Reducing Hallucinations Utilizing Prompt Engineering and Object Detection

Preetham Pangaluru[1], Derek Wang[2], Kelsey Lin[3], and Pierce Chong[4]

[1]vignesh99@g.ucla.edu
[2]dkwang@g.ucla.edu
[3]klin0522@g.ucla.edu
[4]piercecch@g.ucla.edu

Abstract

Large Vision Language Models (LVLMs) excel at generating and analyzing images and text data, but they often suffer from hallucinations—a misalignment between the actual visual content and the corresponding generated textual description. Hallucinations in LVLMs are a significant challenge, stemming from dataset biases, training deficiencies, and complex multimodal reasoning demands (Liu et al., 2024). Hallucinations reduce the reliability of LVLMs and can severely limit their adoption in the real-world. To address this, our study investigates pre-processed object detection and prompt engineering as potential methods for reducing hallucinations. We aim to identify strategies to improve the accuracy of LVLMs in tasks as well as analyze underlying causes of hallucinations. From our results, we discovered that utilizing object detection significantly improved the performance of the LVLMs. Thus, for future works, it would be beneficial to investigate related methods and focus on other forms of image pre-processing.

1 Introduction

Large Vision Language Models (LVLMs) are powerful tools for comprehending multimodal data and processing a combination of text and image input. They can reason across a variety of input data and are thus useful in a broad range of applications including video summarization, visual question answering, VR/AR, and content moderation. This field is continuing to evolve, and continued research will continue to make LVLMs powerful candidates in downstream applications.

However, LVLMs are not perfect, and improving their performance will be crucial for deploying them reliably in real-world applications. The biggest problem they face are hallucinations. A hallucination is a disagreement between the factual visual image and the corresponding generated textual response (Liu et al., 2024). For example, the model may 'see' an object that is not actually present in the ground-truth image or incorrectly form relationships between visual elements.

Figure 1: Sample input image.

In this image of a dog running with a tennis ball in its mouth, possible hallucinations of an LVLM may include describing multiple dogs present or failing to capture the ball's presence completely. This problem can severely im-

pact applications in safety domains where reliability and high visual accuracy are critical, such as medical imaging.

To ensure LVLMs are more dependable and robust so that there are no real-world consequences, we aim to evaluate existing benchmarks to improve model performance while keeping computational requirements manageable. Our approach will focus on input engineering utilizing RAM (Recognize Anything Model) along with prompt engineering techniques such as double-prompting to guide more precise LVLM responses. We will assess the performance of InternVL, a large-scale vision-language foundation model, on established benchmarks such as Polling-based Object Probing Evaluation (POPE) and HallusionBench (Image-Context Reasoning Benchmark). We will start with evaluating InternVL's baseline performance and compare it to its performance when integrated with our methodology.

2 Related Works

Our approach builds upon prior work regarding mitigating hallucinations in LVLMs. In this section, two main studies that inspired our approach are discussed.

In *Evaluating Object Hallucination in Large Vision-Language Models*, Li et al. proposes a benchmark known as Polling-based Object Probing Evaluation (POPE) specifically designed to test an LVLM's ability to identify objects correctly. It introduces a yes/no questioning framework that asks the model questions about objects that exist in an image, as well as questions on nonexistent objects generated through random, popular, and adversarial sampling (Li et al., 2023a). The evaluation set is thus composed of images, questions on objects within each image, and ground-truth yes/no answers. One main category of this dataset is adversarial, which aims to challenge the model by asking about objects that do not appear in a specific image but are likely to cause a hallucination due to commonly co-occurring objects in the training data.

Additionally, *Visual Evidence Prompting Mitigates Hallucinations in Multimodal Large Language Models*, proposed by Li et al., introduces an approach that incorporates small visual models. These smaller, more specialized models are used to identify present objects in images that are inserted into the LVLM prompt to enhance the model's contextual understanding (Li et al., 2023b). This paper demonstrates that integrating a form of image tagging into the prompt complements the generalist capabilities of LVLMs and substantially reduces hallucinations.

3 Setup

Inspired by the methodology of *Visual Evidence Prompting Mitigates Hallucinations in Multimodal Large Language Models*, we decided to focus on utilizing specialized small vision models to enhance the LVLM's performance. By leveraging the high accuracy of these small models, we can add more detailed information into the prompt, informing the LVLM about what is in the image. Besides utilizing small models, we also attempt to use the LVLM itself to analyze the image before prompting it for input. This is to see how the performance would be different and if there is merit in using LVLMs to assist the processing of other LVLMs.

We utilized a state-of-the-art LVLM to process and understand multimodal data, specifically, **InternVL2-4B**. We decided to utilize InternVL2 instead of other LVLMs such as LLAVA because InternVL2 provides smaller models that we can efficiently run on our systems. In addition, according to their results, InternVL2-4B outperforms LLAVA-NEXT-8B, which is impressive.

To enhance the prompt, we integrated **RAM (Recognize Anything Model)** into our pipeline. The model we specifically chose uses the Swin-L Transformer backbone with 234M parameters. We decided to utilize RAM as our small vision model because of its outstanding accuracy and simple implementation. Our project relied on a combination of cloud-based and local computational resources. We used the Nvidia T4 GPU on GCP along with an Nvidia RTX 4090, with 16 GB and 24 GB of VRAM respectively. In addition, all experiments were performed in a Python 3.12 environment.

To ease the computation during LVLM inference, we utilized RAM to pre-generate image tags for each image. These tags would include every object and attribute detected by the RAM model. For POPE, we generated tags for all 500 unique images. We did the same for HallusionBench, which has 346 unique images. Tags were stored in a JSON file indexed by file name for quick lookup during evaluation.

4 Methodology

Our framework focuses on prompt enhancement for InternVL in a multi-step approach that integrates object detection, prompt engineering, and benchmark evaluations. The main process follows these steps:

1. **Object Detection with RAM**: Before sending image and textual prompts through InternVL, we use RAM as our specialized visual language model to analyze the image and output the image tags. For example, if we were using the prior image of a dog with a tennis ball, the identified tags would be:

   ```
   dog, tennis ball
   ```

2. **Prompt Enhancement**: Combine the generated RAM tags with the textual prompt and feed it to InternVL to guide it for more accurate responses.

   Example:

   - Original Baseline Input: `Describe this image.`
   - Modified Input: `This image has a dog and a tennis ball. Describe this image.`

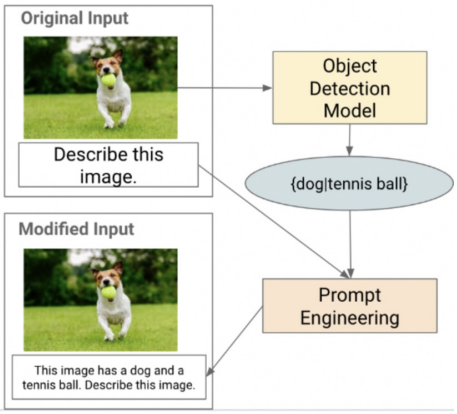

Figure 2: Prompt Enhancement.

To evaluate the effectiveness of this approach on InternVL, we will utilize two benchmark datasets: POPE and HallusionBench.

- **POPE**: This dataset includes 3,000 adversarial test cases structured as yes/no questions that evaluate an LVLM's ability to correctly identify the presence or absence of objects.
- **HallusionBench**: This dataset consists of 951 test cases designed to evaluate more complex aspects such as logical reasoning and visual understanding. It challenges the model with visual IQ-like tasks such as mathematical reasoning, interpreting maps, and spatial reasoning.

These datasets holistically assess an LVLM's performance across a variety of scenarios applicable to real-world situations that require high-level reasoning.

4.1 Evaluation Procedure

For each benchmark dataset (POPE and HallusionBench), we will run the following configurations:

1. **Baseline InternVL**: Run InternVL without the use of RAM or prompt enhancement. This serves as the baseline group.

2. **InternVL with RAM Tags**: Run InternVL with RAM-generated image tags. The prompt will be enhanced to include object tags alongside the original image and prompt.

   Example:

   ```
   This image has a dog and a
   tennis ball. Describe this
   image.
   ```

3. **Double Prompting with RAM Tags**: Run InternVL using a two-step prompting approach with RAM tags. In the first step, prompt InternVL to generate a description using RAM tags for guidance. In the second step, use this result as context and ask the main or task-specific question.

   Example:

   ```
   Based on the description 'A
   dog with a tennis ball on
   the grass', is there a dog
   in the image?
   ```

This framework addresses limitations of LVLMs by helping the model focus on relevant objects present in an image. This reduces the likelihood of hallucinations since the model is provided with explicit context of the visual input as guidance. Additionally, our double prompting method further enhances the framework by encouraging deeper reasoning from the LVLM.

5 Results and Challenges

When evaluating **InternVL**'s performance across both **POPE** and **HallusionBench** datasets among the three configurations (Baseline, RAM Tags, and Double Prompting with RAM Tags), metrics such as accuracy, precision, recall, and F1-score were computed.

Test Case	Accuracy	Precision	Recall	F1
RAM_pope	0.75067	0.92247	0.54733	0.68703
2xPrompting_RAM_pope	0.72667	0.89352	0.51467	0.65313
base_pope	0.49300	0.49223	0.44333	0.46650

Figure 3: POPE Dataset Results.

Test Case	Accuracy	Precision	Recall	F1
RAM_hallusion	0.53838	0.42149	0.25436	0.31726
2xPrompting_RAM_hallusion	0.53102	0.39130	0.20200	0.26645
base_hallusion	0.52681	0.40892	0.27431	0.32836

Figure 4: HallusionBench Dataset Results.

5.1 Baseline InternVL

For the baseline model performance:

- **POPE Accuracy**: 49.3%
- **HallusionBench Accuracy**: 52.7%

These results reflect that the model has basic object recognition abilities, but when it comes to adversarial cases, it struggles to correctly formulate observations.

Within the POPE dataset, the high false positive and false negative rates from the confusion matrix suggests that the baseline model frequently incorrectly identifies objects that do not actually exist in the input image or fails to correctly identify objects that do appear. Within HallusionBench, the baseline model achieves a slightly higher accuracy score compared to POPE. However, it is possible that this is due to the nature of the dataset focusing on broader reasoning rather than detail-oriented, adversarial test cases found in POPE.

The results of the baseline model highlight there is a high rate of hallucination and the model can benefit from explicit guidance and contextual clues for more accurate conclusions.

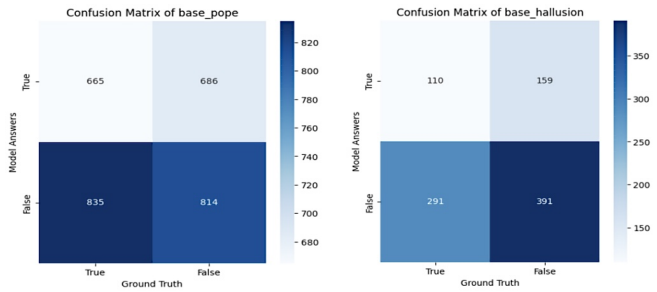

Figure 5: Baseline Confusion Matrices.

5.2 InternVL with RAM Tags

Compared to the baseline accuracy of 49.3% on **POPE**, introducing RAM tags significantly increased the accuracy on POPE to **75.1%**. In total, this approach improved InternVL's accuracy by approximately **25%**. Thus, we can see that adding explicit object tags proves to be helpful in guiding the model to identify the presence and absence of objects.

In the case of predicting the presence of an object, InternVL with RAM becomes highly accurate, achieving a precision of **92.2%**. This means the model is more confident in correctly identifying objects that do exist in an image. It also means that the hallucination has decreased, as the model is less likely to state that an object is present when it is not. The recall improves from 44.3% (baseline) to 54.7%. This indicates that the model still struggles to

find all objects in the image even when provided with tags, so there is no guarantee that it can detect every possible object present.

Compared to the baseline accuracy of 52.7% on **HallusionBench**, RAM tags yield marginal improvement, achieving an accuracy of **53.8%**. The precision also increased slightly from 40.1% to 42.1%. This indicates that InternVL still struggles to accurately identify objects and relationships in more abstract scenarios. As a result, RAM tagging may not be sufficient when it comes to tasks that require higher-order reasoning since object detection does not significantly enhance the model's logical reasoning abilities. However, this method is still viable as it does not significantly increase the memory or computation of the model.

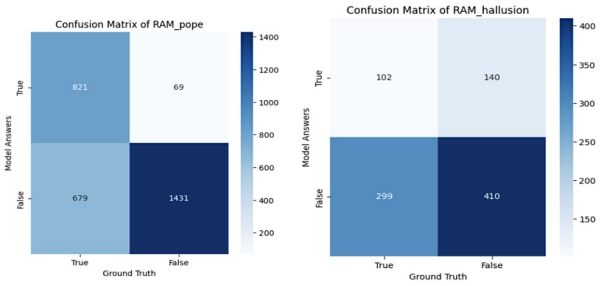

Figure 6: RAM Tagging Confusion Matrices.

5.3 Double Prompting with RAM Tags

The double prompting approach with RAM tagging was also evaluated on both POPE and HallusionBench, and it involved generation of a descriptive response as context in the second prompt for the main question.

POPE achieved an **Accuracy** of **72.67%** (compared to 75.1% with RAM Tags alone). Metrics of accuracy, precision, and recall for double prompting with RAM on POPE are slightly lower compared to RAM tagging alone. Nonetheless, the 72.67% accuracy is still substantial improvement over the baseline accuracy of 49.3% The slight drop in metric scores can be attributed to a slight increase in false negatives, indicating a narrower range of identifiable objects. This may be due to a compounded hallucination effect from double prompting, which introduces additional opportunities for misinterpretation. However, the overall high accuracy still indicates that the model benefits from context provided within the first prompt, even if there are occasional misinterpretations.

HallusionBench double prompting achieved an **Accuracy** of **53.1%** (compared to 53.8% with RAM Tags alone). Similarly, metrics of precision, and recall for double prompting with RAM on HallusionBench are slightly lower compared to RAM tagging alone. It can be concluded that higher level prompts, like those from HallusionBench, require deeper reasoning beyond object tagging aiding the prompts, which is why accuracy remains essentially the same as before.

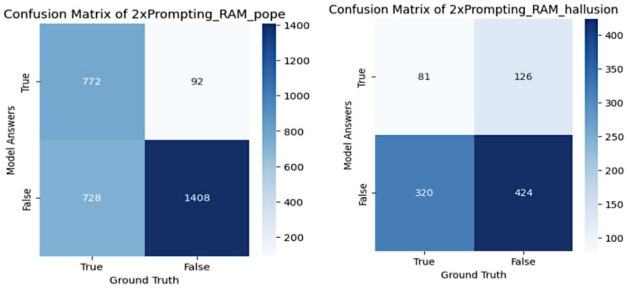

Figure 7: 2x Prompt with RAM Tagging.

5.4 Analysis of Configurations

We can conclude that **InternVL with RAM tagging** significantly enhances the performance in object identification tasks as the model becomes highly accurate. This yields a reduction in object-existence hallucinations. However, when it comes to tasks that require visual logical reasoning or deeper contextual reasoning, RAM tagging does not achieve sizable improvement. This is due to the fact that descriptive tags are too general and do not necessarily guide the model through reasoning.

However, if we look deeper into how well our methods performed in **HallusionBench**, we can identify some trends. When it comes to tasks that require deeper reasoning, we can conclude that RAM tagging slightly increases accuracy in **math, figure, and chart** related questions. However, for interpreting **maps and character recognition**, RAM tagging slightly reduces accuracy. This may be due to these types of images containing densely packed

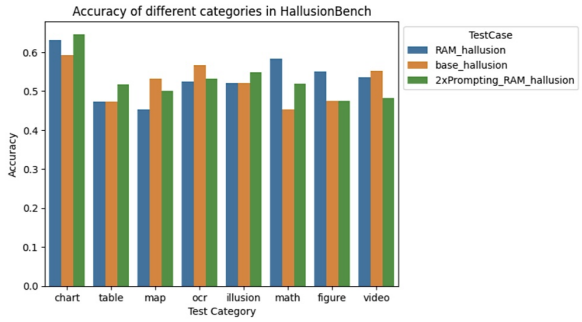

Figure 8: HallusionBench by Categories.

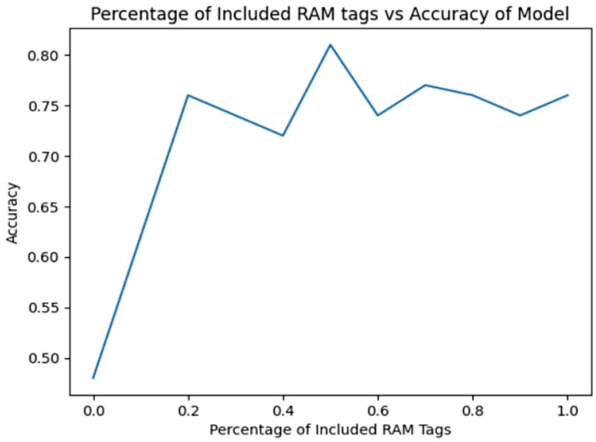

Figure 9: Experiment 1 Results.

elements or very complex relationships, making it difficult for RAM to generate accurate tags. The tags may collapse or oversimplify necessary visual details, or there's too much information/tags, which confuses the LVLM.

As for our **double prompting with RAM tagging method**, the test categories that achieved the highest accuracy amongst all configurations are charts, table, and illusions. This may be attributed to the generation of descriptive context from the first prompt, which more accurately guides the model when answering the main prompt.

5.5 Analysis of RAM Tagging

From our current results, we can definitely see that utilizing RAM tags to enhance the LVLM prompt does lead to better performance in object detection. However, we still do not know how effective these RAM tags are or how they negatively affect the model when incorrect. Thus, we ran a few experiments to test the efficacy and impact of RAM tags.

5.5.1 Experiment 1: Impact of Relevant RAM Tags

In this experiment, we try to show two things: how additional RAM tags affect model performance and how important it is to have the desired object within the tags. For our test cases, we use the first 100 POPE test cases (due to computational complexity) and evaluate them at varying distributions of all RAM tags ranging from 0% to 100%. The RAM tags are sorted by their similarity to the desired object. So tags in the beginning (0-10%) are very similar to the desired object while tags at the end (90-100%) are not similar whatsoever. To calculate the similarity between 2 words, we used Levenshtein distance since it's very simple to import and use.

For example, if our desired word is "snowboard" but "snowboard" is not in the list of RAM tags, other similar words such as "snowboarder" or "snow" will be sorted earlier as they are very similar to "snowboard". The point of sorting and distributing the tags is to see the impact of including the desired object as well as the impact of having additional tags.

The chart shows that as the percentage of included RAM tags increases to around 20%, the accuracy increases significantly. This spike is where the RAM tags should have included the desired object (if RAM detected it), showing that most of the improved accuracy of our method is having the desired object in our list of RAM tags. Once the relevant tag is included, additional tags don't severely impact the accuracy. As we go from 20% to 100% distribution, the accuracy fluctuates but doesn't stray far from 75%.

An interesting note is that at 50% distribution, the accuracy reaches a max of 80%. Perhaps this indicates that a specific distribution of RAM tags does improve the model. However, the data is small, so more testing should be done. Overall, we've shown that having the correct object within the RAM tags is crucial for increasing model accuracy and enhancing performance. In addition, having

additional tags in the prompt does not severely impact the model's performance. Thus, trimming/pruning RAM tags is not necessary. However, there might be a distribution that gives high accuracy, but this requires more testing.

5.5.2 Experiment 2: Effects of Incorrect RAM Tags

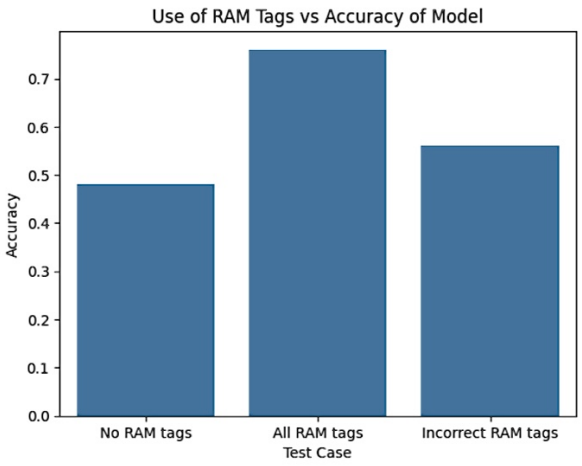

Figure 10: Experiment 2 Results.

For this experiment, we test to see how incorrect RAM tags impact the LVLM model performance. Instead of using the correct RAM tags for the first 100 POPE cases, we use the RAM tags from the last 100 POPE cases to purposely create incorrect tags for our test cases.

Surprisingly, the results show that using completely incorrect RAM tags still yields higher accuracy compared to using the baseline model. Looking deeper into the results, we can see that the model with incorrect tags defaults to returning "False" around 90% of the time, meaning the model doesn't detect most of the desired objects in the image. This indicates that the incorrect tags are causing the LVLM to grow confused and uncertain.

Thus, when the LVLM receives incorrect RAM tags, the model rarely hallucinates objects that are not present and becomes very negative. This might signify that the LVLM puts a lot of focus/trust into the prompt, which makes the discrepancy between prompt and image very effective.

6 Limitations and Future Work

Our methodology demonstrates significant improvement in InternVL performance through prompt engineering and object detection, but there are still limitations and areas for future work. For example, incorporating chain-of-thought prompting could potentially improve the model's reasoning ability. A step-by-step reasoning process could guide the model to generate more context-aware responses that improve model accuracy. Further research can explore adjusting the parameters of RAM tags or finding methods for dynamic selection criteria to further improve relevance

Additionally, our model currently struggles on tasks requiring deeper reasoning beyond simply object detection, so integrating other specialized models such as VQA models for extracting object-attribute relationships can enhance the robustness of prompt engineering. Object-detection models such as YOLO for domain-specific detection tasks can further improve performance in specialized applications.

Our approach was assessed on InternVL. Expanding evaluation to other LVLMs of varying architecture, such as LLaVA, or other datasets that focus on object-attribute and object-relation hallucinations would help us assess the generalizability of our findings. Furthermore, it would highlight aspects that LVLMs still struggle on, providing a more comprehensive understanding of model performance.

Exploring these directions would further refine our proposed methodology to achieve accurate and consistent results in a variety of real-world scenarios.

7 Conclusion

Our approach highlights the importance of reducing hallucinations in LVLMs to improve the reliability of applications in downstream tasks. The integration of prompt engineering with object detection provides a robust framework that reduces hallucinations without model retraining. The plug-and-play method is straightforward to implement, and there is no requirement to modify the LVLM's architecture, making it adaptable for use cases. Additionally, our methodology does not significantly increase model size, maintaining LVLM efficiency. Such

advantages make our framework a practical approach to improve the accuracy and performance of LVLMs while mitigating hallucinations.

References

[1] Li, W., Huang, Z., Lu, L., Lu, Y., Tian, X., Shen, X., & Ye, J. (2023). *Visual Evidence Prompting Mitigates Hallucinations in Multimodal Large Language Models*. International Conference on Learning Representations (ICLR). Retrieved from https://openreview.net/forum?id=xh3XUaB8M9

[2] Li, Y., Du, Y., Zhang, L., Wang, Y., & Wen, J. (2023). *Evaluating Object Hallucination in Large Vision-Language Models*. Conference on Empirical Methods in Natural Language Processing (EMNLP). Retrieved from https://www.semanticscholar.org/paper/Evaluating-Object-Hallucination-in-Large-Models-Li-Du/206400aba5f12f734cdd2e4ab48ef6014ea60773

[3] Liu, Y., Zhang, T., & Wang, X. (2024). *A Survey on Hallucination in Large Vision-Language Models*. Retrieved from https://arxiv.org/abs/2402.00253