evaluates whether the model's confidence is reliable, with higher values indicating a stronger alignment between correctness and confidence. Figure 3 provides a concrete example of a RAC curve, which may aid understanding of this metric.

Table 4. Hallucination detection performance (AUROC, %) on four QA benchmarks across three model families. Bold indicates best performance per model-dataset pair. * indicates severe label imbalance (see Table 5). Our method consistently outperforms baselines in most settings.

Model	Method	TriviaQA	NQ-Open	MMLU-Pro	WebQuestions
LLaMA 2 Chat 7B	HaloScope	77.40	67.0	**79.6**	72.6
	Semantic Entropy	81.23	77.57	76.76	80.12
	Ours (question)	82.10	79.87	71.22	80.79
	Ours (answer)	**88.96**	**83.76**	77.85*	**82.76**
Qwen-2.5-7B-Instruct	HaloScope	85.5	70.3	81.1	80.4
	Semantic Entropy	78.48	80.22	74.05	80.71
	Ours (question)	87.31	83.28	83.21	84.01
	Ours (answer)	**93.95**	**88.43**	**87.08**	**87.67**
Gemma-3-4b-it	HaloScope	76.5	79.62	**77.7**	76.7
	Semantic Entropy	76.08	75.48	57.92	76.76
	Ours (question)	83.93	82.96	74.65	79.63
	Ours (answer)	**88.80**	**85.65***	75.53*	**82.51**

Baselines We compare our method with the highest performing open baselines, including HaloScope (Du et al., 2024) and Semantic Entropy (Farquhar et al., 2024).

Correctness labeling We use GPT-4o to obtain correctness labels for LLM-generated answers. For each question, we prompt GPT-4o with the original question, the ground-truth answer, and the LLM-generated answer, asking it to return a binary correctness label (see Appendix A for the full prompt). This approach provides more robust evaluation than rule-based string matching, particularly for open-ended questions where semantically equivalent answers may differ in surface form.

4.1. Main Results

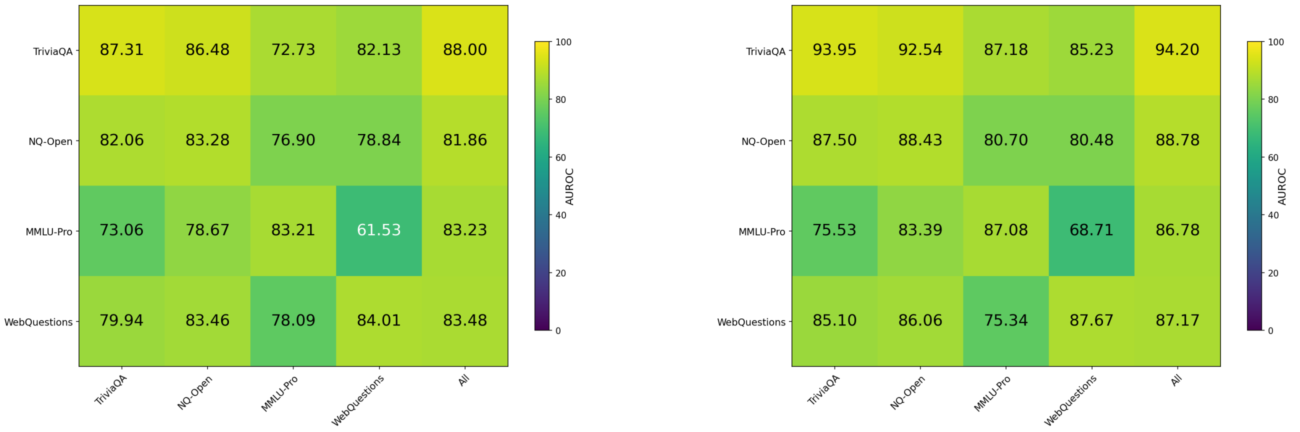
Table 4 presents the main results. Our method achieves the best AUROC on 10 out of 12 model-dataset combinations. Qwen-2.5-7B-Instruct shows the strongest performance, where our approach outperforms all baselines across all four benchmarks, with gains of up to 8.5 points on TriviaQA and 8.2 points on NQ-Open. On LLaMA 2 Chat 7B and Gemma-3-4b-it, we observe similar improvements, surpassing prior methods by 7.7 and 12.3 points on TriviaQA respectively. Using answer token representations (Ours answer) consistently outperforms question token representations (Ours question) across all settings, suggesting that answer tokens encode richer information about the model's factual confidence. The two exceptions occur on MMLU-Pro for LLaMA 2 and Gemma-3, where HaloScope achieves slightly higher AUROC. We attribute this to two factors: (1) a format mismatch, as TriviaQA, NQ-Open, and WebQuestions are open-ended QA tasks while MMLU-Pro is a multiple-choice benchmark where the model selects from predefined options rather than generating free-form answers; and (2) label imbalance, as LLaMA 2 and Gemma-3 achieve only 15.4% and 27.1% accuracy on MMLU-Pro respectively (see Table 5), resulting in limited positive samples for training the detector. Overall, these results demonstrate that intermediate-layer representations provide a strong signal for hallucination detection, competitive with or superior to methods that require sampling multiple outputs.

We evaluate whether our detector generalizes under distribution shift. Unless otherwise specified, we extract hidden states from layer 19 of Qwen-2.5-7B-Instruct. We train the detector on each of the four datasets and evaluate it on the remaining datasets. We also train on the union of all datasets (*All*) and test on each dataset. Figure 2 reports the AUROC for each train–test pair.

The detector exhibits strong out-of-distribution generalization. Most off-diagonal entries remain in the 75–93 AUROC range, indicating that the intermediate representations capture dataset-agnostic signals predictive of hallucination rather than dataset-specific patterns. Training on the union (*All*) yields a robust model that achieves AUROC of 94.20 on TriviaQA, 88.78 on NQ-Open, 86.78 on MMLU-Pro, and 87.17 on WebQuestions. While *All* does not uniformly dominate the best single-source model for every target, it provides competitive and stable performance across all datasets, reducing sensitivity to the particular source distribution. These results support our conclusion that the learned features are largely transferable across different QA domains.

4.2. Accuracy after Hallucination Removed

Beyond binary hallucination detection, our method can be used for *selective prediction*: abstaining from answering when the model is likely to hallucinate. Crucially, because we use only question token representations, the detector



(a) Using question tokens only

(b) Using question + answer tokens

Figure 2. Out-of-distribution generalization. Heatmap of AUROC when training the detector on one dataset (columns) and evaluating it on another (rows), including training on the union of all datasets (*All*).

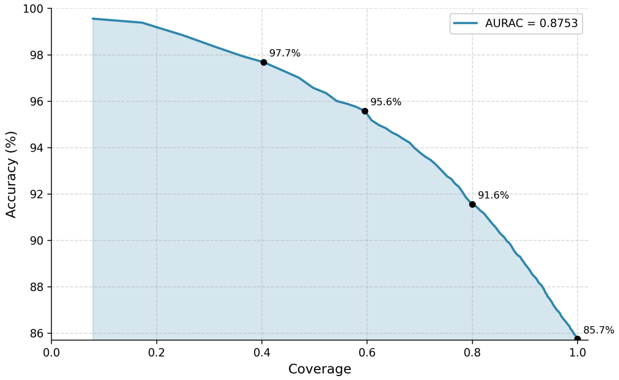


Figure 3. Rejection-Accuracy Curve (RAC) for Qwen-2.5-7B on TriviaQA using only question token representations. As we reject samples with low confidence, accuracy increases from 85.7% at full coverage to 97.7% at 40% coverage. AURAC = 0.8753 indicates strong alignment between confidence and correctness.

can evaluate hallucination risk *before* the model generates any answer, introducing zero additional latency. We use the detector's output probability as a confidence score and reject samples below a threshold. Figure 3 shows the rejection-accuracy curve (RAC) for Qwen-2.5-7B on TriviaQA, where we vary the confidence threshold and measure accuracy on the retained samples.

At full coverage, the model achieves 85.7% accuracy. By rejecting the 20% least confident samples, accuracy rises to 91.6%; at 60% coverage, it reaches 95.6%; and at 40% coverage, it climbs to 97.7%. The area under the RAC (AURAC = 0.8753) quantifies this strong alignment between detector confidence and correctness.

This capability supports the LLM router architecture described in Figure 1: queries flagged as high-risk can be routed to more reliable pipelines, while confident queries proceed without additional latency. The steep accuracy gains at moderate rejection rates suggest that even conservative abstention thresholds can substantially improve system reliability.

5. Ablation Studies

To understand the design and limitations underlying our hallucination detector, we conduct several ablation studies analyzing how different component affect performance. In particular, we explore (1) which transformer layers provide the most informative representations for hallucination detection, (2) how model scale influences both classification separability and ranking quality under selective deployment, (3) how different detector architectures and representation aggregation strategies impact performance, and (4) whether we can use a detector trained on the question and answer to outperform a detector trained on only the question, *during* generation for parallel evaluation. Unless otherwise stated, all ablations are performed using the Qwen family of models.

5.1. Effect of Intermediate Layers on Performance

Recent work has shown that different layers in transformers encode qualitatively different information: early layers capture surface-level features, while deeper layers encode higher-level semantics and factual knowledge (Mir, 2025; Ni et al., 2025). We hypothesize that intermediate layers, which balance low-level token representations with high-level semantic understanding, may provide the strongest signals for hallucination detection. To test this, we train separate detectors on hidden states extracted from each layer and compare their performance.

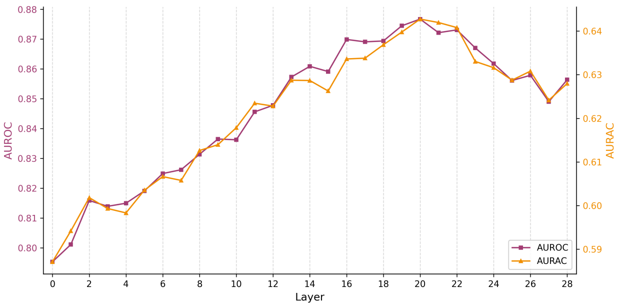


Figure 4. AUROC and AURAC across layers 0–28 for Qwen2.5-7B on Webquestions. Both metrics peak around layer 20, suggesting that intermediate-to-late layers encode the most informative signals for uncertainty detection.

Figure 4 shows AUROC and AURAC across all 29 layers of Qwen2.5-7B. Both metrics increase monotonically from early layers and peak around layer 20, achieving AUROC of 0.877 and AURAC of 0.643. Performance then gradually declines in the final layers. This pattern suggests that intermediate-to-late layers, which are believed to encode higher-level semantic and factual information, provide the strongest signals for detecting hallucinations. We observe similar trends for LLaMA-2-7B and Gemma-3-4B (see Appendix C in the Appendix).

5.2. Effect of Model Size on Ranking Quality

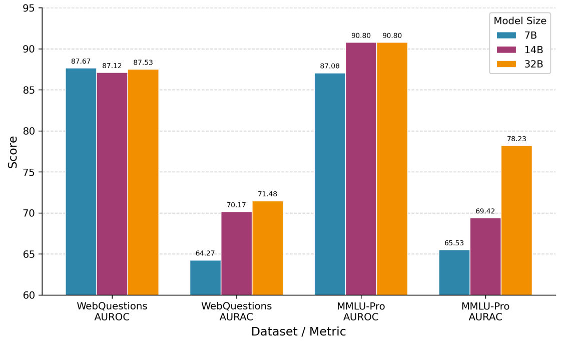


Figure 5. Effect of model scale (7B, 14B, 32B) on AUROC and AURAC. While AUROC remains largely stable across scales, AURAC increases consistently, indicating improved ranking quality under selective deployment for larger models.

We study the effect of model scale on selective deployment performance by evaluating Qwen2.5 models at 7B, 14B, and 32B parameters on the same question sets. As shown in Figure 5, AUROC remains largely stable across scales on both WebQuestions and MMLU-Pro, indicating that overall classification separability changes little with model size. In contrast, AURAC improves consistently with scale, reflecting substantially better ranking quality for selective prediction. The improvement is most pronounced on MMLU-Pro, where the 32B model achieves a 12.7-point AURAC gain over the 7B model. This trend indicates that larger models produce confidence scores that are better aligned with correctness, enabling more effective filtering of unreliable responses under selective deployment.

5.3. Effect of Detector Architecture

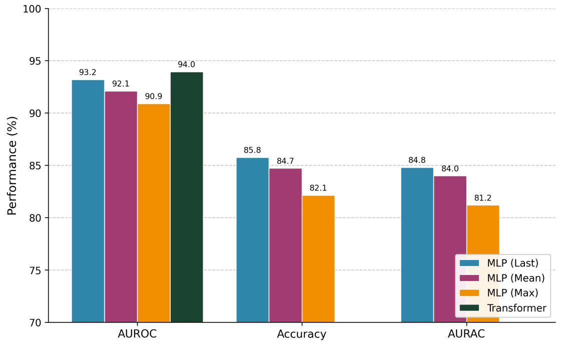


Figure 6. Comparison of detector architectures on TriviaQA (Qwen2.5-7B, layer 19). Among MLP variants, last-token pooling achieves the highest AUROC (93.2%), outperforming mean pooling (92.1%) and max pooling (90.9%). The transformer architecture achieves the best AUROC (93.95%) with attention pooling.

We compare the two detector architectures described in Section 3: an MLP with input-stage pooling and a lightweight transformer with attention pooling (Table 2). For the MLP, we further compare three pooling strategies: last token, mean pooling, and max pooling. All experiments use hidden states from layer 19 of Qwen2.5-7B on TriviaQA.

As shown in Figure 6, last-token pooling consistently outperforms mean and max pooling across all metrics, achieving 93.2% AUROC compared to 92.1% and 90.9%, respectively. The transformer architecture with attention pooling achieves the highest AUROC (93.95%), slightly outperforming the best MLP variant. Based on these results, we use the transformer architecture with attention pooling in our main experiments.

5.4. Effect of Number of Answer Tokens Used

Our main experiments compare two settings: using only question tokens versus using question plus all answer tokens. This ablation fills the gap between these two extremes by progressively increasing the proportion of answer tokens. Specifically, we train detectors using question tokens plus the first $X\%$ of answer tokens, where X ranges from 5% to 100%. Figure 7 shows the results on TriviaQA using Qwen-2.5-7B.

Both AUROC and AURAC increase monotonically with the token threshold. At 5% answer tokens, the detector achieves 82.2% AUROC; at 100%, this rises to 87.1%. AURAC follows a similar trend, increasing from 0.628 to 0.655. These results confirm that answer tokens provide additional signal

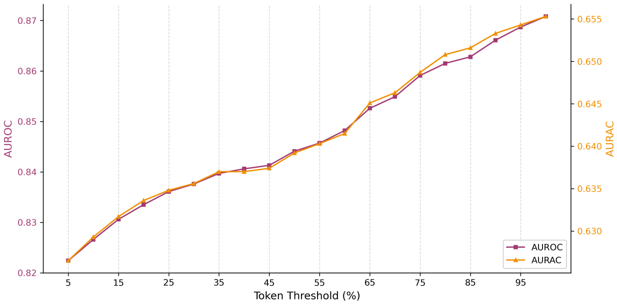


Figure 7. Effect of answer token inclusion on detection performance. We vary the percentage of answer tokens used (5%–100%) alongside all question tokens. Both AUROC and AURAC increase monotonically with more answer tokens.

for hallucination detection, and this signal accumulates as more tokens are included.

6. Discussion

Limitations Our experiments focus on single-turn question-answering tasks with relatively short contexts. In multi-turn dialogues or long-context scenarios, the intermediate representations of the current question may be influenced by the preceding conversation history, potentially degrading detector performance. Adapting our method to such settings may require collecting training data that reflects these longer contexts, and employing compression or sliding window strategies to select informative residual representations for training and inference. However, we emphasize that the length of the *generated answer* does not affect our method, since we are able to achieve superior results in most settings depending only on question token representations, and can make predictions before answer tokens are produced.

Scaling Trends In our model scale ablation (Figure 5), we find that our method performs better when applied to larger models. A similar phenomenon has been reported in prior work (Du et al., 2024; Kadavath et al., 2022). We therefore conjecture that our method generalizes to stronger, larger LLMs. One possible explanation is that larger models have greater capacity and may perform richer latent computation beyond next-token prediction. As a result, the final text can be less faithful to the intermediate state. This can increase the gap between intermediate representations and surface outputs, making intermediate features more informative for our detector and improving its effectiveness on larger models.

Future Directions Beyond hallucination detection and selective prediction, our method may benefit reinforcement learning for LLMs. In tree search algorithms such as MCTS, the uncertainty signal from our detector could guide exploration: branches where the model is uncertain warrant more exploration, while confident branches can be pruned early. This could help prevent the model from converging to local minima driven by overconfident but incorrect predictions, leading to more robust policy optimization.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Berant, J., Chou, A., Frostig, R., and Liang, P. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1533–1544, 2013. URL https://aclanthology.org/D13-1160.

Bürger, L., Hamprecht, F. A., and Nadler, B. Truth is universal: Robust detection of lies in llms, 2024. URL https://arxiv.org/abs/2407.12831.

Chen, C., Liu, K., Chen, Z., Gu, Y., Wu, Y., Tao, M., Fu, Z., and Ye, J. Inside: Llms' internal states retain the power of hallucination detection, 2024. URL https://arxiv.org/abs/2402.03744.

Chen, Y., Benton, J., Radhakrishnan, A., Uesato, J., Denison, C., Schulman, J., Somani, A., Hase, P., Wagner, M., Roger, F., Mikulik, V., Bowman, S. R., Leike, J., Kaplan, J., and Perez, E. Reasoning models don't always say what they think, 2025. URL https://arxiv.org/abs/2505.05410.

Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., Marris, L., Petulla, S., et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL https://arxiv.org/abs/2507.06261.

Du, X., Xiao, C., and Li, Y. Haloscope: Harnessing unlabeled llm generations for hallucination detection, 2024.

Farquhar, S., Kossen, J., Kuhn, L., and Gal, Y. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07421-0. URL https://doi.org/10.1038/s41586-024-07421-0.

Hu, J., Tu, G., Cheng, S., Li, J., Wang, J., Chen, R., Zhou, Z., and Shan, D. Harp: Hallucination detection via reasoning subspace projection, 2025.

Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension, 2017. URL https://arxiv.org/abs/1705.03551.

Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Hatfield-Dodds, Z., DasSarma, N., Tran-Johnson, E., Johnston, S., El-Showk, S., Jones, A., Elhage, N., Hume, T., Chen, A., Bai, Y., Bowman, S., Fort, S., Ganguli, D., Hernandez, D., Jacobson, J., Kernion, J., Kravec, S., Lovitt, L., Ndousse, K., Olsson, C., Ringer, S., Amodei, D., Brown, T., Clark, J., Joseph, N., Mann, B., McCandlish, S., Olah, C., and Kaplan, J. Language models (mostly) know what they know, 2022. URL https://arxiv.org/abs/2207.05221.

Kuhn, L., Gal, Y., and Farquhar, S. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation, 2023. URL https://arxiv.org/abs/2302.09664.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A. P., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., Toutanova, K., Jones, L., Kelcey, M., Chang, M.-W., Dai, A. M., Uszkoreit, J., Le, Q., and Petrov, S. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. URL https://aclanthology.org/Q19-1026.

Lin, Z., Trivedi, S., and Sun, J. Generating with confidence: Uncertainty quantification for black-box large language models, 2024. URL https://arxiv.org/abs/2305.19187.

Lindsey, J., Gurnee, W., Ameisen, E., Chen, B., Pearce, A., Turner, N. L., Citro, C., Abrahams, D., Carter, S., Hosmer, B., Marcus, J., Sklar, M., Templeton, A., Bricken, T., McDougall, C., Cunningham, H., Henighan, T., Jermyn, A., Jones, A., Persic, A., Qi, Z., Thompson, T. B., Zimmerman, S., Rivoire, K., Conerly, T., Olah, C., and Batson, J. On the biology of a large language model. https://transformer-circuits.pub/2025/attribution-graphs/biology.html, 2025. Accessed: 2025-10-07; mechanistic interpretability study of Claude 3.5 Haiku.

Manakul, P., Liusie, A., and Gales, M. J. F. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models, 2023. URL https://arxiv.org/abs/2303.08896.

Mir, A. H. The geometry of truth: Layer-wise semantic dynamics for hallucination detection in large language models, 2025. URL https://arxiv.org/abs/2510.04933.

Ni, S., Bi, K., Guo, J., Tang, M., Wu, J., Han, Z., and Cheng, X. Annotation-efficient universal honesty alignment, 2025. URL https://arxiv.org/abs/2510.17509.

Park, S., Du, X., Yeh, M.-H., Wang, H., and Li, Y. Steer llm latents for hallucination detection, 2025. URL https://arxiv.org/abs/2503.01917.

Ren, J., Luo, J., Zhao, Y., Krishna, K., Saleh, M., Lakshminarayanan, B., and Liu, P. J. Out-of-distribution detection and selective generation for conditional language models, 2023. URL https://arxiv.org/abs/2209.15558.

Team, G. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Kirnap, U., Kivlichan, I., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Koura, P. S., Bhosale, S., Narayanan, D., Angelidis, A., Shankar, V., Wolf, T., Rodriguez, A., Stojanov, S., Lample, G., Rocktäschel, T., Joulin, A., Bojanowski, P., Grave, E., and Conneau, A. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Wang, Y., Ma, X., Zhang, G., Ni, Y., Chandra, A., Guo, S., Ren, W., Arulraj, A., He, X., Jiang, Z., Li, T., Ku, M., Wang, K., Zhuang, A., Fan, R., Yue, X., and Chen, W. Mmlu-pro: A more robust and challenging multitask language understanding benchmark, 2024. URL https://arxiv.org/abs/2406.01574.

Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Tang, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., and Qiu, Z. Qwen2.5 technical report. arXiv preprint arXiv:2412.15115v2, 2024. URL https://arxiv.org/abs/2412.15115v2.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., and Qiu, Z. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

A. Correctness Evaluation Prompt

We use the following prompt template to obtain correctness labels from GPT-4o.

```
Evaluate whether the generated answer is CORRECT or INCORRECT.
Question: {question}
Ground truth: {answer}
Generated: {generated_text}
A generated answer is CORRECT if it expresses the same meaning
as the ground truth, without introducing incorrect, conflicting,
or extra information. Otherwise, it is INCORRECT.
Respond with EXACTLY "true" or "false".
```

B. Dataset Statistics

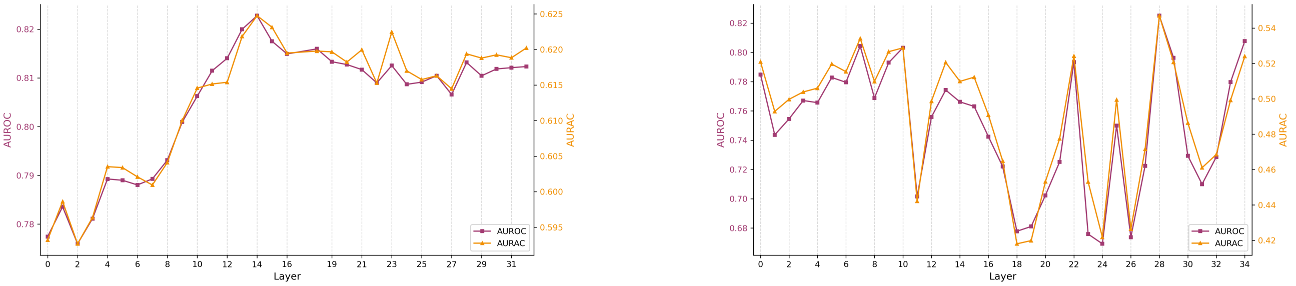
Table 5. Correctness label statistics across models and benchmarks. Each model has two rows: the first shows the number of correct/incorrect samples, and the second shows the corresponding accuracy (%). For NQ-Open and TriviaQA, 20K samples were randomly selected from the original training set; the remaining benchmarks use full training sets. All benchmarks include the complete test set. The counts shown represent the combined training subset and test set.

Model	Metric	TriviaQA	NQ-Open	MMLU-Pro	WebQuestions
Qwen2.5-7B	Count	17,743 / 13,570	7,287 / 16,323	4,403 / 7,629	2,153 / 3,645
	Accuracy	56.7%	30.9%	36.6%	37.1%
LLaMA-2-7B	Count	18,585 / 12,728	7,946 / 15,664	1,851 / 10,180	2,215 / 3,570
	Accuracy	59.4%	33.7%	15.4%	38.3%
Gemma-3-4B	Count	14,945 / 16,368	5,537 / 18,037	3,264 / 8,767	1,835 / 3,975
	Accuracy	47.7%	23.5 %	27.1%	31.6%

C. Ablation Study 2: Layer Ablation on Additional Models

We extend our layer ablation study to LLaMA-2-7B and Gemma-3-4B to verify that the trend observed in Qwen2.5-7B generalizes across model families. Figure 8 shows AUROC and AURAC across all layers for both models on the WebQuestions dataset.

For LLaMA-2-7B (32 layers), performance peaks around layer 14, with AUROC and AURAC declining in both earlier and later layers. For Gemma-3-4B (26 layers), the peak occurs around layer 18–20. Interestingly, the Gemma-3-4B curve exhibits notable fluctuations across layers, suggesting that different layers may encode qualitatively different information; this warrants further investigation. In both cases, intermediate-to-late layers outperform early layers and the final layers, confirming that the optimal layer for hallucination detection is not the last layer but rather an intermediate one. This pattern aligns with findings in interpretability literature suggesting that middle layers encode richer semantic representations.



(a) LLaMA-2-7B

(b) Gemma-3-4B

Figure 8. Layer ablation results for LLaMA-2-7B and Gemma-3-4B on WebQuestions. Both models show peak performance at intermediate layers, consistent with the trend observed in Qwen2.5-7B.