EnsemHalDet: Robust VLM Hallucination Detection via Ensemble of Internal State Detectors

Anonymous ACL submission

Abstract

Vision-Language Models (VLMs) excel at multimodal tasks, but they remain vulnerable to hallucinations that are factually incorrect or ungrounded in the input image. Recent works show that internal representations enable efficient and accurate hallucination detection, but existing methods often rely on a single representation or detector, limiting their ability to capture diverse hallucination signals. In this paper, we propose EnsemHalDet, an ensemble-based hallucination detection framework that leverages multiple internal representations of VLMs, including attention outputs and hidden states. EnsemHalDet trains independent detectors for each representation and combines them through ensemble learning. Experimental results across multiple VQA datasets and VLMs show that EnsemHalDet generally outperforms prior methods and single-detector models in terms of AUC. These results demonstrate that ensembling diverse internal signals significantly improves robustness in multimodal hallucination detection.[1]

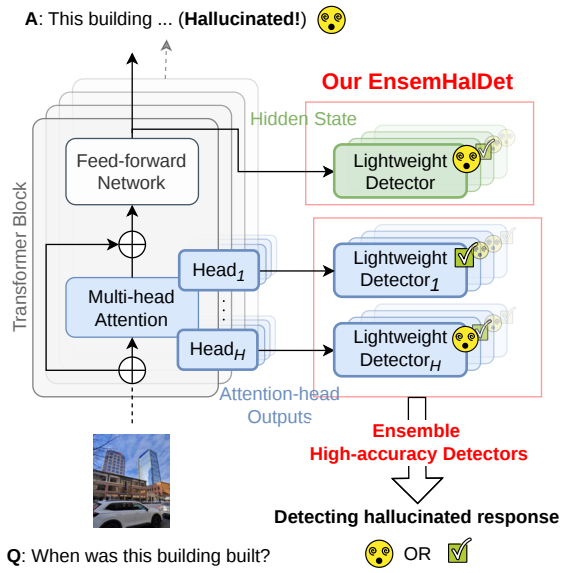


Figure 1: EnsemHalDet is a hallucination detection framework for VLMs that leverages ensemble learning to integrate multiple detectors trained on internal representations, enabling more robust and accurate hallucination detection.

1 Introduction

Reliable prediction often requires more than a single view of the data. In machine learning, ensemble learning is a well-established strategy for improving generalization and robustness by combining multiple predictors that are individually informative yet make partially different errors (Dietterich, 2000; Kuncheva, 2004). When different predictors capture complementary signals, their combination can yield a more reliable decision than any single predictor alone.

The ensemble learning can be applied to hallucination detection in Vision-Language Models (VLMs). Although VLMs have shown strong performance across a wide range of multimodal tasks (Liu et al., 2023; Yin et al., 2024; Chen et al., 2024b; Yue et al., 2025), they still suffer from hallucinations, where the models confidently generate outputs that are either factually incorrect or ungrounded in the inputs. This remains a serious challenge from the perspectives of reliability and safety (Chen et al., 2024a; Yang et al., 2025; Jiang et al., 2025). However, such failures are difficult to detect because unreliability may not appear in a single signal. Instead, it can be distributed across outputs, probabilities, and internal representations.

Internal-representation-based hallucination detection has emerged as a promising approach. Unlike methods that rely on external knowledge (Li et al., 2023a; Min et al., 2023) or post-hoc analysis of generated outputs (Manakul et al., 2023), these methods detect hallucination signals from a

[1]The source code will be available at https://anonymous.4open.science/r/line_hal_detection-CBC5

model's internal states. By training detectors in advance, they can potentially provide faster and more accurate detection by capturing signals that are not directly observable from the generated output (Azaria and Mitchell, 2023; Su et al., 2024).

However, most existing methods rely on a single internal representation (Azaria and Mitchell, 2023; Su et al., 2024; Zhang et al., 2025b), or single-detector using concatenated features from multiple internal states (Nath et al., 2025; Zhang et al., 2025a). As a result, they may fail to capture the diverse characteristics of hallucinations. To support this, prior work (Deng et al., 2026) has reported that these approaches lack robustness across tasks and domains. We hypothesize that integrating hallucination signals captured by different internal representations through ensemble learning can improve hallucination detection accuracy and robustness.

In this paper, we propose EnsemHalDet, an ensemble-based hallucination detection framework for VLMs that trains independent detectors on attention outputs and hidden states and combines them through detector-level ensemble grounded in the ensemble learning theory. Specifically, EnsemHalDet extracts and token-averages the attention outputs and hidden states from the VLMs to serve as features. Then we train individual detectors for each internal state and aggregate them using stacking ensemble.

To evaluate EnsemHalDet, we formulate hallucination detection as a binary classification task using multiple VQA datasets and VLMs. The results show that EnsemHalDet generally outperforms existing baselines in most settings. The improvement is also found in cross-domain settings, where detectors are trained and tested on different datasets. Ablation studies further demonstrate that the ensemble-based detectors outperform single-representation detectors. Additional analyses reveal that detectors trained on different internal states exhibit complementarity, and ensembling them improves both robustness and accuracy, especially for detectors based on attention-head outputs. EnsemHalDet is also computationally efficient and model-agnostic for open-source models, as it relies on lightweight classifiers.

2 Related Work

2.1 Hallucinations in Vision-Language Models

Hallucination in language models refers to the phenomenon where the model generates fluent and plausible content that is not based on the input or facts (Koehn and Knowles, 2017; Maynez et al., 2020; Ji et al., 2023). Hallucinations in VLMs are characterized not only by linguistic errors inherited from language models but also by generations that are inconsistent with the visual input (Huang et al., 2025), such as object hallucinations referring to nonexistent objects, attribute hallucinations involving incorrect object properties, or multi-modal conflicting hallucinations that contradict visual inputs (Liu et al., 2025).

2.2 Hallucination Detection Methods

Existing hallucination detection methods can be broadly categorized into three groups: output-based methods, external-knowledge-based methods, and internal-representation-based methods.

Output-based methods detect hallucinations by analyzing generated texts or prediction statistics. These methods analyze model outputs, such as consistency across multiple samples (e.g., SelfCheckGPT (Manakul et al., 2023)) or token-level uncertainty / log-probabilities (Guerreiro et al., 2023; Zhang et al., 2023). While model-agnostic, they often fail to detect "confident hallucinations".

External-knowledge-based methods detect hallucinations by comparing against external knowledge resources like Wikipedia or web searches to verify facts (Li et al., 2023a; Min et al., 2023). Although these methods can achieve high detection accuracy, they incur high computational costs and depend on retrieval accuracy and the quality of external data.

Internal-representation-based approaches are based on the premise that a model's internal states encode the truthfulness of its outputs (Li et al., 2023b; Marks and Tegmark, 2024). Early methods focused on training classifiers using single features (SAPLMA (Azaria and Mitchell, 2023), MIND (Su et al., 2024)). Recent studies have expanded this to multiple features, such as cross-attention patterns (DHCP (Zhang et al., 2025b)) or the concatenation of hidden states and FFN activations into a single feature vector (MHAD (Zhang et al., 2025a)). HALLUSHIFT++ (Nath et al., 2025) projects internal states into semantically interpretable statistics rather than directly concatenating raw internal representations. Internal-representation-based methods tend to achieve higher accuracy than other approaches. However, most existing methods rely on a single internal representation or a single-detector, which may limit their ability to capture the diverse characteristics of hallucinations.

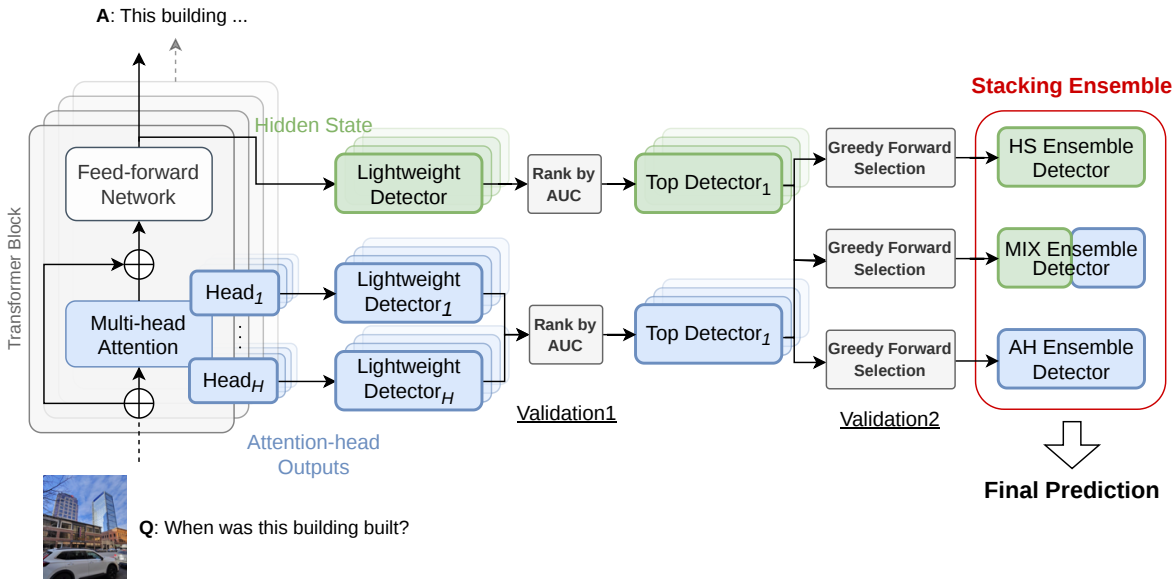


Figure 2: Overview of EnsemHalDet: Attention-head outputs and hidden states are extracted from multiple layers as internal representations. For each representation, binary classifiers are trained via logistic regression to detect hallucinations. The trained detectors are first ranked on validation$_1$, filtered to top-K candidates, further refined by greedy forward selection on validation$_2$, and finally combined by a stacking-based meta-classifier (logistic regression).

3 EnsemHalDet

An overview of the proposed method is illustrated in Figure 2. EnsemHalDet trains hallucination detectors independently for each attention-head and the hidden states of each layer, and then ensembles these detectors.

3.1 Extract Internal State

We extract attention outputs and hidden states. Attention outputs are obtained from the multi-head attention module before the output projection and hidden states are taken after the final layer normalization of each transformer block. Attention outputs are extracted at the head level, whereas hidden states are extracted at the layer level.

During generation, at each decoding step t, we extract the vector corresponding to the last token from each internal representation, denoted as $\mathbf{e}_t$, which is used for the next token's prediction. We then compute the average over all generated tokens (T tokens),

$$\bar{\mathbf{e}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{e}_t, \tag{1}$$

and use this averaged vector as the feature representation for each internal state. The hidden states are compressed to the same dimension as attention outputs using principal component analysis (PCA) because applying PCA consistently improves performance (Appendix Table 8).

3.2 Train Detectors

We train hallucination detectors separately for the attention outputs of each layer and head, as well as for the hidden states of each layer. Given a model with L layers and H attention-heads, this results in $L \times H$ attention-based detectors and L hidden-state-based detectors. Each detector is trained via supervised learning using logistic regression, with the extracted internal representations $\bar{\mathbf{e}}$ as input features and a binary label y indicating whether the response is hallucinated:

$$y = \sigma\left(\mathbf{w_1}^{\top}\bar{\mathbf{e}} + b_1\right), \tag{2}$$

where $\sigma(\cdot)$ denotes the sigmoid function, $\mathbf{w_1}$ is the weight vector, and b_1 is the bias term. We use logistic regression due to its simplicity and computational efficiency. All detectors are trained using logistic regression with L2 regularization, a regularization strength of 1.0 (C=1.0) to reduce overfitting while keeping the setup simple, and a maximum of 300 optimization iterations to ensure convergence. We use the same hyperparameters for all detectors. A detector output probability of 0.5 or higher is regarded as indicating a hallucination.

3.3 Ensemble Detectors

Figure 2 illustrates the ensemble pipeline for integrating detectors. We randomly split the validation dataset into two subsets, denoted as validation_1 and validation_2 and use them to determine the optimal ensemble combination.

First, we evaluate all trained detectors on validation_1 and rank them according to AUC. Based on this ranking, we select the top-k detectors for each feature type, specifically 30 attention-based detectors (AH) and 10 hidden-state-based detectors (HS).

Next, we perform greedy forward selection on validation_2. Starting from an empty set, we iteratively add the detector that yields the largest improvement in AUC. We stop when no further improvement is observed or when the number of detectors reaches 10. For aggregating the detectors, we adopt stacking ensemble based on preliminary experiments showing superior performance compared to other aggregation methods (Figure 2).

4 Experimental Setup

In this section, we present the experimental setup for evaluating EnsemHalDet. We compare to prior methods under both in-domain and cross-domain settings. In addition, we conduct ablation study to examine the contribution of different internal representations, the impact of ensemble strategies, and the effect of dimensionality reduction. We also evaluate computational efficiency.

4.1 Models, Datasets, and Evaluation Metrics

Models. We conduct experiments using three widely used VLMs: Llama-3.2-11B-Vision-Instruct (Llama-3.2-11B),[2] Qwen2.5-VL-7B-Instruct (Qwen2.5-VL-7B),[3] and Pixtral-12B-2409 (Pixtral-12B).[4] The architectural details of each model are summarized in Appendix A.

Datasets. As the evaluation dataset, we use CRAG-MM (Wang et al., 2025) and MMMU-Pro (Yue et al., 2025), both of which are VQA benchmarks providing images along with corresponding questions and answers. In this work, we follow the evaluation protocol of CRAG-MM, where the accuracy of generated answers is assessed through an LLM-as-a-Judge framework with GPT-4o-mini[5]. Specifically, the LLM-based judge compares each generated answer against a human-annotated reference to identify potential hallucinations (Prompt: Appendix B.1) and label a generated answer as hallucinated if it is judged to be incorrect compared to the ground truth. Although MMMU-Pro is inherently a multiple-choice benchmark, we reformat it into a free-form generation task and apply the same evaluation protocol as CRAG-MM.

When generating answers to VQA, we provide only image and question as the user prompt. The CRAG-MM dataset contains 3,874 samples, while MMMU-Pro consists of 1,730 samples. We adapt a stratified split for data splitting to preserve label distributions and divide the dataset into training, validation, and test sets with an 8:1:1 ratio.

Human Validation. To validate the reliability of the LLM-as-a-Judge evaluation, we additionally conducted a human validation study. Following the same setting as the LLM-based judge, we asked human annotators to determine whether each generated answer was correct or incorrect given the question, the ground-truth answer, and the generated answer. We then compared the human annotations with the LLM-as-a-judge predictions. The LLM judge achieved 85.4% accuracy, 72.7% precision, 100.0% recall, and 84.2% F1 score for hallucination detection (Table 6), suggesting that the LLM-based labels are largely consistent with human judgments. Details of the human validation procedure are provided in Appendix B.2.

Metrics. Hallucination detection is formulated as a binary classification task. The performance is measured using the area under the ROC curve (AUC), which measures the ability to distinguish between hallucinated and non-hallucinated responses. Detection performance is evaluated as the average over five runs of feature extraction with different random seeds.

We also measure feature extraction time and hallucination detection time. All experiments are conducted on a single NVIDIA RTX 6000 Ada GPU with a batch size of 1.

[2] https://huggingface.co/meta-llama/Llama-3.2-11B-Vision-Instruct

[3] https://huggingface.co/Qwen/Qwen2.5-VL-7B-Instruct

[4] https://huggingface.co/mistral-community/pixtral-12b

[5] https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

Methods	CRAG-MM (Wang et al., 2025)			MMMU-Pro (Yue et al., 2025)		
	Llama-3.2-11B	Qwen2.5-VL-7B	Pixtral-12B	Llama-3.2-11B	Qwen2.5-VL-7B	Pixtral-12B
SAPLMA (Azaria and Mitchell, 2023)	0.816 ± 0.028	0.793 ± 0.016	0.714 ± 0.039	0.794 ± 0.020	0.806 ± 0.031	0.814 ± 0.024
MIND (Su et al., 2024)	0.822 ± 0.020	0.837 ± 0.018	0.742 ± 0.013	0.803 ± 0.028	0.857 ± 0.029	0.806 ± 0.035
$\text{MHAD}_{\text{attn}}$ (Zhang et al., 2025a)	0.851 ± 0.018	0.865 ± 0.011	0.787 ± 0.011	0.800 ± 0.036	0.852 ± 0.017	0.824 ± 0.029
MHAD_{hs} (Zhang et al., 2025a)	0.841 ± 0.019	0.851 ± 0.008	0.759 ± 0.033	0.831 ± 0.022	0.847 ± 0.016	0.806 ± 0.025
MHAD_{ffn} (Zhang et al., 2025a)	0.841 ± 0.022	0.856 ± 0.013	0.776 ± 0.035	0.830 ± 0.021	0.851 ± 0.019	$\mathbf{0.834 \pm 0.035}$
$\text{EnsemHalDet}_{\text{AH}}$ (**Ours**)	0.843 ± 0.020	0.871 ± 0.021	0.800 ± 0.029	0.818 ± 0.033	0.862 ± 0.024	0.776 ± 0.029
$\text{EnsemHalDet}_{\text{HS}}$ (**ours**)	$\mathbf{0.860 \pm 0.013}$	0.864 ± 0.012	0.817 ± 0.013	0.831 ± 0.049	0.858 ± 0.036	0.768 ± 0.055
$\text{EnsemHalDet}_{\text{MIX}}$ (**ours**)	0.849 ± 0.021	$\mathbf{0.874 \pm 0.019}$	$\mathbf{0.818 \pm 0.021}$	$\mathbf{0.840 \pm 0.039}$	$\mathbf{0.869 \pm 0.019}$	0.781 ± 0.027

Table 1: Main comparison of hallucination detection methods with AUC. All results are reported as the mean and standard deviation over five runs of feature extraction with different random seeds. Best results are in **bold**.

4.2 Baselines.

We compare our method with three well-known hallucination detection approaches based on internal representations: SAPLMA, MIND, and MHAD all of which are applicable to open-source language models. We use the official implementations for SAPLMA and MIND. Since the implementation of MHAD is not publicly available, we re-implement it based on the descriptions in the original paper.

SAPLMA (Azaria and Mitchell, 2023). SAPLMA uses the hidden state of the last token from a specific transformer layer as input features. A 3-layer MLP is used as the classifier.

MIND (Su et al., 2024). MIND employs hidden states obtained by concatenating the mean-pooled representation of the final layer and the representation of the last token. A 4-layer MLP is used as the classifier.

MHAD (Zhang et al., 2025a). MHAD utilizes outputs from multiple internal components of the model, including attention block outputs (attn), hidden states (hs), and feed-forward networks (ffn). The first and last tokens from these components are extracted, and a 2-layer MLP is trained to select informative layers and neurons. The selected representations are concatenated and passed to a 4-layer MLP classifier.

4.3 In-Domain and Cross-Domain

To evaluate the generalization ability of hallucination detectors, we compare detection performance under both in-domain and cross-domain settings. In the in-domain setting, the detector is trained and validated on one dataset and evaluated on a held-out test split from the same dataset, whereas in the cross-domain setting, it is trained and validated on one dataset and evaluated on a held-out test split from a different dataset.

4.4 Ablation Study

We conduct an ablation study to analyze the contribution of each component in EnsemHalDet. Specifically, we evaluate the following aspects:

Feature Type. We compare attention-head-based features (AH), hidden-state-based features (HS), and their combination (MIX).

Combination Strategy. We compare different strategies for combining multiple detectors, including single-detector (top1), feature concatenation (concat), ensemble averaging (average), weighted averaging (weighted), and stacking (stack).

Dimensionality Reduction. For hidden state based features, we compare performance with and without dimensionality reduction (Appendix C).

5 Results

5.1 Comparison with Baselines

Table 1 shows the comparison of hallucination detection performance across different methods. All results are reported as the mean and standard deviation over five runs of feature extraction with different random seeds.

EnsemHalDet generally outperforms existing approaches across both datasets and all models. On CRAG-MM, EnsemHalDet outperforms all baselines across all three models, achieving the best AUC in each case. For Qwen2.5-VL-7B and Pixtral-12B, the method that combines both attention-head (AH) and hidden-state (HS) representations achieves the highest performance. In contrast, for Llama-3.2-11B, the HS-based methods achieves the best AUC. On MMMU-Pro,

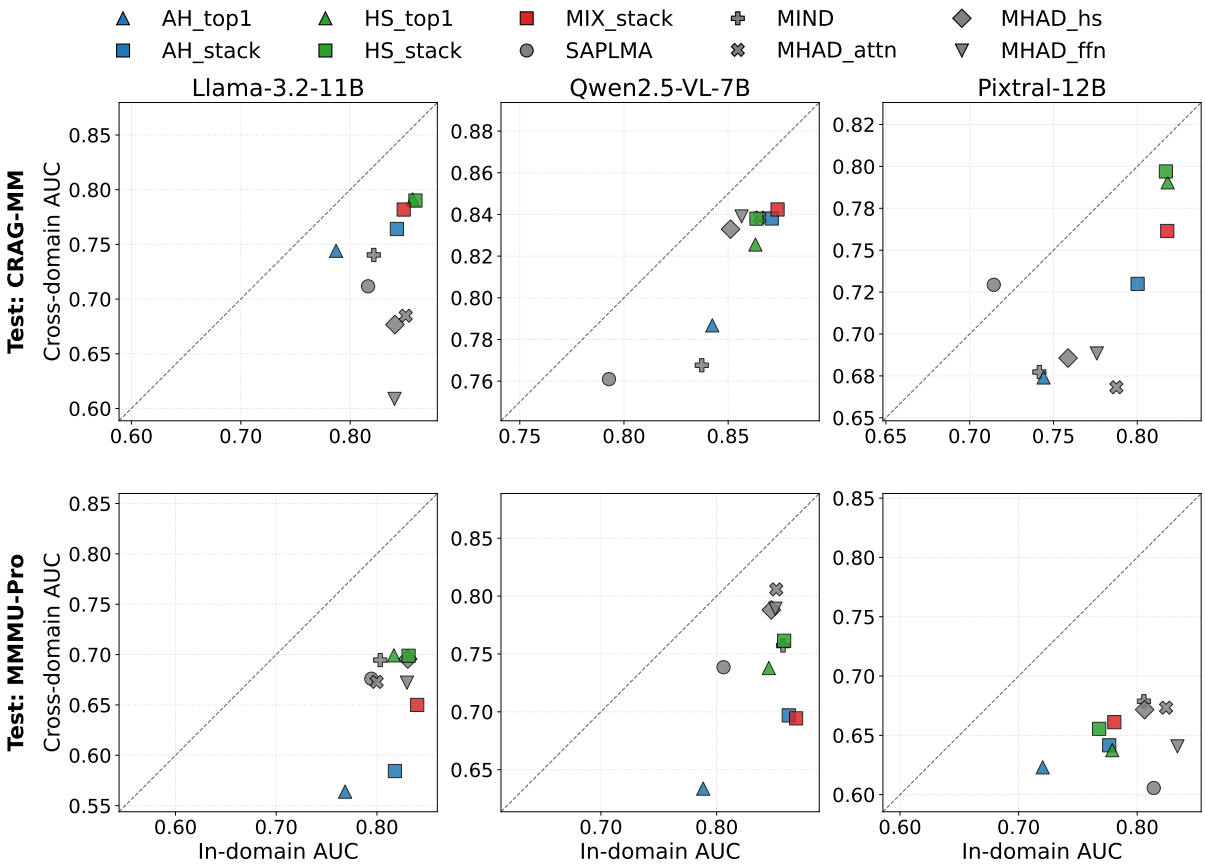


Figure 3: In-Domain and Cross-Domain performance with AUC. Each point represents a hallucination detector, with the x-axis showing in-domain AUC and the y-axis showing cross-domain AUC. Points closer to the upper-right corner indicate stronger performance in both settings, while points farther below the diagonal line indicate larger degradation under domain shift.

EnsemHalDet achieves the best performance on Llama-3.2-11B and Qwen2.5-VL-7B, while it does not outperform other baselines on Pixtral-12B.

These results indicate that EnsemHalDet generalizes well across different models and datasets. Although EnsemHalDet employs a simple logistic regression classifier, its performance suggests that effectively combining multiple internal representations improves robustness and enables accurate hallucination detection.

5.2 In-Domain and Cross-Domain

As shown in Figure 3, our methods outperform the baselines on CRAG-MM across all VLMs in both in-domain and cross-domain settings. In contrast, on MMMU-Pro, our methods outperform the baselines for Llama-3.2-11B in both settings, but underperform the strongest baselines in the cross-domain setting for Qwen2.5-VL-7B and Pixtral-12B. This suggests that the effectiveness of EnsemHalDet in cross-domain settings still depends on both the target dataset and the underlying VLM.

We further compare single-detectors (top1) with stacking ensembles. Across both in-domain and cross-domain settings, stacking generally improves performance over the corresponding top1 detector. This improvement is particularly pronounced for AH-based methods, indicating that attention-head outputs provide complementary signals that can be effectively exploited by ensemble learning.

5.3 Ablation Study

Table 2 presents the ablation study results of EnsemHalDet. All results are reported as the mean and standard deviation over five runs of feature extraction with different random seeds.

Effect of Feature Types. Regarding feature types, HS-based methods outperform AH-based methods in three settings: both datasets with Llama-3.2-11B and CRAG-MM with Pixtral-12B. In contrast, AH-based methods outperform HS-based methods for Qwen2.5-VL-7B on both datasets and for Pixtral-12B on MMMU-Pro, suggesting that

Variants	CRAG-MM			MMMU-Pro		
	Llama-3.2-11B	Qwen2.5-VL-7B	Pixtral-12B	Llama-3.2-11B	Qwen2.5-VL-7B	Pixtral-12B
Attention-based (AH)						
AH_{top1}	0.787 ± 0.022	0.842 ± 0.029	0.744 ± 0.046	0.768 ± 0.038	0.788 ± 0.046	0.720 ± 0.088
AH_{concat}	0.795 ± 0.031	0.833 ± 0.017	0.734 ± 0.029	0.797 ± 0.039	0.803 ± 0.050	0.736 ± 0.044
$AH_{average}$	0.843 ± 0.021	0.870 ± 0.021	0.801 ± 0.029	0.816 ± 0.033	0.871 ± 0.015	0.782 ± 0.030
$AH_{weighted}$	0.842 ± 0.021	0.871 ± 0.023	0.805 ± 0.026	0.815 ± 0.031	0.868 ± 0.020	0.786 ± 0.024
AH_{stack}	0.843 ± 0.020	0.871 ± 0.021	0.800 ± 0.029	0.818 ± 0.033	0.862 ± 0.024	0.776 ± 0.029
Hidden-state-based (HS)						
HS_{top1}	0.857 ± 0.015	0.863 ± 0.014	0.818 ± 0.024	0.817 ± 0.026	0.845 ± 0.019	0.779 ± 0.057
HS_{concat}	0.858 ± 0.011	0.862 ± 0.013	0.818 ± 0.015	0.840 ± 0.040	0.846 ± 0.030	0.771 ± 0.069
$HS_{average}$	0.858 ± 0.014	0.865 ± 0.012	0.816 ± 0.014	0.833 ± 0.047	0.859 ± 0.037	0.768 ± 0.055
$HS_{weighted}$	0.858 ± 0.014	0.865 ± 0.012	0.816 ± 0.014	0.828 ± 0.045	0.859 ± 0.037	0.768 ± 0.055
HS_{stack}	0.860 ± 0.013	0.864 ± 0.012	0.817 ± 0.013	0.831 ± 0.049	0.858 ± 0.036	0.768 ± 0.055
Mixed (AH + HS)						
MIX_{concat}	0.842 ± 0.023	0.864 ± 0.020	0.817 ± 0.016	0.843 ± 0.044	0.851 ± 0.030	0.789 ± 0.036
$MIX_{average}$	0.848 ± 0.022	0.873 ± 0.019	0.813 ± 0.024	0.838 ± 0.036	0.875 ± 0.018	0.785 ± 0.032
$MIX_{weighted}$	0.847 ± 0.023	0.873 ± 0.019	0.814 ± 0.026	0.831 ± 0.028	0.870 ± 0.016	0.767 ± 0.038
MIX_{stack}	0.849 ± 0.021	0.874 ± 0.019	0.818 ± 0.021	0.840 ± 0.039	0.869 ± 0.019	0.781 ± 0.027

Table 2: Ablation study of EnsemHalDet with AUC. We compare feature types (AH, HS, MIX) and combination strategies (top1, concatenation, average ensemble, weighted ensemble, and stacking).

Method	AUC	Gain	# Detectors	Error Overlap Gap	Correctness Disagreement
AH_{top1}	0.787	–	–	–	–
AH_{stack}	0.843	+0.056	5.200	0.088	0.108
HS_{top1}	0.857	–	–	–	–
HS_{stack}	0.860	+0.003	1.600	0.098	0.064
Mix_{stack}	0.849	-0.008	4.000	0.088	0.102

Table 3: Complementarity analysis of ensemble detectors with Llama-3.2-11B on CRAG-MM. Performance gain is the difference in detection performance between the ensemble and the corresponding single-detector. Ensemble size is the average number of detectors for ensemble. Error Overlap Gap measures deviation from independent error overlap, while Correctness Disagreement measures how often only one detector is correct.

attention-head features can also capture useful hallucination signals depending on the model. MIX-based methods achieve the best performance in four out of six settings and tie for the best performance in one additional setting. These results suggest that AH and HS features provide complementary information. While HS is generally a strong and stable standalone feature type, MIX offers the best overall performance by leveraging both types of internal representations.

Effect of Combination Strategies. Regarding combination strategies, ensemble-based methods generally outperform top1 and concat. The improvement is especially large for AH-based methods, suggesting that aggregating multiple attention-head detectors helps mitigate the noisiness of individual attention features. For HS-based methods, however, the gains from ensembling are relatively small, and HS_{top1} sometimes matches or even achieve better ensemble-based approach. Among ensemble strategies, averaging, weighted averaging, and stacking show only minor differences, suggesting that feature type is more important than the specific aggregation strategy.

This observation is consistent with the Correctness Disagreement reported in Table 3. The error overlap gap measures whether two detectors make errors on the same examples more often than expected under independence. A larger positive value indicates more shared failure modes, while a smaller value indicates more complementary errors. Correctness disagreement measures the proportion of examples where one detector is correct and the other is incorrect, capturing how often detectors can compensate for each other. AH-based detectors exhibit higher disagreement, suggesting that detectors trained on different attention-head outputs capture diverse hallucination patterns. This diversity increases the complementarity among detectors and makes ensembling particularly effective. By contrast, HS-based detectors show lower disagreement, indicating that different layers tend to

Method	Feature Extraction (s)	Detection (s)	Total (s)	Percentage (%)
Baseline	2.511	–	2.511	100.0
SAPLMA (Azaria and Mitchell, 2023)	3.767	0.000060	3.767	150.0
MIND (Su et al., 2024)	5.591	0.001465	5.592	222.7
$\text{MHAD}_{\text{attn}}$ (Zhang et al., 2025a)	3.095	0.001835	3.097	123.3
MHAD_{hs} (Zhang et al., 2025a)	2.952	0.002083	2.954	117.7
MHAD_{ffn} (Zhang et al., 2025a)	3.182	0.001325	3.183	126.8
$\text{EnsemHalDet}_{\text{AH}}$ (**Ours**)	3.095	0.000023	3.095	123.2
$\text{EnsemHalDet}_{\text{HS}}$ (**Ours**)	4.618	0.000054	4.618	183.9
$\text{EnsemHalDet}_{\text{MIX}}$ (**Ours**)	5.220	0.000120	5.220	207.9

Table 4: Inference time comparison for hallucination detection. The percentage indicates the total runtime relative to the baseline generation time (100%). We report GPU-based inference results for SAPLMA, MIND, and MHAD.

encode similar hallucination-related signals. Consequently, the additional benefit of ensembling HS-based detectors is limited compared with AH-based detectors.

5.4 Inference Time Analysis

Table 4 shows inference time of each method by decomposing it into feature extraction and hallucination detection. The feature extraction stage dominates the total runtime across all methods. This indicates that the computational bottleneck lies in extracting internal representations rather than in classifier inference. Our proposed method, EnsemHalDet, introduces additional overhead due to the use of multiple representations. Specifically, $\text{EnsemHalDet}_{\text{AH}}$ achieves a runtime comparable to $\text{MHAD}_{\text{attn}}$, while $\text{EnsemHalDet}_{\text{HS}}$ and $\text{EnsemHalDet}_{\text{MIX}}$ require longer processing time due to the aggregation of multiple features. This is mainly due to our design choice of extracting and aggregating representations from all generated tokens, whereas prior methods such as SAPLMA and MHAD only utilize limited tokens (e.g., the first and last tokens).

Despite the increased feature extraction cost, the detection stage of EnsemHalDet remains highly efficient. In particular, EnsemHalDet achieves significantly faster detection, being roughly 20–80× faster than MHAD, though EnsemHalDet does not always outperform MHAD in terms of detection accuracy. This is because our method employs a lightweight logistic regression classifier, whereas prior methods rely on MLPs.

These results highlight a trade-off between computational cost and detection performance. While EnsemHalDet introduces additional overhead in feature extraction, it maintains efficient detection and achieves superior performance, making it a practical approach for hallucination detection. We further observe that the AH- and HS-based ensemble variants already achieve sufficiently strong performance. Considering the additional cost of extracting multiple types of representations, using detectors trained on a single type of representation and ensembling them can provide a more efficient and practical alternative.

6 Conclusion

In this paper, we proposed EnsemHalDet, an ensemble-based hallucination detection framework for VLMs that ensembles multiple detectors trained on diverse internal representations. Experimental results on CRAG-MM and MMMU-Pro across multiple VLMs demonstrate that EnsemHalDet generally outperforms existing baselines under both in-domain and cross-domain settings. Through ablation studies, we showed that combining multiple detectors through ensemble learning generally improves detection performance over single-detectors. In particular, the improvement is most pronounced for AH-based detectors, indicating that attention-head representations provide complementary signals that can be effectively exploited by the ensemble. While our method achieves strong performance, it incurs additional computational cost due to the extraction of multiple internal states. Addressing this efficiency–performance trade-off remains an important direction for future work. In future work, we plan to explore more efficient feature selection strategies and investigate how internal representations capture hallucination signals.

Limitations

EnsemHalDet requires access to internal representations of VLMs, which limits its applicability to closed-source or API-only models. Although detector inference is lightweight, extracting multiple internal representations introduces additional computational overhead, making real-time hallucination detection still challenging. We also do not fully analyze what hallucination signals are captured by each internal representation or how detection performance varies across hallucination types. Furthermore, our labels rely on an LLM-as-a-Judge framework, and potential biases or errors in the judge may affect detector training and evaluation. Finally, while we conduct cross-domain evaluation by training and testing on different datasets, we do not explicitly control for the underlying domain distributions.

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*.

Xiang Chen, Chenxi Wang, Yida Xue, Ningyu Zhang, Xiaoyan Yang, Qiang Li, Yue Shen, Lei Liang, Jinjie Gu, and Huajun Chen. 2024a. Unified hallucination detection for multimodal large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. 2024b. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Yongxin Deng, Zhen Fang, Sharon Li, and Ling Chen. 2026. Beyond in-domain detection: Spikescore for cross-domain hallucination detection. In *Proceedings of the Fourteenth International Conference on Learning Representations*.

Thomas G. Dietterich. 2000. Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15. Springer Berlin Heidelberg.

Nuno M. Guerreiro, Elena Voita, and André Martins. 2023. 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*, pages 1059–1075.

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).

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

Zhangqi Jiang, Junkai Chen, Beier Zhu, Tingjin Luo, Yankun Shen, and Xu Yang. 2025. Devils in middle layers of large vision-language models: Interpreting, detecting and mitigating object hallucinations via attention lens. In *Proceedings of the Computer Vision and Pattern Recognition Conference*.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*.

Ludmila I. Kuncheva. 2004. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023a. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023b. Inference-time intervention: Eliciting truthful answers from a language model. In *Proceedings of the 37th Conference on Neural Information Processing Systems*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*.

Jiazhen Liu, Yuhan Fu, Ruobing Xie, Runquan Xie, Xingwu Sun, Fengzong Lian, Zhanhui Kang, and Xirong Li. 2025. Phd: A chatgpt-prompted visual hallucination evaluation dataset.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. 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*.

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

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100.

Sujoy Nath, Arkaprabha Basu, Sharanya Dasgupta, and Swagatam Das. 2025. Hallushift++: Bridging language and vision through internal representation shifts for hierarchical hallucinations in mllms. *arXiv preprint arXiv:2512.07687*.

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*.

Jiaqi Wang, Xiao Yang, Kai Sun, Parth Suresh, Sanat Sharma, Adam Czyzewski, Derek Andersen, Surya Appini, Arkav Banerjee, Sajal Choudhary, and 1 others. 2025. Crag-mm: Multi-modal multi-turn comprehensive rag benchmark. *arXiv preprint arXiv:2510.26160*.

Tianyun Yang, Ziniu Li, Juan Cao, and Chang Xu. 2025. Understanding and mitigating hallucination in large vision-language models via modular attribution and intervention. In *Proceedings of the 13h International Conference on Learning Representations*.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2024. A survey on multimodal large language models. *National Science Review*, 11(12):nwae403.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. 2025. MMMU-pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15134–15186.

Luan Zhang, Dandan Song, Zhijing Wu, Yuhang Tian, Changzhi Zhou, Jing Xu, Ziyi Yang, and Shuhao Zhang. 2025a. Detecting hallucination in large language models through deep internal representation analysis. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pages 8357–8365. Main Track.

Tianhang Zhang, Lin Qiu, Qipeng Guo, Cheng Deng, Yue Zhang, Zheng Zhang, Chenghu Zhou, Xinbing Wang, and Luoyi Fu. 2023. Enhancing uncertainty-based hallucination detection with stronger focus. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 915–932.

Yudong Zhang, Ruobing Xie, Xingwu Sun, Yiqing Huang, Jiansheng Chen, Zhanhui Kang, Di Wang, and Yu Wang. 2025b. Dhcp: Detecting hallucinations by cross-modal attention pattern in large vision-language models. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 3555–3564.

A VLMs architectures

Table 5 shows the architectures of each VLM that we used in the experiments. Llama-3.2-11B-Vision-Instruct integrates multimodal information by computing cross-attention between image and text representations at specific layers. In contrast, Qwen2.5-VL-7B-Instruct and Pixtral-12B-2409 project visual features into the same embedding space as text, treating image tokens equivalently to text tokens and integrating multimodal information via self-attention. The evaluated models exhibit relatively high hallucination rates, ranging from 79.5% to 83.7%. This is primarily attributed to the challenging nature of zero-shot VQA on the CRAG-MM benchmarks, particularly as we do not employ any system prompts to guide the model's behavior.

B LLM-as-a-Judge

B.1 LLM-as-a-Judge Prompt

Following the evaluation protocol of CRAG-MM (Wang et al., 2025), we use GPT-4o-mini as the evaluator to judge whether the generated answer is hallucinated. Using the same evaluator ensures consistency with prior work and allows for fair comparison across methods. The exact prompt used in our experiments is shown below.

```
You are an expert evaluator for question
answering systems.
Your task is to determine if a prediction correctly
answers a question based on the ground truth.

Rules:
1. The prediction is correct if it captures all the
key information from the ground truth.
2. The prediction is correct even if phrased
differently as long as the meaning is the same.
3. The prediction is incorrect if it contains
incorrect information or is missing essential
details.

Output a JSON object with a single field
'accuracy' whose value is true or false.
```

Figure 4: Prompt used for hallucination evaluation. We follow the CRAG-MM evaluation protocol.

B.2 Human Validation of LLM-as-a-Judge

To validate the reliability of our LLM-as-a-judge evaluation, we conducted a human evaluation on a subset of model responses. We asked eight annotators with graduate-level or higher backgrounds in information science and engineering, including both male and female participants, to participate in the annotation task. We randomly sampled 200 responses from the evaluation set, stratified by the LLM judge's binary hallucination labels: 100 responses predicted as hallucinated and 100 responses predicted as non-hallucinated. The sampled responses were divided into four subsets of 50 examples each. Each subset was independently annotated by two human annotators using Google Spreadsheets (Fig 5), following the annotation instructions shown in Fig 6. For each sample, annotators were shown the question, the ground-truth answer, and the model response, and were asked to judge whether the response contained a hallucination. The annotation took approximately 15 minutes per subset. We used a binary label scheme, where hallucination indicates that the response is factually incorrect, unsupported by the ground truth, or fails to correctly answer the question, while non-hallucination indicates that the response is consistent with the ground truth and correctly answers the question.

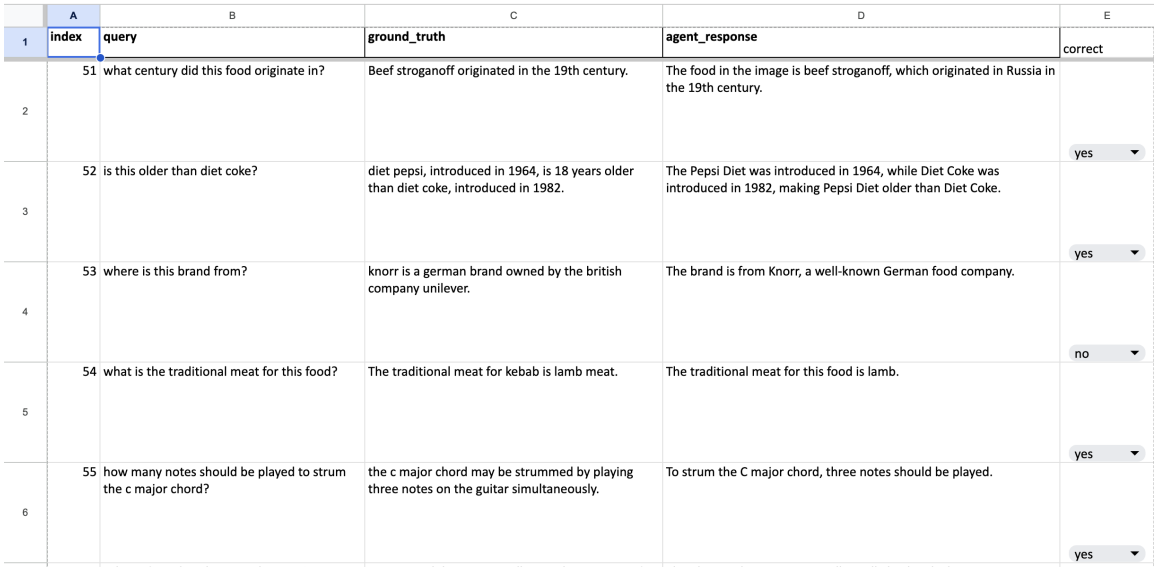
Among the 200 annotated examples, the two annotators agreed on 164 examples. On these agreed examples, the LLM judge achieved 85.4% accuracy, 72.7% precision, 100.0% recall, and 84.2% F1 score for hallucination detection (Table 6). The confusion matrix at Table 7 shows that the LLM judge did not miss any hallucinated responses in the agreed subset, but it classified 24 human-labeled non-hallucinated responses as hallucinations. This suggests that our LLM-as-a-judge evaluation is highly effective at detecting hallucinations, while tending to over-identify some correct responses as hallucinated.

C Effect of PCA

We further analyze the effect of PCA on hidden-state-based detectors in Table 8. Applying PCA consistently improves performance for most combination strategies, particularly for top1 and ensemble-based methods. This suggests that dimensionality reduction helps remove noise and improves the stability of learned detectors. In contrast, the effect on feature concatenation is less consistent, likely due to differences in how high-dimensional features are utilized.

Model	#Layers	#Heads	Hidden Dim	Multimodal Fusion	Hallucination Rate
Llama-3.2-11B-Vision-Instruct	40	32	4096	Cross-attention	0.795
Qwen2.5-VL-7B-Instruct	28	28	3584	Projector + Self-attention	0.837
Pixtral-12B-2409	40	32	4096	Projector + Self-attention	0.823

Table 5: Architectural details of VLMs, including hidden state dimensionality.



	A	B	C	D	E
1	index	query	ground_truth	agent_response	correct
2	51	what century did this food originate in?	Beef stroganoff originated in the 19th century.	The food in the image is beef stroganoff, which originated in Russia in the 19th century.	yes
3	52	is this older than diet coke?	diet pepsi, introduced in 1964, is 18 years older than diet coke, introduced in 1982.	The Pepsi Diet was introduced in 1964, while Diet Coke was introduced in 1982, making Pepsi Diet older than Diet Coke.	yes
4	53	where is this brand from?	knorr is a german brand owned by the british company unilever.	The brand is from Knorr, a well-known German food company.	no
5	54	what is the traditional meat for this food?	The traditional meat for kebab is lamb meat.	The traditional meat for this food is lamb.	yes
6	55	how many notes should be played to strum the c major chord?	the c major chord may be strummed by playing three notes on the guitar simultaneously.	To strum the C major chord, three notes should be played.	yes

Figure 5: Human evaluation interface on Google Spreadsheets

spreadsheet-link

Please judge whether the answer (`agent_response`) to the question (`query`) is correct by comparing it with the ground truth (`ground_truth`).
If you think the answer is correct, write `yes` in the `correct` column. If you think it is incorrect, write `no`.

【Instruction】
You are an expert evaluator for question answering systems.
Your task is to determine if a prediction correctly answers a question based on the ground truth.

Rules:
1. The prediction is correct if it captures all the key information from the ground truth.
2. The prediction is correct even if phrased differently as long as the meaning is the same.
3. The prediction is incorrect if it contains incorrect information or is missing essential details.

Figure 6: the instructions provided to the human annotators

Accuracy	Precision	Recall	F1
85.4	72.7	100.0	84.2

Table 6: Human validation of the LLM-as-a-judge for binary hallucination detection. Precision, recall, and F1 are computed with hallucination as the positive class.

D Ablation of Inference Time

Table 9 presents a detailed breakdown of inference time across different methods.

We compared CPU-based and GPU-based inference for prior methods. Across SAPLMA, MIND, and MHAD, the difference between CPU and GPU execution is minimal, as the detection stage accounts for only a small fraction of the total runtime. This indicates that accelerating the classifier alone provides limited benefit, since the overall cost is dominated by feature extraction.

	LLM-as-a-judge	
Human label	Correct	Hallucination
Correct	76	24
Hallucination	0	64

Table 7: Confusion matrix between agreed human labels and LLM-as-a-judge predictions.

Methods	CRAG-MM			MMMU-Pro		
	Llama-3.2-11B	Qwen2.5-VL-7B	Pixtral-12B	Llama-3.2-11B	Qwen2.5-VL-7B	Pixtral-12B
HS_{top1} (w/ PCA)	0.857	0.863	0.818	0.817	0.845	0.779
HS_{top1} (w/o PCA)	0.755	0.775	0.721	0.728	0.728	0.723
HS_{stack} (w/ PCA)	0.860	0.864	0.817	0.831	0.858	0.768
HS_{stack} (w/o PCA)	0.774	0.782	0.750	0.763	0.756	0.735

Table 8: Ablation study on the effect of PCA for hidden-state-based detectors. Applying PCA consistently improves performance for top1, average ensemble, weighted ensemble, and stacking, while the effect on concatenation is mixed.

Method	Feature Extraction (s)	Detection (s)	Total (s)	Percentage (%)
Baseline	2.511	–	2.511	100.0
SAPLMA (CPU)	3.767	0.000219	3.767	150.0
SAPLMA (GPU)	3.767	0.000060	3.767	150.0
MIND (CPU)	5.591	0.001465	5.592	222.7
MIND (GPU)	5.591	0.001430	5.592	222.7
$MHAD_{attn}$ (CPU)	3.095	0.001835	3.097	123.3
$MHAD_{attn}$ (GPU)	3.095	0.001309	3.096	123.3
$MHAD_{hs}$ (CPU)	2.952	0.002083	2.954	117.7
$MHAD_{hs}$ (GPU)	2.952	0.001412	2.954	117.7
$MHAD_{ffn}$ (CPU)	3.182	0.001325	3.183	126.8
$MHAD_{ffn}$ (GPU)	3.182	0.001465	3.184	126.8
$EnsemHalDet_{AH}$ (ours)	3.095	0.000023	3.095	123.2
$EnsemHalDet_{HS}$ (ours)	4.618	0.000054	4.618	183.9
$EnsemHalDet_{MIX}$ (ours)	5.220	0.000120	5.220	207.9

Table 9: Detailed runtime breakdown. Feature extraction corresponds to internal representation extraction, while hallucination detection denotes classifier inference time.