OSR: DETECTING HALLUCINATION BY MLLMS' OBJECT-LEVEL SELF-REFLECTION MECHANISM

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite the important role of multimodal large language models (MLLMs) in many domains, hallucinations that occur during answer generation remain a non-negligible problem. Existing methods proposed in the research community are mostly constrained by the capability of the models' visual modules. To address this issue, we propose a novel hallucination detection method OSR based on the **O**bject-level **S**elf-**R**eflection mechanism of MLLMs. Specifically, OSR decouples object recognition and relational reasoning, which are originally performed simultaneously within a single inference. OSR first leverages a chain-of-thought approach to guide the model in object recognition, then conducts object-level self-reflection on image understanding based on the recognition results, and finally generates the answer. Furthermore, when computing the semantic consistency of answers, we are inspired by the von Mises–Fisher (vMF) distribution and replace the conventional entropy-based metric with the mean resultant vector length, which exhibits greater stability when the sample size is small. We conduct extensive experiments and ablation studies across multiple models and diverse datasets to demonstrate the effectiveness of OSR.

1 INTRODUCTION

Although recent years multi-modal large language models (MLLMs) have achieved remarkable progress and shown strong performance on tasks such as visual question answering and multi-modal reasoning, they may still produce content that is inconsistent with facts or the given input when generating answers, a phenomenon referred to as hallucination. Such hallucinations undermine the reliability of model outputs and may pose serious safety risks in high-stakes domains, including medical diagnosis, autonomous systems, legal reasoning, scientific discovery, and educational guidance. As MLLMs are increasingly integrated into real-world applications, ensuring their reliability and factual consistency is essential. Therefore, mitigating hallucinations has become a critical challenge for their safe deployment.

To detect hallucinations, various methods have been proposed in the research community, which can be broadly categorized into white-box and black-box approaches. White-box methods typically rely on access to the model's internal states, such as analyzing attention weights, hidden activations, or gradient information. While these methods can leverage rich internal signals, they usually require access to the model architecture, which makes them infeasible for closed-source models. In contrast, black-box methods do not require access to the model's internal mechanisms. They evaluate hallucinations based on observable outputs, for example, the generated responses or their consistency with external knowledge bases. Although black-box methods offer greater flexibility in deployment, their detection accuracy is often limited by the reliability of the model outputs themselves. However, current methods are both limited by the capabilities of the model's visual module. Hallucinations occur when the model fails to accurately recognize objects and understand the relationships between them simultaneously.

Actually, the presence of this kind of hallucination does not necessarily imply that the model completely lacks the ability to produce correct answers. When hallucinations arise from the tight coupling between object recognition and relational understanding in the visual reasoning process, we first decouple these two tasks into sequential steps. The model is prompted to perform object recognition first, and then to conduct self-reflection on its preliminary understanding of the image based

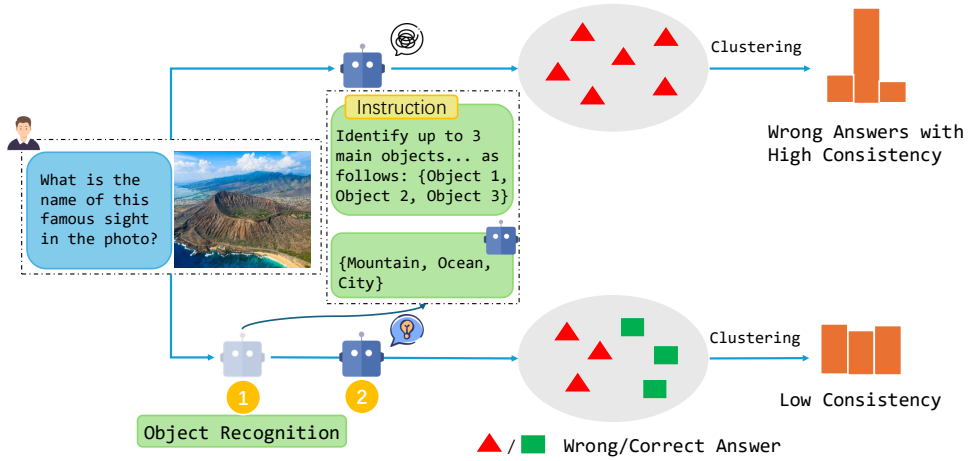

Figure 1: A comparison of two methods for eliciting responses from a model. When directly queried, the model may produce only similar hallucinated answers, resulting in high semantic consistency. However, by decoupling reasoning and allowing the model to answer questions in two steps, it becomes possible to generate correct answers, thereby reducing semantic consistency.

on the recognition results. This capacity for self-reflection requires no structural modifications to the model and can be elicited solely through simple prompting. Specifically, by adopting step-by-step reasoning paradigms such as Chain-of-Thought (CoT), the model is able to perform complex visual reasoning tasks through its inherent self-reflection ability, thereby enhancing the accuracy and reliability of its responses. This approach aligns closely with the CoT methodology widely embraced in the large language model community, proving particularly suitable for mitigating hallucinations caused by object recognition biases.

In this work, we design a novel hallucination detection method OSR based on the **O**bject-level **S**elf-**R**eflection mechanism of MLLMs and demonstrate its effectiveness. The core pipeline consists of two stages. In the first stage, we sample the same input multiple times to collect both the model's original responses and its self-reflective responses guided by a two-step CoT prompt. In the second stage, we assess hallucination by measuring the semantic consistency between these two types of responses. Unlike traditional approaches that rely on entropy-based measures, we are inspired by the von Mises–Fisher (vMF) distribution and propose to quantify semantic consistency via the mean resultant length of the response embeddings. This metric is not only simple and efficient but also overcomes the drawback of entropy, which tends to fluctuate when the sample size is small.

Our main contributions are summarized as follows:

- We find that MLLMs can be prompted to exhibit object-level self-reflection capability and systematically apply this property to hallucination detection for the first time, leading to a novel detection method OSR.
- We introduce a semantic consistency metric inspired by vMF. This train-free method is model-agnostic, which means it can be seamlessly integrated as a plug-and-play module into existing hallucination detection pipelines.
- We conduct extensive experiments. Results show that our method achieves state-of-the-art or competitive performance across most scenarios. We further carry out comprehensive ablation studies to validate its effectiveness.

2 Related Work

Hallucination Detection Methods. With the widespread application of MLLMs, the hallucination problem they generate has become a critical obstacle hindering their reliable deployment. Accordingly, techniques for detecting hallucinations have also been extensively studied. Existing methods can be broadly categorized into three classes: 1) methods based on internal model signals; 2) methods based on external knowledge; and 3) methods based on consistency.

The core idea of the first category is to detect uncertainty by analyzing the model's own outputs or internal states without relying on external resources, and to use this as a proxy indicator for hallucinations. The second category of methods verifies the factuality of model-generated content by introducing external knowledge sources as objective ground truths. These methods are generally more reliable but come with higher computational costs and depend on the completeness and accuracy of external knowledge. The third category of methods detects hallucinations by comparing whether outputs from multiple different sources or perspectives are consistent. The underlying assumption is that if multiple independent judgments agree, the result is more trustworthy.

Introspective Chain-of-Thought. CoT prompting guides large language models (LLMs) to generate a series of intermediate reasoning steps, which significantly improves their performance on complex reasoning tasks. However, standard CoT follows a "one-way generation" process: once the model produces an incorrect premise or reasoning step, it inevitably leads to an incorrect final answer, while lacking mechanisms for self-verification and correction.

To overcome this limitation, researchers have proposed Introspective Chain-of-Thought or Self-Reflection reasoning methods. The core idea of these approaches is to enable the model to critically examine and verify its own reasoning process, either before or after producing the final answer, so as to identify and correct potential errors. Depending on their implementation and the timing of reflection, existing work can be broadly divided into two categories: 1) post-hoc verification and correction; and 2) process-integrated self-questioning.

The first category follows a "generate–verify–revise" paradigm: the model first generates an initial reasoning chain and answer, which is then reviewed by an independent verifier module (either the same model or a different one). The verifier detects logical flaws, factual inconsistencies, or calculation mistakes in the reasoning process, and the initial answer is subsequently revised based on this review. The second category integrates reflection directly into the reasoning process itself rather than performing it afterwards. At each reasoning step, the model engages in self-questioning, thereby avoiding errors before they propagate.

3 Method

3.1 Problem Statement

This work focuses on assessing the reliability of model responses in a black-box setting, without relying on internal parameters or external knowledge bases. In this paper, we denote the i-th sample in the dataset as $p_i = \{I_i, t_i\}$, since it consists of an image I_i and a task t_i. We define the model's response as $y = \mathrm{MLLM}(x)$, where MLLM denotes the multi-modal large language model mapping from input x to output y. For sample p_i, we first disable sampling to obtain a single deterministic answer $A_i = \mathrm{MLLM}(p_i)$, which will be used for the final comparison against the ground truth. Next, we enable sampling and query the same input K times, which is similar to the process in Eigenscore, obtaining an answer set

$$s_{ori} = \{a_1, a_2, \ldots, a_K\}, \tag{1}$$

where $a_i = \mathrm{MLLM}(p_i)$.

Traditionally, we can assess the consistency of answers in s_{ori} to determine whether the model's response is correct. However, due to the model's potential limitations in understanding the inputs, it may repeatedly generate the same incorrect answer, leading to a situation where the consistency within s_{ori} is high while the response is actually wrong. To address this issue, we further introduce an object-level self-reflection mechanism, prompting the model to answer under more specific instructions to get another answer set s_{ref}, thereby increasing the diversity of the answer set and forming a contrastive basis for evaluation.

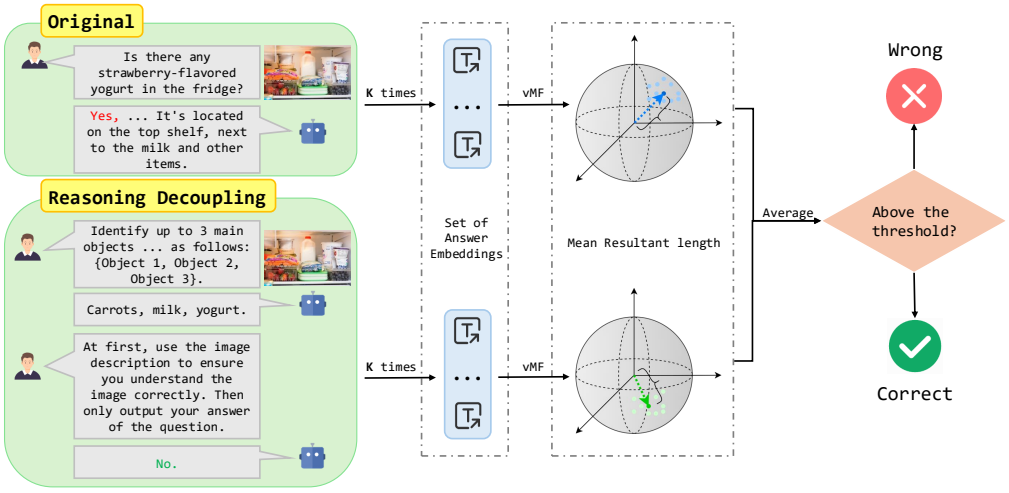

Figure 2: Overall illustration of our proposed method. For each sample, the pipeline obtains two sets of answers, calculates the mean resultant vector length using the vMF distribution for each set, then takes their average, and finally compares it against a fixed threshold.

After getting s_{ori} and s_{ref}, we evaluate the semantic consistency metric for each set, compute their average, and then compare the resulting value against a fixed threshold τ to assess whether the model's response is accurate.

3.2 Object-Level Self-Reflection by CoT

We implement object-level self-reflection using a two-stage CoT prompting scheme to generate a set of self-reflective responses s_{ref}.

Specifically, we begin with an object recognition prompt p_{obj} that instructs the model to identify the salient visual entities in the image. This step is crucial because objects serve as the fundamental building blocks of visual semantics which provide the primary cues for interpreting what the scene contains and how it should be understood. Moreover, the relationships among these prominent objects largely determine the overall meaning of the image, influencing subsequent reasoning about actions, attributes, and higher-level contextual understanding. Therefore, establishing an accurate inventory of objects through p_{obj} forms the foundation upon which later stages of semantic analysis and reasoning can be reliably built. The full content of p_{obj} is *"Identify up to 3 main objects in the image. Note: do not use abbreviations for object names, please provide the full name. Format your output exactly as follows: {Object 1, Object 2, Object 3}."* Mathematically, the recognition result r_i is obtained as follows

$$r_i = \text{MLLM}(\{I_i, p_{obj}\}). \tag{2}$$

Then, we introduce a reflection prompt p_{ref}, which guides the model to re-examine the image with the aid of the previously recognized objects and to subsequently generate an answer. The purpose of this step is to encourage the model to reflect on its initial understanding of the scene, validate it against the object-level information, and thereby reduce the likelihood of spurious or hallucinated responses. In other words, p_{ref} serves as a mechanism for self-reflection, prompting the model to explicitly confirm its comprehension before committing to an answer. The full content of p_{ref} is given as *"At first, use the image description to ensure you understand the image correctly. Then only output your answer of the question."*, which enforces a two-stage reasoning process—first grounding the answer in visual semantics, and then producing the final response.With the reflection prompt p_{ref}, we prompt the model to generate K responses to form the reflection answer set s_{ref} as below

$$s_{ref} = \{b_1, b_2, \ldots, b_K\}, \tag{3}$$

where $b_i = \text{MLLM}(\{I_i, q_i || p_{ref} || r_i\})$.

3.3 vMF-Based Semantic Consistency

An important indicator of hallucination is the divergence among responses: if the model's multiple answers are dispersed in the semantic space, it usually implies uncertainty in its understanding of the question and thus a higher risk of hallucination. Inspired by the vMF distribution, we adopt a geometric approach to quantify this consistency. The probability density function of the vMF distribution is

$$p = (x|\mu, \kappa) = C_d(\kappa) \exp(\kappa \mu^T x), \tag{4}$$

where $x \in \mathbb{R}^d$ is a unit vector, $\mu \in \mathbb{R}^d$ is the mean direction with $\|\mu\| = 1$, $\kappa \geq 0$ is the concentration parameter, and $C_d(\kappa)$ is the normalization constant in d dimensions.

On a unit spherical surface, if the response vectors follow a vMF distribution, their concentration is determined by the parameter κ. However, the range of κ is large, making it difficult to set an accurate threshold. To solve this, directional statistics theory indicates that the sample mean resultant length $\bar{R} = \frac{1}{K}\| \sum_{i=1}^{K} \hat{x_i}\|$ is an unbiased estimator of κ, satisfying

$$\bar{R} = A_d(\kappa) = \frac{I_{d/2}(\kappa)}{I_{d/2-1}(\kappa)}, \tag{5}$$

where $A_d(\kappa)$ denotes the ratio of modified Bessel functions that characterizes the mean resultant length in d-dimensional space, I_ν is the modified Bessel function of the first kind.

Therefore, $\bar{R}$ can be directly used to measure the concentration of the response distribution. Furthermore, a larger $\bar{R}$ is equivalent to a larger κ in the vMF distribution, indicating higher response consistency and lower hallucination risk. To compute $\bar{R}$, we utilize the nli-roberta-large model to convert both answer sets into embedded vector sets es_{ori} and es_{ref}, respectively. The specific conversion process is expressed as

$$es_{ori} = \{ea_1, ea_2, \ldots, ea_K\}, \quad es_{ref} = \{eb_1, eb_2, \ldots, eb_K\}, \tag{6}$$

where $ea_i = \mathrm{nrl}(a_i)$ and $eb_i = \mathrm{nrl}(b_i)$.

Because the vMF distribution is defined over directional data, i.e., points lying on the surface of a unit hypersphere, it is necessary to normalize the vectors so that they have unit length. This normalization removes the influence of magnitude, preserves only directional information, and enables the vMF distribution to capture their angular relationships on the hypersphere. Hence, we subsequently normalize es_{ori} and es_{ref} onto a unit hypersphere like

$$\hat{es}_{ori} = \{\hat{ea}_i\}_{i=1}^{K} = \{\frac{ea_i}{\|ea_i\|}\}_{i=1}^{K}, \quad \hat{es}_{ref} = \{\hat{eb}_i\}_{i=1}^{K} = \{\frac{eb_i}{\|eb_i\|}\}_{i=1}^{K}. \tag{7}$$

After normalizing the embedded vectors onto the unit hypersphere, we can compute $\bar{R}_{\hat{es}_{ori}}$ and $\bar{R}_{\hat{es}_{ref}}$ on $\hat{es}_{ori}$ and $\hat{es}_{ref}$, where

$$\bar{R}_{\hat{es}_{ori}} = \frac{1}{K}\| \sum_{i=1}^{K} \hat{ea}_i\|, \quad \bar{R}_{\hat{es}_{ref}} = \frac{1}{K}\| \sum_{i=1}^{K} \hat{eb}_i\|. \tag{8}$$

Lastly, we take the average of $\bar{R}_{\hat{es}_{ori}}$ and $\bar{R}_{\hat{es}_{ref}}$ to get

$$\bar{R}_{avg} = \frac{1}{2}(\bar{R}_{\hat{es}_{ori}} + \bar{R}_{\hat{es}_{ref}}). \tag{9}$$

3.4 Hallucination Detection Criterion

We determine the presence of hallucination by comparing the averaged consistency score $\bar{R}$avg with a pre-defined threshold τ. Formally, the hallucination flag is given by

$$flag = \begin{cases} 1, & \text{if } \bar{R}_{\text{avg}} > \tau, \\ 0, & \text{if } \bar{R}_{\text{avg}} \leq \tau, \end{cases} \tag{10}$$

where $flag = 1$ denotes that the model's responses exhibit sufficiently high internal consistency and are therefore judged as non-hallucinatory, while $flag = 0$ denotes low consistency and is treated

as a hallucination case. This criterion operationalizes the intuition that reliable answers should be directionally aligned across multiple reasoning paths, whereas hallucinations typically result in more divergent representations.

Once the hallucination flag for a sample is determined, we evaluate the detection performance by comparing the predicted answer A_i against its ground-truth label gt_i. Specifically, we check whether the correctness of A_i matches the predicted hallucination flag. The detection accuracy is then defined as

$$Accuracy = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\left[\mathbb{I}[A_i = gt_i] = flag\right], \tag{11}$$

where $\mathbb{I}[\cdot]$ denotes the indicator function and N is the total number of evaluation samples. Generally, a detection is counted as correct if the hallucination flag aligns with the actual correctness of the answer, and the overall accuracy measures the proportion of such correct detections across the dataset.

4 Experiment

4.1 Experimental Setup

Datasets. To ensure a fair and direct comparison with the baseline, we evaluate on widely adopted datasets categorized into free-form and multiple-choice formats. For free-form evaluation, we use MM-Vet which is a comprehensive benchmark with 218 open-ended questions assessing integrated visual-language abilities and LLaVA-Bench, which contains 60 image-question pairs probing higher-level competencies like reasoning and contextual inference. For multiple-choice evaluation, we employ MMMU, a challenging college-level dataset with 11.5K questions across 30 subjects, and ScienceQA, which includes 21.2K structured science questions from elementary and high school curricula to measure factual knowledge and systematic reasoning.

Models. In addition to the datasets, the MLLMs we evaluated are also identical to those used in the baseline. We evaluate ten prominent multi-modal models from four architectural families, consistent with the baseline. The Qwen2VL series (2B, 7B, 72B) employs an end-to-end integrated vision-language architecture for strong generalization. The InternVL2 series (1B, 8B, 26B) focuses on high-resolution visual encoding and cross-modal alignment. The LLaVA-1.5 series (7B, 13B) uses a simple linear projection between CLIP and a language model, offering a balance of simplicity and effectiveness. The LLaVA-NeXT series (7B, 13B) introduces improvements such as higher resolution and optimized training. This selection enables a structured comparison across diverse model paradigms and scales.

Baselines. We compare our method with some several representative baselines GAVIE, Semantic Entropy, VL-Uncertainty and EigenScore. These baselines cover a wide range of mainstream hallucination detection paradigms, including entropy-based, sampling-based, and metric-based approaches, ensuring a comprehensive evaluation.

Implementation Details. The parameter configuration for all models in this experiment is set to *temperature* = 0.1 when using the greedy strategy, and to *temperature* = 0.5, *top-k* = 10, *top-p* = 0.99 when generating responses with the sampling strategy. For the sampling time K, we set it to 10. The threshold τ we use to measure the confidence of responses is 0.48. All experiments were conducted on multiple NVIDIA GeForce RTX 3090 GPUs.

4.2 Main Results

Based on the experimental results presented in the Table 1, the proposed method demonstrates competitive performance across multiple multimodal benchmarks when compared to state-of-the-art baselines.

Firstly, we analyze the results on free-form datasets. For the LLaVABench dataset, our method achieves an average accuracy of 64.50%, ranking second overall while slightly outperforming EigenScore's 63.83%. Notably, our approach achieves the best performance of 73.33% when using the InternVL2-1B model, showing particularly strong adaptation to smaller-scale models. The MM-Vet

Table 1: Comparison between our method and other state-of-the-art baselines. Q, I, L, and LN denote Qwen2VL, InternVL2, LLaVA-1.5, and LLaVA-NeXT, respectively. The reported results are detection accuracies in percentage. Bold numbers indicate the best performance, while underlined numbers represent the second best. As shown, our method achieves either the best or the second-best performance in most scenarios, demonstrating its superior effectiveness.

Datasets	Methods \ Models	Q2B	Q7B	Q72B	I1B	I8B	I26B	L7B	L13B	LN7B	LN13B	Avg
LLaVABench	GAVIE (2024)	25.00	26.67	40.00	30.00	31.67	31.67	15.00	20.00	45.00	35.00	30.00
	Semantic Entropy (2024)	61.67	55.00	61.67	65.00	60.00	53.33	70.00	**70.00**	61.67	**65.00**	62.33
	KLE (2024)	28.33	48.33	58.33	40.00	46.67	50.00	23.33	45.00	43.33	33.33	41.67
	EigenScore (2024)	63.33	**58.33**	**63.33**	63.33	55.00	**61.67**	68.33	**70.00**	**70.00**	**65.00**	63.83
	VL-Uncertainty (2024)	55.67	53.33	53.33	60.00	60.00	51.67	73.33	63.33	61.67	61.67	59.40
	Ours	**68.33**	**58.33**	53.33	**73.33**	**61.67**	55.00	**80.00**	66.67	63.33	**65.00**	**64.50**
MMVet	GAVIE (2024)	29.36	43.58	51.38	30.73	30.73	22.48	23.39	24.77	37.61	43.58	33.76
	Semantic Entropy (2024)	60.55	57.80	62.84	72.94	55.05	58.72	72.48	79.36	61.01	72.48	65.32
	KLE (2024)	45.41	51.83	56.88	42.20	46.79	51.38	41.74	41.28	41.74	42.66	46.19
	EigenScore (2024)	**73.85**	70.64	**72.48**	77.98	67.43	**76.61**	73.85	78.44	73.85	77.52	74.27
	VL-Uncertainty (2025)	64.22	67.43	66.97	65.60	62.39	64.67	**79.35**	**80.28**	66.06	69.72	68.67
	Ours	**73.85**	**72.94**	71.10	**78.44**	**72.02**	75.23	76.61	77.98	**76.15**	**77.98**	**75.23**
MMMU	GAVIE (2024)	37.82	48.36	57.09	40.61	48.12	33.21	37.58	44.61	43.64	45.82	43.69
	Semantic Entropy (2024)	53.82	54.91	60.36	53.82	54.91	52.48	52.61	50.18	52.61	50.18	53.59
	KLE (2024)	43.88	53.33	62.91	46.42	49.58	59.52	51.03	45.33	47.39	51.52	51.09
	EigenScore (2024)	51.43	64.48	67.64	51.15	**59.39**	56.24	**59.03**	54.42	55.39	49.21	56.84
	VL-Uncertainty (2025)	57.33	58.55	65.94	55.15	57.33	57.21	56.36	55.15	57.58	**56.24**	57.68
	Ours	**60.84**	**64.61**	**68.72**	**57.58**	**59.39**	**59.64**	58.55	**56.48**	**60.85**	55.52	**60.22**
ScienceQA	GAVIE (2024)	61.82	77.09	85.23	53.94	86.71	89.19	58.5	66.39	62.27	65.2	70.63
	Semantic Entropy (2024)	54.04	77.94	87.06	64.45	**90.08**	91.32	61.77	68.02	67.67	65.34	72.77
	KLE (2024)	62.22	76.45	86.91	67.63	89.64	90.43	60.73	67.97	65.94	66.23	73.42
	EigenScore (2024)	62.57	72.73	84.63	62.22	57.51	87.31	64.15	65.64	70.80	64.95	69.25
	VL-Uncertainty (2025)	**66.83**	**80.71**	88.60	64.50	89.54	**91.57**	65.79	68.57	68.67	**67.67**	75.25
	Ours	65.00	79.87	**88.80**	**68.32**	**90.08**	91.37	**66.48**	**72.43**	**70.95**	67.43	**76.07**

dataset results show our method attaining the highest average accuracy of 75.23%, significantly surpassing EigenScore's 74.27%. The method maintains consistent leadership across the InternVL2 model series, achieving 78.44%, 72.02%, and 75.23% on the I1B, I8B, and I26B models respectively, demonstrating effective scalability across different model sizes.

Then we pay attention to the results on multiple-choice datasets. For the MMMU dataset, our method achieves a notable average accuracy of 60.22%, exceeding VL-Uncertainty's 57.68% by a considerable margin of 2.54 percentage points. The method reaches its peak performance of 68.72% using the Qwen2VL-72B model, highlighting its advantage with larger-scale models. On the ScienceQA dataset, our method achieves the highest average accuracy of 76.07%, outperforming VL-Uncertainty's 75.25%. The approach matches the best performance of 90.08% when using the InternVL2-8B model, demonstrating strong capability in scientific question answering tasks.

Overall, the proposed method achieves the best average performance on three out of four benchmarks and ranks second on the remaining one. The consistent performance across various model scales, ranging from 2B to 26B parameters, demonstrates the method's robustness and generalization capability. The particularly strong results on complex reasoning benchmarks such as MM-Vet and MMMU indicate that our approach effectively enhances multimodal models' uncertainty estimation and quality assessment abilities across diverse task requirements.

In summary, our method not only surpasses strong baselines with large margins on datasets like LLaVABench and MM-Vet, but also maintains stable competitiveness on more difficult benchmarks such as MMMU and ScienceQA, demonstrating its superior generalization and robustness.

Table 2: Ablation study of our method on two datasets by removing two key components. We compare the full method with two variants: without object-level self-reflection by CoT (ours w/o CoT) and without vMF-based semantic consistency (ours w/o vMF). The reported numbers are detection accuracies in percentage. Bold numbers indicate results superior to the baseline methods.

Datasets	Methods \ Models	Q2B	Q7B	Q72B	I1B	I8B	I26B	L7B	L13B	LN7B	LN13B	Avg
MMMU	Ours	60.84	**64.61**	**68.72**	57.58	59.39	59.64	58.55	56.48	**60.85**	55.52	**60.22**
	Ours w/o CoT	**61.94**	62.18	66.30	55.27	59.64	**59.88**	59.27	57.94	59.88	55.39	59.77
	Ours w/o vMF	60.96	59.35	67.15	**59.61**	**60.73**	55.39	**59.76**	**59.96**	58.84	**58.38**	60.01
	Ours w/o CoT,vMF	53.82	54.91	60.36	53.82	54.91	52.48	52.61	50.18	52.61	50.18	53.59
ScienceQA	Ours	65.00	**79.87**	**88.80**	**68.32**	**90.08**	91.37	66.48	**72.43**	70.95	67.43	**76.07**
	Ours w/o CoT	65.99	79.03	88.00	68.12	89.99	**91.52**	**66.98**	70.9	**71.15**	67.58	75.93
	Ours w/o vMF	**66.21**	65.47	61.63	63.99	69.33	81.11	63.91	64.71	66.04	**68.57**	67.10
	Ours w/o CoT,vMF	54.04	77.94	87.06	64.45	**90.08**	91.32	61.77	68.02	67.67	65.34	72.77

4.3 Ablation Studies

Presence of Two Key Components. First, since the sample sizes of the LLaVABench and MM-Vet datasets are too small, we conduct ablation studies on the MMMU and ScienceQA datasets to avoid the randomness affecting our conclusions. Table 2 presents the ablation results of our method on these two datasets, where two key components are removed: object-level self-reflection and vMF-based semantic consistency.

We can observe that regardless of which component is removed, although there may be improvements in certain test scenarios, the overall performance is negatively affected. Moreover, the method without any components performs significantly worse than the complete approach.

Sensitivity to Hyperparameters. The hyperparameters used during answer generation, including temperature, maximum generation length, top-k, and top-p, play a crucial role in shaping the model's responses. To systematically evaluate their influence, we carried out a sensitivity analysis with the Qwen2VL-2B model on the MM-Vet benchmark. As summarized in Figure 3, the model exhibits distinct behaviors under different hyperparameter settings. As shown in Figure 3(a), performance remains largely consistent across a wide range of temperature values, suggesting that the model is robust to variations in this parameter. Illustrated in Figure 3(b), accuracy shows minimal fluctuation as the maximum generation length changes, indicating that the model is not highly sensitive to this constraint within the tested range. Results in Figure 3(c) demonstrate that modifying the top-k value has limited impact on performance, further underscoring the stability of the model under different sampling breadths. In contrast, Figure 3(d) reveals a clear positive correlation between top-p values and accuracy. Performance improves consistently as top-p increases, highlighting a substantial influence of this parameter on output quality.

In short, while temperature, maximum generation length, and top-k introduce only marginal variations, top-p stands out as the most impactful hyperparameter, with higher values leading to significantly better accuracy.

5 Conclusion

Accurately detecting hallucinations is a crucial safeguard for the safe use of MLLMs. In this work, we first leverage CoT prompting to decouple visual reasoning, enabling the model to perform object-level self-reflection and thereby enhance its ability to generate correct answers. We are then inspired by the vMF distribution and find that measuring semantic consistency through the mean resultant vector length of answer embeddings provides a more stable and efficient metric than traditional approaches. Finally, through extensive experiments, we demonstrate the effectiveness of OSR. Although OSR only makes simple use of CoT, we believe that more sophisticated prompt engineering can further improve its performance across broader application scenarios. We hope that OSR's ex-

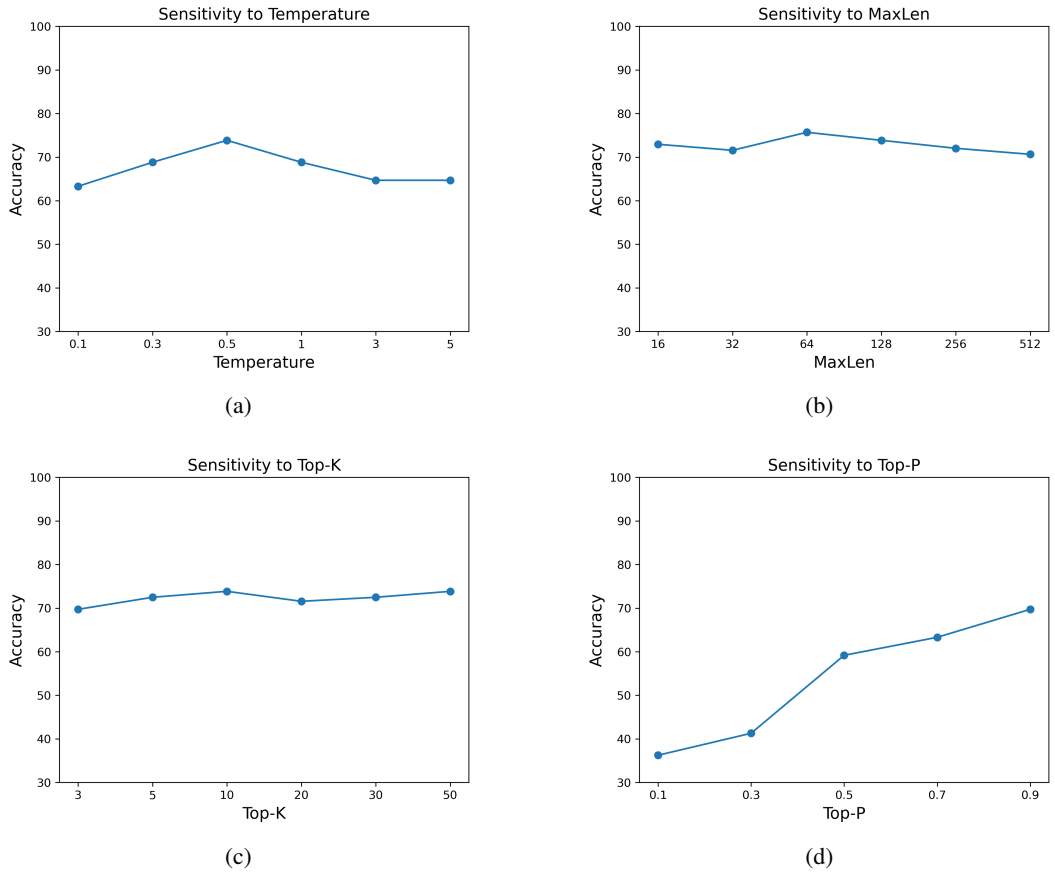

Figure 3: (a) Performance sensitivity to temperature. (b) Performance sensitivity to maximum generation length. (c) Performance sensitivity to top-k. (c) Performance sensitivity to top-p. The performance is measured by accuracy.

ploration of the model's intrinsic capabilities will bring new insights to the field of hallucination detection.