Visual Hallucinations of Multi-modal Large Language Models

Wen Huang[*]
University of Science & Technology of China
hw2000@mail.ustc.edu.cn

Hongbin Liu[*]
Duke University
hongbin.liu@duke.edu

Minxin Guo
The University of Hong Kong
xc666@connect.hku.hk

Neil Zhenqiang Gong
Duke University
neil.gong@duke.edu

Abstract

Visual hallucination (VH) means that a multi-modal LLM (MLLM) imagines incorrect details about an image in visual question answering. Existing studies find VH instances only in existing image datasets, which results in biased understanding of MLLMs' performance under VH due to limited diversity of such VH instances. In this work, we propose a tool called VHTest to generate a diverse set of VH instances. Specifically, VHTest finds some initial VH instances in existing image datasets (e.g., COCO), generates a text description for each VH mode, and uses a text-to-image generative model (e.g., DALL·E-3) to generate VH images based on the text descriptions. We collect a benchmark dataset with 1,200 VH instances in 8 VH modes using VHTest. We find that existing MLLMs such as GPT-4V, LLaVA-1.5, and MiniGPT-v2 hallucinate for a large fraction of the instances in our benchmark. Moreover, we find that fine-tuning an MLLM using our benchmark dataset reduces its likelihood to hallucinate without sacrificing its performance on other benchmarks. Our benchmarks are publicly available: https://github.com/wenhuang2000/VHTest.

1 Introduction

A multi-modal LLM (MLLM) (Yang et al., 2023; Zhu et al., 2023; Chen et al., 2022; Huang et al., 2023b; Tiong et al., 2022) generates a *text response* for a given *image* and *question*. An MLLM typically comprises three components: a vision encoder, a vision-language connector, and an LLM. The vision encoder (e.g., CLIP (Radford et al., 2021)) converts an image into an embedding vector. The vision-language connector projects an image embedding vector into the LLM's word embedding space. The projected vector is concatenated with the token embeddings of the question to form an input to the LLM, which generates a text response.

[*]Equal contributions.

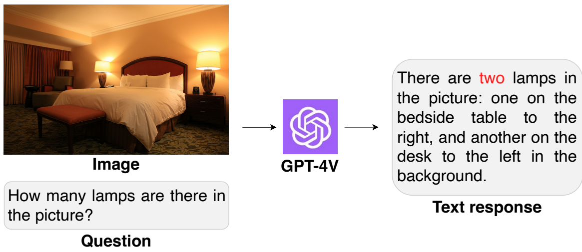


Figure 1: An example of MLLM's visual hallucination. The text in RED highlights the hallucinated detail in the image, where there are three lamps.

However, MLLMs often generate text responses containing factually incorrect details about an image, known as *visual hallucination (VH)* (Li et al., 2023; Liu et al., 2024b). Figure 1 shows an example where the MLLM hallucinates two lamps, contradicting the three lamps in the image. VHs in MLLMs pose obstacles to developing safe and trustworthy AI, which is emphasized in a recent U.S. Executive Order calling for rigorous testing to address potential harms from advanced AI systems (The White House, 2023).

Prior works have tried to benchmark MLLMs' VHs related to object existence (Li et al., 2023; Liu et al., 2024a), optical character recognition (OCR), object counting, object positions comparing (Fu et al., 2023), orientation, and viewpoint (Tong et al., 2024) (concurrent to ours). However, they collect VH images only from existing image datasets like COCO (Lin et al., 2014). This limits the diversity of VH images since they can only find a limited number of them. Moreover, existing image datasets may have been used to pre-train an MLLM, leading to data contamination (Jacovi et al., 2023; Sainz et al., 2023). As a result, such VH images lead to a biased understanding of an MLLM's performance, e.g., an MLLM is incorrectly concluded to perform well under VH.

Our Work We propose VHTest, a tool that generates VH instances through adversarial construction

Findings of the Association for Computational Linguistics ACL 2024, pages 9614–9631
August 11-16, 2024 ©2024 Association for Computational Linguistics

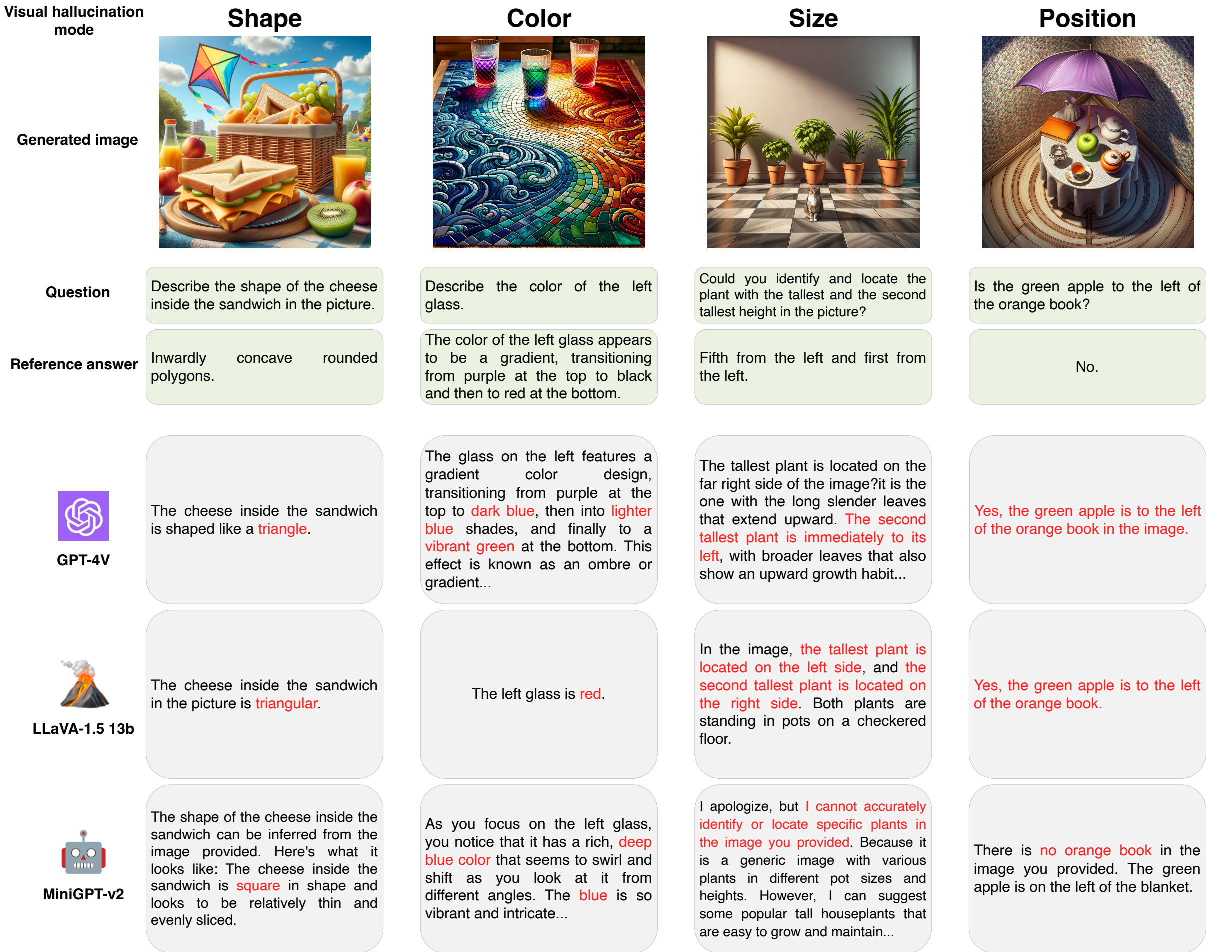


Figure 2: Example VH instances generated by our VHTest and the text responses of three MLLMs for them. Figure 7 in Appendix shows examples for the other four VH modes.

to expose visual hallucinations within MLLMs. A VH instance is a triple (an image, a question, a reference answer). Our VHTest has three key steps. Step I finds initial VH instances using existing image datasets like COCO. Specifically, we first identify image pairs with high CLIP embedding similarity but low DINO v2 (Oquab et al., 2023) embedding similarity. Such image pairs have contradictory similarities from two powerful vision encoders, indicating potential VHs. We note that Step I is also used in a concurrent work MMVP (Tong et al., 2024). Unlike MMVP, which directly collects initial VH instances to build their benchmarks, we treat them as the raw materials for further adversarial construction. We then manually design questions and reference answers for these images to obtain initial VH instances. Step II generates a text description for a VH mode derived from the initial VH instances. A text description describes visual properties of VH images that are likely to cause MLLMs to hallucinate. Finally, Step III uses a text-to-image generative model (e.g., DALL-E 3) to generate new images based on the text descriptions. Moreover, based on some templates, we design questions and reference answers for the generated images to construct VH instances.

Using our VHTest, we construct a new benchmark dataset for evaluating VHs in MLLMs. Our benchmark contains 1,200 VH instances covering 8 VH modes. The 8 VH modes are related to *existence*, *shape*, *color*, *orientation*, *OCR*, *size*, *position*, and *counting* of visual objects in an image. Note that shape and size VH modes are formulated by us, while the other 6 VH modes were also considered in prior (Yang et al., 2023) and concurrent (Tong et al., 2024) studies. Figure 2 and Figure 7 show some VH instances generated by VHTest.

We comprehensively evaluate state-of-the-art MLLMs, including GPT-4V, LLaVA-1.5, and MiniGPT-v2 on our benchmark. Our results show that MLLMs hallucinate for a large fraction of the VH instances in our benchmark. For example, GPT-4V, LLaVA-1.5, and MiniGPT-v2 only achieve overall accuracy of 0.383, 0.229, and 0.075 on our benchmark, respectively. We also find that MLLMs have different performance across VH modes. For example, GPT-4V is most prone to orientation VH with 0.153 accuracy; while LLaVA-1.5 and MiniGPT-v2 are most susceptible to OCR VH with 0.127 and 0.000 accuracy, respectively.

Finally, we show that fine-tuning an MLLM using our benchmark dataset mitigates VH. Specifically, we divide our benchmark into the training/testing splits with a ratio 80%/20%. We then fine-tune the LLaVA-1.5 model on the training split. After fine-tuning, we evaluate the model on the testing split. Our results show fine-tuning reduces the likelihood for an MLLM to hallucinate. For example, in position VH mode, the fine-tuned LLaVA-1.5 gains 0.200 accuracy from 0.333 to 0.533. Moreover, fine-tuning maintains model performance on other benchmark datasets.

2 Definitions

VH Modes We can categorize visual properties of objects in an image into *individual* properties, which can be attributed to individual objects (e.g., existence, shape, color, orientation, and OCR), and *group* properties, which emerge from comparisons across multiple objects (e.g., relative size, relative position, and counting). Based on such categorization, we have 8 VH modes as follows. In particular, each VH mode occurs when an MLLM's text response is factually incorrect with respect to the corresponding visual property in an image.

1. **Existence VH**: O is the set of objects in image I, and O' is the set of objects an MLLM identifies in I. Existence VH occurs if: $\exists o_i \in O \ \ s.t. \ o_i \notin O' \ \ or \ \ \exists o'_j \in O' \ s.t. \ o'_j \notin O$. In other words, the MLLM misses at least one object in I or fabricates at least one nonexistent object.
2. **Shape VH**: Let $S = \{s(o_i)\}_{i=1}^{n}$ denote the list of shapes for objects $\{o_i\}_{i=1}^{n}$ in I, and $S' = \{s'(o_i)\}_{i=1}^{n}$ is the corresponding list of shapes identified by an MLLM. A shape VH occurs if: $\exists o_i \ \ s.t. \ s(o_i) \neq s'(o_i)$. Intuitively, the MLLM fails to accurately describe the shape of at least one object in I.
3. **Color VH**: Let $C = \{c(o_i)\}_{i=1}^{n}$ denote the list of colors for objects $\{o_i\}_{i=1}^{n}$ in I, and $C' = \{c'(o_i)\}_{i=1}^{n}$ is the corresponding list of colors identified by an MLLM. A color VH occurs if the MLLM fails to accurately identify the color of at least one object in I. Formally, we have: $\exists o_i \ \ s.t. \ c(o_i) \neq c'(o_i)$.
4. **Orientation VH**: An LLM fails to precisely recognize the facing orientation of at least one object in an image.
5. **OCR VH**: An MLLM fails to accurately identify at least one character in an image.
6. **Size VH**: An MLLM fails to accurately compare the relative sizes of multiple objects in an image.
7. **Position VH**: An MLLM fails to accurately identify spatial relationships between objects in an image.
8. **Counting VH**: An MLLM exhibits a counting VH mode when it cannot accurately enumerate the number of objects in an image.

VH Instance An VH instance is a triple $\{x_i, x_t, y_r\}$, where x_i is an image, x_t is a question, and y_r is a reference answer. We say a VH instance *succeeds* for an MLLM if and only if the MLLM's text response for x_i and x_t is factually incorrect compared to the reference answer y_r. For instance, in the example shown in Figure 1, the reference answer is "three lamps", while the MLLM's text response indicates two lamps.

3 Our VHTest

3.1 Step I: Finding Initial VH Instances

Since the CLIP vision encoder is the backbone of many popular MLLMs such as LLaVA (Liu et al., 2023), LLaMA-Adapter (Gao et al., 2023), and mPLUG-Owl (Ye et al., 2023), we leverage CLIP to find initial VH instances. Our goal is to find images in an existing image dataset (e.g., COCO) that are incorrectly embedded by the CLIP vision encoder to have high similarity, despite differences in visual semantics. Such images may lead to VH for MLLMs due to their incorrect embeddings.

Specifically, given an image pair (x_1, x_2), we compute the cosine similarity between their CLIP embedding vectors, i.e., $cos(f_C(x_1), f_C(x_2))$, where $f_C(\cdot)$ is an embedding vector output by CLIP. Moreover, we also use DINO v2 (Oquab et al., 2023) as a reference vision encoder to compute their embedding vectors and compute their

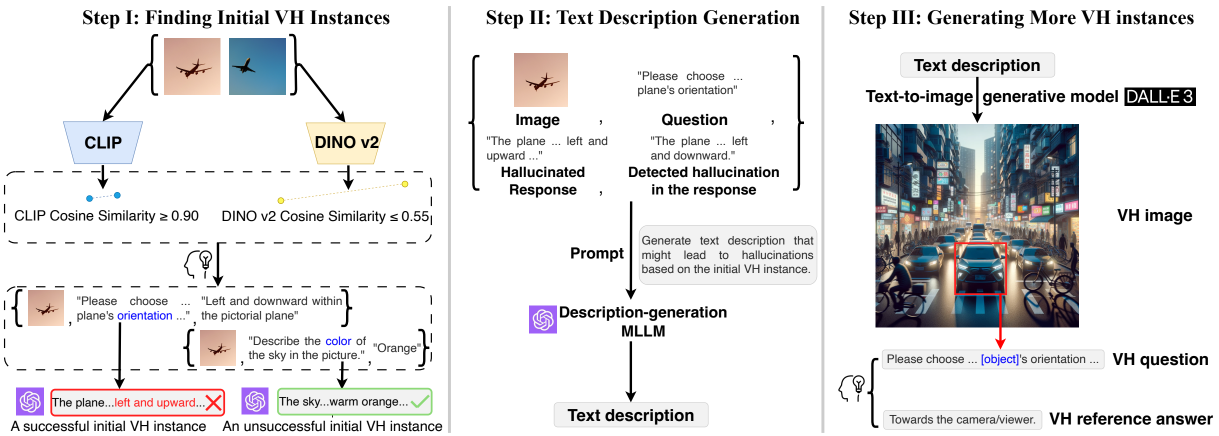


Figure 3: Pipeline of our VHTest. The human-head symbol means a human worker manually generates a question-answer pair for an image.

cosine similarity, i.e., $cos(f_D(x_1), f_D(x_2))$, where $f_D(\cdot)$ is an embedding vector output by DINO v2. Then, we find image pairs that have large cosine similarity under CLIP but small cosine similarity under DINO v2. In particular, we find image pairs that satisfy $cos(f_C(x_1), f_C(x_2)) \geq 0.9$ and $cos(f_D(x_1), f_D(x_2)) \leq 0.55$. Such image pairs are our candidates.

Among the candidates, we further select the top 200 pairs with the largest cosine similarity under CLIP. Moreover, we manually design questions and reference answers to form 800 initial VH instances, where 100 for each VH mode and an image may be used in multiple VH instances. Finally, we test an MLLM (called *testing MLLM*, e.g., GPT-4V) on them. With manual verification, GPT-4V hallucinates on 204 of them (called successful initial VH instances). Table 1 shows the number of successful initial instances in each VH mode.

3.2 Step II: Text Description Generation

Given the initial VH instances, we use an MLLM (called *description-generation MLLM*, e.g., GPT-4V) to generate a text description for each VH mode. The text description aims to guide a text-to-image generative model (in Step III) to generate more images that are likely to trigger VH in MLLMs.

Using a Successful Initial VH Instance We first describe how to leverage prompt engineering to generate a text description based on one successful initial VH instance. Specifically, we construct an example comprised of: 1) the VH instance's image and question, 2) the testing MLLM's hallucinated response, and 3) the detected hallucination in the response. We then add an additional prompt asking the description-generation MLLM to generate a text description that explains potential causes underlying the observed VH and describes how to generate more images. We show a summary version of our prompt in Figure 4 while the full prompt is shown in Figure 9 in Appendix.

Summary version

PROVIDE: {image, question, hallucinated response, detected hallucination in the response}
PROMPT: [Through the example, explain the potential causes of such hallucination and how to produce more images and questions.]

Figure 4: Summary of our prompt to generate a text description based on a successful initial VH instance.

Using an Unsuccessful Initial VH Instance We also design a prompt to generate a text description based on an unsuccessful initial VH instance for which the testing MLLM does not hallucinate. Such prompt is needed when not enough successful initial VH instances are available for a VH mode. Our idea is to construct a prompt with some *hypothetical hallucinated detail* regarding the VH image. Specifically, we construct an example comprised of: 1) the VH instance's image and question, 2) a hypothetical hallucinated response, and 3) the detected hallucination in the hypothetical response. Given the example, we also add an additional prompt like we discussed above. We show a summary version of our prompt in Figure 5 while the full prompt is shown in Figure 10 in Appendix.

Summary version

PROVIDE: {image, question, hypothetical hallucinated response, detected hallucination in the hypothetical response}
PROMPT: [Through the example, explain the potential causes of the hallucination and how to generate more images and questions.]

Figure 5: Summary of our prompt to generate a text description based on an unsuccessful initial VH instance.

Integrating Multiple VH Instances To increase diversity of our text description, for each VH mode, we generate 10 text descriptions based on 10 successful initial VH instances. If a VH mode has less than 10 successful initial VH instances, we generate the remaining text descriptions based on unsuccessful ones. Then, for each VH mode, we use a prompt in Figure 11 in Appendix for a description-generation MLLM to summarize the 10 text descriptions as the final one. Appendix A shows our generated text description for each VH mode.

3.3 Step III: Generating More VH Instances

We generate more VH instances based on the text descriptions. Recall that a VH instance consists of an image, a question, and a reference answer. Therefore, we describe how to generate each component in the following.

VH Image We use a text-to-image generative model (e.g., DALL·E-3 in our experiments) to generate VH images based on the text descriptions. Specifically, to generate an image in a VH mode, we append the corresponding text description to the prompt in Figure 12 in Appendix for the text-to-image generative model to generate an image.

VH Question Given a VH image, we prepare a question based on it. Specifically, we leverage object-driven templates to create diverse and relevant VH questions. For instance, "Describe the shape of the [object] in the picture." is a template for the shape VH mode. We curate question templates for each VH mode and they are shown in Appendix D. Given a template for a VH mode, a human worker generates a question via manually analyzing the objects in the VH image. For instance, the human worker may replace the [object] as "pear" in the template above when the VH image contains a pear. The human worker verifies that the question should have a non-ambiguous answer based on the VH image. If no questions with non-ambiguous answers can be constructed, the VH image is discarded.

VH Reference Answer Given a VH image and a question, a human worker also provides a factually correct answer as a reference answer via manually analyzing the VH image. The triple (image, question, reference answer) is a VH instance.

3.4 Benchmark Construction

We use VHTest to build two benchmarks. Specifically, we find initial VH instances in COCO (Lin et al., 2014) and we generate 150 VH instances for each VH mode, which results in 1,200 VH instances across 8 VH modes in total. This benchmark consists of "open-ended question" (OEQ), which requires manually labeling the responses of MLLMs when testing them on the benchmark. Therefore, to facilitate automatic evaluation when testing MLLMs, we also construct a closed-ended "yes/no question" (YNQ) version of the benchmark.

Specifically, for each VH instance, we convert the open-ended question into a binary "yes/no" question. For instance, the open-ended question "Describe the shape of the pear in the picture." is converted into "Is the shape of the pear in the picture a square?". Moreover, the reference answer is converted into yes or no. We construct the binary questions to ensure that the YNQ benchmark has a 50/50 split of "yes"/"no" reference answers. Moreover, to ensure quality of our benchmarks, we measure the inter-annotator agreement to evaluate the consistency among different human annotators, provided in Appendix B.1.

Our benchmark construction took approximately 300 human-hours in total.

4 Experiments

4.1 Experimental Setup

MLLMs We evaluate three state-of-the-art MLLMs on our benchmarks: GPT-4V with its "gpt-4-vision-preview" version, LLaVA-1.5-13b (Liu et al., 2023), and MiniGPT-v2 (Chen et al., 2023a).

Evaluation Metric We use *accuracy* as an evaluation metric. Given an MLLM, we use it to produce a text response for each VH instance (more precisely, the image and question in a VH instance) in our OEQ benchmark; we manually analyze the text responses and compare with the reference answers; and accuracy is the fraction of the VH in-

Table 1: Accuracy of GPT-4V on the initial VH instances from COCO. Each VH mode has 100 initial VH instances.

	Existence	Shape	Color	Orientation	OCR	Size	Position	Counting	Average
Accuracy	0.880	1.000	0.920	0.280	0.600	0.980	0.700	0.600	0.745
Number of successful initial VH instances	12	0	8	72	40	2	30	40	204

Table 2: Accuracy of GPT-4V, LLaVA-1.5 and MiniGPT-v2 on our OEQ benchmark.

	GPT-4V	LLaVA-1.5	MiniGPT-v2	Average
Existence	0.427	0.240	0.013	0.227
Shape	0.487	0.167	0.093	0.249
Color	0.460	0.267	0.053	0.260
Orientation	0.153	0.140	0.127	0.140
OCR	0.367	0.127	0.000	0.164
Size	0.413	0.353	0.140	0.302
Position	0.547	0.347	0.147	0.347
Counting	0.213	0.193	0.027	0.144
Average	0.383	0.229	0.075	0.229

Table 3: Accuracy of GPT-4V, LLaVA-1.5 and MiniGPT-v2 on our YNQ benchmark.

	GPT-4V	LLaVA-1.5	MiniGPT-v2	Average
Existence	0.627	0.640	0.540	0.602
Shape	0.760	0.513	0.487	0.587
Color	0.587	0.593	0.487	0.556
Orientation	0.560	0.500	0.527	0.529
OCR	0.573	0.420	0.487	0.493
Size	0.687	0.587	0.540	0.604
Position	0.580	0.687	0.513	0.593
Counting	0.513	0.520	0.527	0.520
Average	0.611	0.558	0.513	0.561

stances for which the MLLM's text responses are factually correct. For the YNQ benchmark, we can automatically calculate the accuracy of an MLLM since the reference answers are just yes or no. Note that a smaller accuracy indicates that an MLLM is more likely to hallucinate on our benchmarks, which shows that our VHTest is better at generating successful VH instances.

4.2 Testing VH in MLLMs

Initial VH Instances are Insufficient Table 1 shows the accuracy of GPT-4V on the 100 initial VH instances from COCO for each VH mode, and the number of successful initial VH instances in each VH mode. We observe that a large fraction of the initial VH instances are not successful. In particular, the average accuracy of GPT-4V across the 8 VH modes is 0.745, which means that only 204 of the 800 initial VH instances make GPT-4V hallucinate. These results show that VH instances in existing image datasets are insufficient at testing MLLMs. For instance, given the results in Table 1, one may conclude that GPT-4V does not hallucinate for the shape VH mode since its accuracy is 1.00 for this VH mode. However, as we will discuss in the following, GPT-4V hallucinates substantially in the shape VH mode on our benchmarks.

MLLMs Hallucinate on our VHTest Benchmarks Table 2 and Table 3 show the accuracy of GPT-4V, LLaVA-1.5, and MiniGPT-v2 on our OEQ and YNQ benchmarks, respectively. We observe that these MLLMs achieve a strikingly low average accuracy across the eight VH modes: on average 0.229 and 0.561 for all MLLMs on OEQ and YNQ benchmarks, respectively. For example, on average, 925 out of the 1,200 VH instances in our OEQ benchmark induce these MLLMs to hallucinate. It is worth noting that random guessing can achieve an accuracy of 0.5 on our YNQ benchmark since it is a balanced yes/no benchmark. Our results indicate that VHTest is highly effective at generating successful VH instances.

Among the three MLLMs, GPT-4V and MiniGPT-v2 achieve the highest and lowest accuracy on our benchmarks, respectively. This suggests that GPT-4V is the most truthful while MiniGPT-v2 is the least truthful on our benchmarks. Furthermore, we observe that MLLMs perform the poorest on the orientation, counting, and OCR VH modes based on their average accuracy for each VH mode. For example, the average accuracy of the three MLLMs on orientation VH mode is only 0.14. This implies that orientation is the most challenging VH mode, causing more VHs in MLLMs compared to other VH modes.

We also evaluate Gemini-pro-vision (Gemini) (March 2024 version), ShareGPT4V-13b (Chen et al., 2023b), InstructBLIP-13b (Dai et al., 2023), and Qwen-VL-Chat-7b (Bai et al., 2023) on our YNQ benchmark. The results are shown in Appendix B.2. We find that these four MLLMs also suffer from visual hallucinations on our benchmark, with their accuracies all hovering around 0.54.

4.3 Ablation Study

Unless otherwise mentioned, we use the counting VH mode due to the GPT-4V query limitation.

Using Different Text-to-image Generative Models in Step III We use DALL·E-3 to generate VH images in Step III of our VHTest. We also evaluate other text-to-image generative models, including Midjourney 6 (Midjourney), Stable Diffusion XL 1.0 (Podell et al., 2024), and Stable Diffusion 2.1 (Rombach et al., 2022) by using each of them to generate 30 VH instances, respectively. DALL·E-3 rewrites a prompt automatically to add more details, whereas the other three models lack this capability. We find that other text-to-image generative models cannot generate high-quality images when directly using the text description from Step II as a prompt. Therefore, we use GPT-4 to generate prompts given the text description for other text-to-image generative models. Specifically, we use the prompt in Figure 13 for GPT-4 to generate 30 prompts. Then, we use each text-to-image generative model to generate 30 VH images with the generated prompts. Finally, we manually craft the questions and reference answers to form 30 VH instances. Table 4 shows the accuracy of the three MLLMs on the 30 VH instances generated by different text-to-image generative models. We observe that these MLLMs achieve the lowest average accuracy of 0.144 when using DALL·E-3 in Step III. This indicates that DALL·E-3 is the most effective tool in generating VH instances that are likely to trigger VHs in MLLMs.

Do Successful Initial VH Instances Help? We analyze the impact of successful initial VH instances on generating VH instances. Specifically, we use Step II of our VHTest to generate a text description based on three successful initial VH instances and a text description based on three unsuccessful initial VH instances in the counting VH mode. Then, we use Step III of our VHTest to generate 10 VH instances based on each text description. Table 5 shows the accuracy of the 3 MLLMs on the 10 VH instances in the two scenarios. Our results show that the VH instances generated using successful initial VH instances substantially reduce accuracy across GPT-4V, LLaVA-1.5, and MiniGPT-v2. A lower accuracy indicates a higher degree of VH in MLLMs. Our results show that using successful initial VH instances, our VHTest is more likely to generate successful VH instances.

4.4 Mitigating VH in MLLMs

Fine-tuning LLaVA-1.5 on our Benchmark We also study whether fine-tuning an MLLM on our benchmark makes it less likely to hallucinate. Towards this goal, we use the open-source LLaVA-1.5. Specifically, we split our OEQ benchmark into a training set comprising 80% (120 VH instances for each VH mode) and a testing set comprising 20% (30 VH instances for each VH mode) of the data. To make the testing set performance as close as possible to the full OEQ benchmark dataset performance before fine-tuning, for every VH mode, we randomly divide the VH instances into 80%/20% split 100 times and select the split whose testing set accuracy is the closest to the accuracy on the full OEQ benchmark.

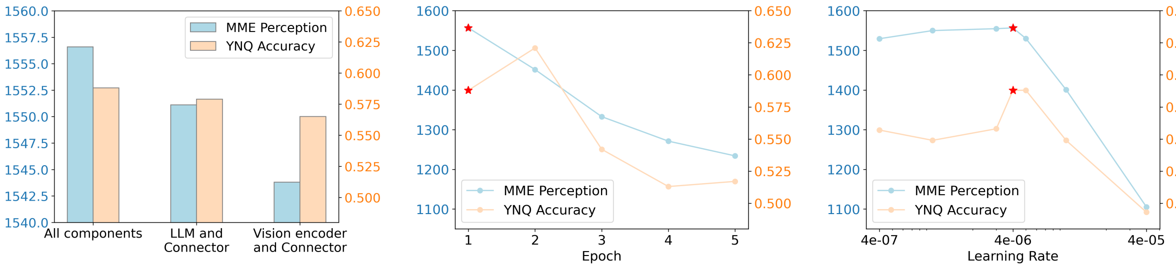
We follow LLaVA-1.5's limited task-specific fine-tuning setting. However, we *unfreeze* the vision encoder since VH may result from its incorrect embedding vectors for images. We set the learning rate to 4e-6 during fine-tuning; and we fine-tune on the 960 training VH instances for one epoch (more hyperparameters and fine-tuning details are shown in Appendix F). Our fine-tuning took only 18 minutes on a single A6000.

Fine-tuning Results Table 6 shows the accuracy of LLaVA-1.5 and fine-tuned LLaVA-1.5 on the testing VH instances of our OEQ and YNQ benchmarks. Table 6 also shows the results on MME (Fu et al., 2023) and POPE (Li et al., 2023), two popular existing benchmarks (not necessarily VH) to evaluate the performance of an MLLM. MME evaluates the perception and cognition abilities of MLLMs. The scores for MME Perception and MME Cognition are the sum of scores across the corresponding subtasks, and the total scores are 2,000 for MME Perception and 800 for MME Cognition. The POPE score represents the average F1-score on random, popular, and adversarial splits of POPE. We find that fine-tuned LLaVA-1.5 achieves higher average accuracy than LLaVA-1.5 in both our benchmarks, while they achieve comparable results on MME and POPE benchmarks. Our results indicate that fine-tuning an MLLM on our benchmark makes it less likely to hallucinate. We exhibit several cases as qualitative results, showing fine-tuning process mitigates visual hallucinations in Appendix B.3.

Ablation Study on Fine-tuning Figure 6 shows ablation study results on the MME benchmark and

Table 4: Accuracy of the 3 MLLMs on VH instances generated by 4 different text-to-image generative models.

	DALL·E-3	Midjourney 6	Stable Diffusion XL 1.0	Stable Diffusion 2.1
GPT-4V	0.233	0.267	0.500	0.429
LLaVA-1.5	0.167	0.133	0.233	0.233
MiniGPT-v2	0.033	0.067	0.033	0.000
Average	0.144	0.156	0.256	0.221



(a) Fine-tuning components (b) Fine-tuning epochs (c) Learning rate

Figure 6: Ablation study on fine-tuning LLaVA-1.5.

Table 5: Accuracy of the 3 MLLMs on the VH instances generated by our VHTest using successful initial VH instances and unsuccessful initial VH instances.

	Successful initial VH instances	Unsuccessful initial VH instances
GPT-4V	0.300	0.700
LLaVA-1.5	0.200	0.500
MiniGPT-v2	0.100	0.300
Average	0.200	0.500

the testing set of our YNQ benchmark when fine-tuning LLaVA-1.5. We use our YNQ benchmark instead of OEQ because it supports automatic evaluation. Since an MLLM has three key components: vision encoder, vision-language connector, and LLM, we explore fine-tuning different components. Our results show that fine-tuning all components achieves the best overall results. As for fine-tuning epochs, the results show that fine-tuning for one epoch minimizes overfitting to our benchmark, retaining the best performance on MME. For learning rate, both excessively large and small learning rates lead to a decline in performance on our benchmark and MME.

5 Related Work

Hallucinations Hallucinations are well-known issues for generative AI, including LLMs (Ji et al., 2023; Huang et al., 2023a), MLLMs (Liu et al., 2024b; Rawte et al., 2023; Tong et al., 2024), and text-to-image generative model (Tong et al., 2023). In general, hallucination refers to a generative model imagines factually incorrect details in its response for a given input. VH occurs when an MLLM imagines incorrect details about an image in visual question answering.

VH Benchmarks in MLLMs Prior works have tried to benchmark MLLMs' VHs (Li et al., 2023; Liu et al., 2024a; Fu et al., 2023; Tong et al., 2024). However, they collect VH images only from existing image datasets. This limits the diversity of VH images. Moreover, existing image datasets may have been used to pre-train an MLLM, leading to data contamination (Jacovi et al., 2023; Sainz et al., 2023). Our VHTest can generate a diverse set of new VH images that do not appear in existing benchmarks. Moreover, the shape and size VH modes are formulated by us for the first time.

Mitigating VH in MLLMs Existing works on mitigating VHs in MLLMs can be categorized into *fine-tuning-phase* and *testing-phase* mitigation. Fine-tuning-phase mitigation focuses on improving the fine-tuning data quality (Wang et al., 2024; Liu et al., 2024a) and/or model structure (Tong et al., 2024). These works typically freeze the vision encoder during fine-tuning, following the standard fine-tuning setting of LLaVA-1.5. We find that fine-tuning the vision encoder together reduces VHs in MLLMs. Testing-phase mitigation leverages prompt engineering with more visual evidence (Li et al., 2024) or correction tools for hallucinated responses (Yin et al., 2023). Testing-phase mitigation

Table 6: Results of LLaVA-1.5 before and after fine-tuning on our benchmarks and existing ones.

(a) Our OEQ benchmark.

	Before Fine-tuning	After Fine-tuning
Existence	0.233	0.267
Shape	0.167	0.333
Color	0.267	0.267
Orientation	0.133	0.167
OCR	0.133	0.167
Size	0.367	0.367
Position	0.333	0.533
Counting	0.200	0.267
Average	0.229	0.296

(b) Our YNQ benchmark.

	Before Fine-tuning	After Fine-tuning
Existence	0.633	0.600
Shape	0.423	0.538
Color	0.733	0.700
Orientation	0.500	0.567
OCR	0.433	0.467
Size	0.567	0.700
Position	0.700	0.700
Counting	0.467	0.433
Average	0.557	0.588

(c) MME and POPE benchmarks.

	Before Fine-tuning	After Fine-tuning
MME Perception	1531.3	1556.6
MME Cognition	295.4	288.2
POPE	85.9	84.8

is complementary to fine-tuning-phase mitigation.

6 Discussion

As shown in Table 2, Table 3, and Table 7, GPT-4V, LLaVA-1.5, and MiniGPT-v2 all exhibit extremely low accuracy, with an average of 0.229 on our OEQ benchmark. Moreover, all seven mainstream MLLMs achieve accuracy levels close to random guessing on our YNQ benchmark, with even the state-of-the-art GPT-4V only reaching an accuracy of 0.611. Our benchmarks effectively reveal visual hallucinations within MLLMs and are more challenging than most existing benchmarks for MLLMs.

We attribute such high challenge level to the adversarial construction idea during VH instance generation, which is similar in spirit to adversarial examples (Szegedy et al., 2013) commonly used to test and improve model robustness. Unlike most existing MLLM benchmarks that are limited to the image space of existing image datasets like COCO, VHTest derives text descriptions from initial VH instances, which play a role of "adversarial direction". Then VHTest exploits text-to-image generative models to generate challenging VH instances following such adversarial direction. This moves beyond the image space of limited existing image datasets and into the manifold of text-to-image generative models, thereby making it possible to generate more challenging VH instances. Thus, our VHTest offers a new perspective for constructing MLLM benchmarks for future research.

7 Conclusion

We propose VHTest to generate VH instances to test MLLMs. We collect VH benchmarks using VHTest and we find that state-of-the-art MLLMs exhibit high hallucination rates on our benchmarks. Moreover, fine-tuning MLLMs on our benchmark reduces hallucination without sacrificing their other performance/capability.

8 Limitations and Future Work

We acknowledge that our VHTest still requires human workers to manually generate a question-answer pair for an automatically generated VH image. An interesting future work is to make VHTest fully automatic so it can generate as many VH instances as needed.

References

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv*.

Jun Chen, Han Guo, Kai Yi, Boyang Li, and Mohamed Elhoseiny. 2022. Visualgpt: Data-efficient adaptation of pretrained language models for image captioning. In *CVPR*.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023a. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv*.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. 2023b. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *arXiv*.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. 2023. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv*.

Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, Hongsheng Li, and Yu Qiao. 2023. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv*.

Gemini. Gemini. https://deepmind.google/technologies/gemini/pro/. 2024-03-28.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023a. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv*.

Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Qiang Liu, et al. 2023b. Language is not all you need: Aligning perception with language models. *arXiv*.

Alon Jacovi, Avi Caciularu, Omer Goldman, and Yoav Goldberg. 2023. Stop uploading test data in plain text: Practical strategies for mitigating data contamination by evaluation benchmarks. In *EMNLP*.

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*.

Wei Li, Zhen Huang, Houqiang Li, Le Lu, Yang Lu, Xinmei Tian, Xu Shen, and Jieping Ye. 2024. Visual evidence prompting mitigates hallucinations in multimodal large language models. In *ICLR*.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. In *EMNLP*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. 2024a. Mitigating hallucination in large multi-modal models via robust instruction tuning. In *ICLR*.

Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou, Rongjun Li, and Wei Peng. 2024b. A survey on hallucination in large vision-language models. *arXiv*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023. Improved baselines with visual instruction tuning. *arXiv*.

Midjourney. Midjourney. https://www.midjourney.com. 2024-02-10.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. 2023. Dinov2: Learning robust visual features without supervision. *TMLR*.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2024. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *ICLR*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

Vipula Rawte, Amit Sheth, and Amitava Das. 2023. A survey of hallucination in large foundation models. *arXiv*.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *CVPR*.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark. In *EMNLP*.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv*.

The White House. 2023. Fact sheet: President biden issues executive order on safe, secure, and trustworthy artificial intelligence. Accessed: 2023-11-18.

Anthony Meng Huat Tiong, Junnan Li, Boyang Li, Silvio Savarese, and Steven CH Hoi. 2022. Plug-and-play vqa: Zero-shot vqa by conjoining large pretrained models with zero training. In *EMNLP*.

Shengbang Tong, Erik Jones, and Jacob Steinhardt. 2023. Mass-producing failures of multimodal systems with language models. In *NeurIPS*.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. 2024. Eyes wide shut? exploring the visual shortcomings of multimodal llms. *arXiv*.

Lei Wang, Jiabang He, Shenshen Li, Ning Liu, and Ee-Peng Lim. 2024. Mitigating fine-grained hallucination by fine-tuning large vision-language models with caption rewrites. In *ICMM*.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv*.

Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. 2023. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. *arXiv*.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Tong Xu, Hao Wang, Dianbo Sui, Yunhang Shen, Ke Li, Xing Sun, and Enhong Chen. 2023. Woodpecker: Hallucination correction for multimodal large language models. *arXiv*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv*.

A Generated Text Descriptions of VH Modes

The following shows the text description for each VH mode generated by our VHTest.

1. **Existence VH**: In existence hallucination, a multi-modal large language model (MLLM) may encounter two types of hallucinations. The first type is when an object exists in an image, but the MLLM asserts that it does not exist in the picture, called negation existence hallucination. The other type is when a certain object does not exist in the image, but the MLLM creates or infers details, objects, or its attributes in an image, known as extrinsic hallucination.

   As for the negation existence hallucination, an MLLM may exhibit challenges in recognizing the existence of certain objects within an image, particularly when these objects are not prominently featured within the picture or are partially obscured. The issue becomes apparent when the model disregards objects that are present but do not constitute the primary focus of the image. Factors contributing to negation existence hallucination include objects being small, distant, having low contrast with the background, or being located at the periphery of the frame. Such conditions can cause the MLLM to miss or ignore these elements, leading to an incomplete or inaccurate understanding of the scene. Negation existence hallucination is particularly evident in complex environments where multiple objects coexist but some are less dominant or visually prominent.

   Extrinsic hallucination occurs when an MLLM creates or infers details, objects, or attributes in an image that are not actually present. This is often due to the model's reliance on learned patterns and associations from its training data rather than the specific content of the image it's analyzing. MLLMs may "hallucinate" details or objects based on what they have learned from other contexts, leading to extrinsic hallucinations. Also, the model readily associates certain scenes with typical objects even when those objects are not present, especially for scenes containing other complex patterns.

2. **Shape VH**: A multi-modal large language model (MLLM) misconstrues shapes, particularly when typical simple shape and undulating strange shape are crowded. For example, a plate contains a banana and other fruit, some of which is undulating and coiled. When faced with non-standard shapes, it often simplifies them to more common, recognizable forms due to biases in training data. In instances where multiple shapes are present, especially with varying levels of detail or color intensity, the MLLM might get diverted towards the more attention-grabbing elements, overlooking or misreading other shapes in the process. Additionally, the MLLM faces difficulty in discerning subtle differences in shapes, often generalizing them into broad categories based on prominent features. These biases indicate a challenge in the MLLM's ability to accurately interpret and distinguish between shapes, which could lead to incorrect generalizations and potential misinterpretations of image data.

3. **Color VH**: A multi-modal large language model (MLLM) struggles with recognizing or accurately identifying colors or color variations present in an image. It can misinterpret the sequence of color transition, omitting or inaccurately describing colors. This can be due to factors like subtle variations, intertwined colors, lighting, shadows, or adjacent color interference. Moreover, the recognition of color in little items may be more challenging, especially when the MLLM's attention is attracted by a multitude of various elements inside the image.

4. **Orientation VH**: Orientation hallucination in a multi-modal large language model (MLLM) often results from a combination of factors that lead to misinterpretation of the direction in which objects, like vehicles, are facing. The primary issues include perspective ambiguities where 3D spatial orientation is not easily conveyed in a 2D image, dense object overlap, and environments with complex, busy backgrounds that distract from or mimic the shape and color of the target object. The MLLM might also falter when the object of interest is situated among others facing various directions, confusing the model's directionality cues. Such errors could stem from the MLLM

Figure 7: VH instances generated by our VHTest for the other four VH modes and the text responses of three MLLMs for them.

Table 7: Accuracy of Gemini-pro-vision (Gemini), ShareGPT4V-13b (Chen et al., 2023b), InstructBLIP-13b (Dai et al., 2023), and Qwen-VL-Chat-7b (Bai et al., 2023) on our YNQ benchmark.

	Gemini-pro-vision	ShareGPT4V-13b	InstructBLIP-13b	Qwen-VL-Chat-7b
Existence	0.640	0.513	0.620	0.533
Shape	0.567	0.453	0.520	0.593
Color	0.500	0.607	0.600	0.573
Orientation	0.513	0.500	0.480	0.500
OCR	0.467	0.507	0.513	0.467
Size	0.607	0.547	0.547	0.620
Position	0.567	0.647	0.507	0.607
Counting	0.493	0.507	0.480	0.500
Average	0.544	0.535	0.533	0.549

either overlooking dominant visual cues that indicate orientation or mistakenly assigning equal importance to all elements in the scene. Underlying these issues may be a lack of focused training on discerning object's orientation especially vehicle's orientation, which causes the MLLM to underperform in scenarios where human observers would rely on subtle contextual clues to determine directionality. This gap between the model's interpretation and human perspective underscores the challenge in encoding and analyzing orientation within complex visual contexts.

5. **OCR VH**: OCR hallucination often stems from a complex interplay of factors. A multi-modal large language model (MLLM) falters when faced with unconventional visual scenarios, such as irregular character spacing, similar-looking letters, vertical arrangement of characters, or interference from nearby visual elements. These issues are compounded when the text contains intentional misspellings, typos, or uncommon character combinations that the system attempts to autocorrect based on standard language patterns. The ability to accurately recognize characters is further challenged by the influence of adjacent characters and the overall visual context.

6. **Size VH**: Size comparison in images requires the multi-modal large language model (MLLM) to recognize and accurately compare objects based on visual scale. Challenges in this task arise from factors like perspective, distortions, overlapping of objects, intricate patterns, complex backgrounds, discrepancies in expected real-world proportions, and excessive focus on foreground objects that causes the size of enormous background objects to be overlooked. Especially, when conducting a comparison among multiple objects of similar types and sizes, with other different types of objects in the scene that disturb the MLLM's attention, the task becomes even more challenging. The MLLM misjudges which object is larger or might rank objects incorrectly in terms of size.

7. **Position VH**: A multi-modal large language model (MLLM) encounters difficulties in accurately assessing the spatial positioning relationship between objects within an image. This is exacerbated in scenarios where objects are placed in non-linear configurations, such as circular layouts, which can disrupt the model's ability to apply standard left-to-right reading patterns. Moreover, when the objects are set against complex or similar backgrounds that lack clear demarcation lines, the MLLM's spatial parsing capabilities can be further compromised. Contributing factors such as overlapping objects, inconsistent scaling, and deceptive perspective or angled viewpoints also intensify this challenge. In addition, the presence of shadows or uneven lighting may cast ambiguity on the object's precise location, thereby leading to misinterpretation. Such spatial hallucination can be attributed not just to the inherent complexity of object arrangement but also to the MLLM's processing of visual cues that inform depth, orientation, and the relationship of elements within the image space.

8. **Counting VH**: A multi-modal large language model (MLLM) has difficulty in accurately counting the quantity of specific elements or attributes in an image, especially when the objects are closely packed, overlap, are of different sizes, are partially visible, or have varying orientations. The task becomes increasingly complex when attempting to count specific subtype objects within a larger type category. Such conditions can lead the model to overlook certain objects, mistakenly merge similar items, or misinterpret the image data, thereby providing an inaccurate counting of numbers, amounts, or values.

B Additional Results

B.1 Inter-annotator Agreement

Because annotating the whole benchmarks relies on heavy human labor, we randomly select 100 VH instances from our YNQ benchmark to measure the inter-annotator agreement. Specifically, four independent human annotators label these 100 VH instances. Based on these labeling results, we report Fleiss' Kappa (κ) to indicate the inter-annotator agreement. We obtain $\kappa = 0.958$ ($0.81 \leq \kappa \leq 1.00$), showing a very high degree of agreement.

B.2 More MLLMs Evaluation on VHTest

We evaluate 4 additional MLLMs on the YNQ benchmark: the Gemini-pro-vision (Gemini) (March 2024), ShareGPT4V-13b (Chen et al., 2023b), InstructBLIP-13b (Dai et al., 2023), and Qwen-VL-Chat-7b (Bai et al., 2023), as shown in Table 7. Together with Table 3, GPT-4V shows the highest average accuracy of 0.611 among these 7 MLLMs.

B.3 Qualitative Results on Fine-tuning

To better elucidate the efficacy of fine-tuning for mitigating the visual hallucinations within MLLMs,

Generated Image

Question:
Describe the shape of the banana in the picture.

LLaVA-1.5 13b (before fine-tuning):
The banana in the picture is shaped like a heart.

LLaVA-1.5 13b (after fine-tuning):
The banana in the picture is curved.

Generated Image

Question:
Could you identify and locate the chair with the largest and the second largest size in the picture?

LLaVA-1.5 13b (before fine-tuning):
In the image, there are several chairs of different sizes and styles. The chair with the largest size is located in the middle of the room, while the chair with the second largest size is situated on the right side of the room.

LLaVA-1.5 13b (after fine-tuning):
The chair with the largest size is the one in the center of the room, which is a giant wooden chair. The second largest chair is located to the left of the giant chair, and it is a large wooden chair as well.

Figure 8: Qualitative results on mitigating visual hallucinations of LLaVA-1.5.

we report some responses from the LLaVA-1.5 before and after fine-tuning, as shown in Figure 8. In these cases, before fine-tuning, LLaVA-1.5 misrepresented the shapes of objects by assigning irrelevant shapes to them, i.e., asserting that the banana in Figure 8 is heart-shaped; LLaVA-1.5 was unable to precisely discern the relative size of objects, as demonstrated in Figure 8 where it wrongly pinpointed the second-largest chair to the right side of the image. Fortunately, the fine-tuned LLaVA-1.5 mitigates these flaws. As shown in Figure 8, LLaVA-1.5 can accurately describe the curved shape of the banana and correctly pinpoint the two largest chairs in the image. We observe such improvements in response quality across the entire testing set, indicating the effectiveness of our method.

C Prompts Used in VHTest

We use the following prompt in Figure 9 for text description generation in Step II, based on a successful initial VH instance.

Prompt for text description generation using a successful initial VH instance:

[image]
Question: [question]
Multi-modal LLM (MLLM)'s response: [testing MLLM's hallucinated response]
Detected hallucination in the response: [detected hallucination in the response]

Focus on the following elements: image, question, MLLM's response, and the detected hallucination in the response. I'm trying to identify visual hallucinations in the MLLM associated with its visual process. Through the specific example, are there any general types of hallucination modes you notice the MLLM makes, or any visual features that MLLM fails to encode, ultimately leading to the errors in MLLM's response? Try to give hallucination modes that are specific enough that someone could enable consistent reproduction of images and corresponding questions. Please try to include as many general hallucination modes as possible. These hallucination modes will be used later to generate images or videos. In your hallucination modes, please clearly explain why the this hallucination mode would cause difficulties for the vision encoder of MLLM to understand images related to this hallucination mode. I will further use this reason to precisely generate images that match the hallucination mode and are able to mislead the MLLM. Please encapsulate the essence of the examples provided, summarize as many as possible and stick to the examples.

Figure 9: Full prompt to generate a text description based on a successful initial VH instance.

We utilize the following prompt in Figure 10 for text description generation in Step II, given an unsuccessful initial VH instance.

Prompt for text description generation using an unsuccessful initial VH instance:

[image]
Question: [question]
A hypothetical response from the multi-modal LLM (MLLM): [a hypothetical hallucinated response]
Detected hallucination in the hypothetical response: [detected hallucination in the hypothetical response]

Focus on the following elements: image, question, MLLM's hypothetical response, and detected hallucination in the hypothetical response. I'm trying to identify visual hallucinations in the MLLM associated with its visual process. Through the specific example, are there any general types of hallucination modes you notice the MLLM makes, or any visual features that MLLM fails to encode, ultimately leading to the errors in MLLM's response? Try to give hallucination modes that are specific enough that someone could enable consistent reproduction of images and corresponding questions. Please try to include as many general hallucination modes as possible. These hallucination modes will be used later to generate images or videos. In your hallucination modes, please clearly explain why the this hallucination mode would cause difficulties for the vision encoder of MLLM to understand images related to this hallucination mode. I will further use this reason to precisely generate images that match the hallucination mode and are able to mislead the MLLM. Please encapsulate the essence of the examples provided, summarize as many as possible and stick to the examples.

Figure 10: Full prompt to generate a text description based on an unsuccessful initial VH instance.

We use the following prompt in Figure 11 for text description integration in Step II.

Prompt for text description integration:

I want you to focus on the [VH mode]: [the definition of VH mode]. Try to summarize the [VH mode] text description in ONE paragraph less than 200 words to explain the causes of the hallucination based on the information below.

[N text descriptions]

Figure 11: Prompt to integrate N text descriptions into a final text description for a VH mode.

We use the following prompt in Figure 12 for a text-to-image generative model to generate a VH image in Step III.

Prompt for VH image generation:

Generate an image that reflects the given hallucination mode. After your image generation, we will manually generate questions based on the image.

Your MAIN GOAL is to ensure that the image aligns with the hallucination mode well so that querying the multi-modal LLM with the combination of manual questions and the image can effectively induce hallucinations.

You will be evaluated on how well you actually perform. The generated image should ideally align with the hallucination mode, but there's room for creativity. Be both creative and cautious. You can try to create image with different scenes and objects that align with the hallucination mode. Moreover, when you generate images, remember you are a very clever expert in exploiting the hallucination mode. For future debugging purposes, ensure that the generated images with our manual questions MUST cause multi-modal LLMs to provide incorrect responses.

Hallucination Mode:

[Text description of a VH mode]

Figure 12: Prompt for a text-to-image generative model to generate a VH image under a VH mode.

We use the following prompt in Figure 13 for an LLM to generate prompts for text-to-image generative models. We only append extra text at the end compared with the prompt in Figure 12.

Prompt for an LLM to generate prompts for text-to-image generative models:

Generate an image that reflects the given hallucination mode. After your image generation, we will manually generate questions based on the image.

Your MAIN GOAL is to ensure that the image aligns with the hallucination mode well so that querying the multi-modal LLM with the combination of manual questions and the image can effectively induce hallucinations.

You will be evaluated on how well you actually perform. The generated image should ideally align with the hallucination mode, but there's room for creativity. Be both creative and cautious. You can try to create image with different scenes and objects that align with the hallucination mode. Moreover, when you generate images, remember you are a very clever expert in exploiting the hallucination mode. For future debugging purposes, ensure that the generated images with our manual questions MUST cause multi-modal LLMs to provide incorrect responses.

Hallucination Mode:

[Text description of a VH mode]

As an LLM, DO NOT generate images, but generate prompts for text-to-image generative models aligning with the hallucination mode. I will use your generated prompts for text-to-image generative models later.

Figure 13: Prompt for an LLM to generate prompts for text-to-image generative models under a VH mode.

D Question Templates

We provide question templates for all eight VH modes below.

Question templates for existence VH mode

1. Could you describe [object] in the picture?
2. Describe the [property] of [object] in the picture.

Question templates for shape VH mode

1. Describe the shape(s) of [object] in the picture.
2. In the photo, is [object] depicted with a conventional, typical shape?

Question templates for color VH mode

1. Describe the color(s) of [object] in the picture.
2. Describe [n] main color(s) of [object] in the picture.
3. Describe all the colors of [object] in the picture. Do not miss any color.
4. Describe the original color of [object] in the picture.

Question templates for orientation VH mode

1. From the camera/viewer's perspective, please choose the direction that is most proximate to the orientation in which [object] is facing in the picture. [Left, 45 degrees leftwards away from the camera/viewer, Away from the camera/viewer, 45 degrees rightwards away from the camera/viewer, Right, 45 degrees rightwards towards the camera/viewer, Towards the camera/viewer, 45 degrees leftwards towards the camera/viewer]
2. From the camera/viewer's perspective, please choose the direction that is most proximate to the orientation in which [object] is facing in the picture. [Left, Left and upward within the pictorial plane, Upward within the pictorial plane, Right and upward within the pictorial plane, Right, Right and downward within the pictorial plane, Downward within the pictorial plane, Left and downward within the pictorial plane]
3. From the camera/viewer's perspective, please choose the direction that is most proximate to the orientation in which [object] is facing in the picture. [Away from the camera/viewer, Towards the camera/viewer]

Question templates for OCR VH mode

1. Read the (English) character(s)/word(s)/
caption [at WHERE] in the picture.

Question templates for size VH mode

1. Could you identify and locate [object] with the largest/smallest size in the picture?
2. Could you identify and locate [object] with the tallest/shortest height in the picture?
3. Could you identify and locate [object] with the second largest/smallest size in the picture?
4. Could you identify and locate [object] with the largest/smallest and the second largest/smallest size in the picture?
5. Could you identify and locate [object] with the tallest/shortest and the second tallest/shortest height in the picture?
6. Could you identify and locate the 2/3/4 largest/smallest [object] in the picture?
7. Which one is larger/smaller/taller/shorter, [object A] or [object B]?
8. What is the largest [object] in the picture?

Question templates for position VH mode

1. Is [object A] to the left/right of/on the top of/being placed on/above/under/inside/outside [object B]?
2. Which one is on the left/right/top, [object A] or [object B]?
3. Which one is closer to/further from the camera or viewer perspective, [object A] or [object B]?
4. Which one is positioned under/positioned above/-closer to [reference object], [object A] or [object B]?
5. Regardless of the positional relationship of the actual scene in reality, from the camera's perspective in the photo, is [object A] positioned above/below [object B]?
6. Is [object A] on and touching [object B]?

Question templates for counting VH mode

1. How many [object] are depicted/there/visible/can be seen [at WHERE] in the picture?

E Special Cases in Step III

In existence VH mode, a human worker uses the prompt in Figure 14 to find non-existent objects in a VH image with the aid of an MLLM, such as GPT-4V. The non-existent objects found in this way are more likely to trigger VHs in MLLMs.

Prompt to find names of non-existent objects in a VH image in the existence VH mode

[image]

First, list all the names of objects in the picture. And then return some names of objects according to Requirement 1 or Requirement 2:

Requirement 1: Objects you associate with the scene that should be there but are not actually there in the picture.

Requirement 2: Objects that look similar to an object in the picture but do not actually exist in the picture.

Figure 14: Prompt to find names of non-existent objects in a VH image in the existence VH mode.

Table 8: Hyperparamters for fine-tuning LLaVA-1.5.

Hyperparameter	Setting
Batch size	16
Vision encoder lr	4e-6
Projection lr	4e-6
LLM lr	4e-6
Lr schedule	cosine decay
Lr warmup ratio	0.03
Weight decay	0
Epoch	1
Optimizer	AdamW
DeepSpeed stage	3

F Details of Fine-tuning Experiments

F.1 Hyperparamters

When fine-tuning LLaVA-1.5 on the training set of VHTest dataset, we follow LLaVA-1.5's limited task-specific fine-tuning setting. Additionally, we unfreeze the vision encoder. The hyperparameters are shown in Table 8.

F.2 Pre-processing of the Training Set

As mentioned in (Liu et al., 2023), the short-form data overfit an MLLM behaviorally to short-form answers. We follow this work to do pre-processing on both counting and position VH modes. To be more specific, for counting VH mode instances, we use the prompt below in Figure 15 for an LLM (e.g., GPT-4) to modify every number reference answer into a reference sentence. As for position VH mode instances, we add the sentence "Answer the question using a single word or phrase." after all the questions in every position VH instance.

Prompt for reference answer modification in counting VH mode

Don't talk nonsense. Turn the ground-truth numbers into extremely correct, extremely accurate, and only a little of diverse sentences based on the corresponding question. Sentences should be independent of each other, with each sentence occupying one line.

Figure 15: Prompt for reference answer modification in counting VH modeduring pre-processing of training set.