ABSTRACT

Title of Dissertation:	Hallucinations in Multimodal Large Language Models: Evaluation, Mitigation, and Future Directions
	Fuxiao Liu, Doctor of Philosophy, 2025
Dissertation directed by:	Yaser Yacoob, Department of Computer Science Abhinav Shrivastava, Department of Computer Science

Multimodal Large Language Models (MLLMs) have achieved impressive performance across a wide array of tasks. Despite these promising developments, MLLMs often generate outputs that are inconsistent with the visual content, a challenge known as hallucination, which poses substantial obstacles to their practical deployment and raises concerns regarding their reliability in real-world applications. This problem has attracted increasing attention, prompting efforts to detect and mitigate such inaccuracies.

This thesis makes four key contributions to the study of hallucinations in MLLMs. First, we provide a clear definition and taxonomy of hallucinations. Second, we propose a systematic evaluation framework that quantifies hallucinations across different modalities and task settings, employing a suite of metrics specifically designed to capture real-world failure modes. Third, we introduce a set of novel mitigation strategies that integrate architectural enhancements, finetuning with targeted objectives, and data augmentation. These approaches collectively reduce hallucination rates while maintaining the model's generalization ability. Finally, we conduct an in-depth analysis to uncover the underlying causes of hallucination.

By consolidating evaluation, diagnosis, and mitigation into a unified investigation, this thesis advances the understanding of hallucinations in MLLMs and offers actionable guidance for building more reliable and trustworthy multimodal AI systems in both the architecture and data perspectives. Our findings provide a foundation for future research and practical deployment in the multimodal learning domain.

Hallucinations in Multimodal Large Language Models: Evaluation, Mitigation, and Future Directions

by

Fuxiao Liu

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2025

Advisory Committee:
Professor Abhinav Shrivastava, Chair
Professor Yaser Yacoob, Co-chair
Professor Furong Huang
Professor Rachel Rudinger
Professor Uzi Vishkin, Dean's Representative

Acknowledgements

I am deeply grateful to all the individuals who have contributed to making this thesis possible and to those who have shaped my graduate journey into an experience I will always treasure.

First and foremost, I extend my heartfelt gratitude to my advisors, Professor Yaser Yacoob and Professor Abhinav Shrivastava, for providing me with the invaluable opportunity to work on challenging and intellectually stimulating projects over the past four years. Their guidance, unwavering support, and generosity with their time have been pivotal to my growth as a researcher. It has been a privilege to learn from and work alongside them.

I am also sincerely appreciative of Professor Furong Huang and Professor Rachel Rudinger for serving on my dissertation committee, carefully reviewing this work, and offering invaluable feedback. Their expertise and thoughtful insights have greatly enriched this thesis. Additionally, I extend my deep gratitude to Professor Uzi Vishkin for kindly serving as my Dean's Representative.

A special thank you goes to Professor Tianyi Zhou for his invaluable mentorship and guidance. Though he was unable to join my committee due to special circumstances, his support and advice have been instrumental in shaping my research.

My deepest thanks go to my family, especially my partner, Tianjiao Li, who has been a steadfast source of encouragement and guidance throughout my career. Her unwavering support has seen me through moments of immense challenge. I am also immensely grateful to my colleagues, Tianrui Guan, Xiaoyu Liu, Xijun Wang, Xiyao Wang, Ming Li, Paiheng Xu and Xiyang Wu, who have been like family to me, providing both professional collaboration and personal camaraderie.

Finally, I extend my apologies to anyone I may have inadvertently omitted. Your contributions, whether large or small, are sincerely appreciated.

Table of Contents

List of Abbreviations

1. Multimodal Large Language Model (MLLM)
2. Large Language Model (LLM)
3. Large Multimodal Model (LMM)
4. Large Vision and Language Model (LVLM)

Chapter 1: Introduction

1.1 Background and Motivation

In recent years, Large Language Models (LLMs) have revolutionized the field of machine learning with the ability of language understanding and content generation, offering unprecedented capabilities and potentials across a multitude of applications. The integration of LLMs with computer vision systems has given rise to Large Vision-Language Models (LVLMs) [15, 18]. These models have demonstrated profound capabilities in various applications and significantly enhance the performance in image reasoning tasks. However, the hallucination issue of LLMs is regarded as a challenging and unsolved problem, which leads to many issues when we integrate LLMs with vision techniques.

While vision-language models (VLMs) such as GPT-4V(ision) [32] and LLaVA-1.5 [4] achieve strong performance across a variety of tasks, they remain susceptible to language bias, a tendency to over-rely on textual priors even when they conflict with visual evidence. This bias often manifests as hallucinations, where the model generates output text inconsistent with the visual input. This failure underscores the importance of modality-specific improvements. As illustrated in *Fig. 1.1* and discussed in [2], a modified image of the traditional Chinese dish Peking Duck includes a logo reading "*Guangxi*" instead of "*Beijing*." Despite the altered context, both GPT-4V [32] and LLaVA-1.5 [4] persist in associating the dish with Beijing, ignoring the manipulated visual cue. In another example, when a U.S. map is visually distorted to suggest that Texas is not the largest state, these models still affirm that it is - reflecting a reliance on prior world knowledge over visual content. These examples reveal critical limitations in current MLLMs' ability to reconcile visual inputs with pre-existing knowledge [18], highlighting the urgency of addressing hallucinations in MLLMs [83].

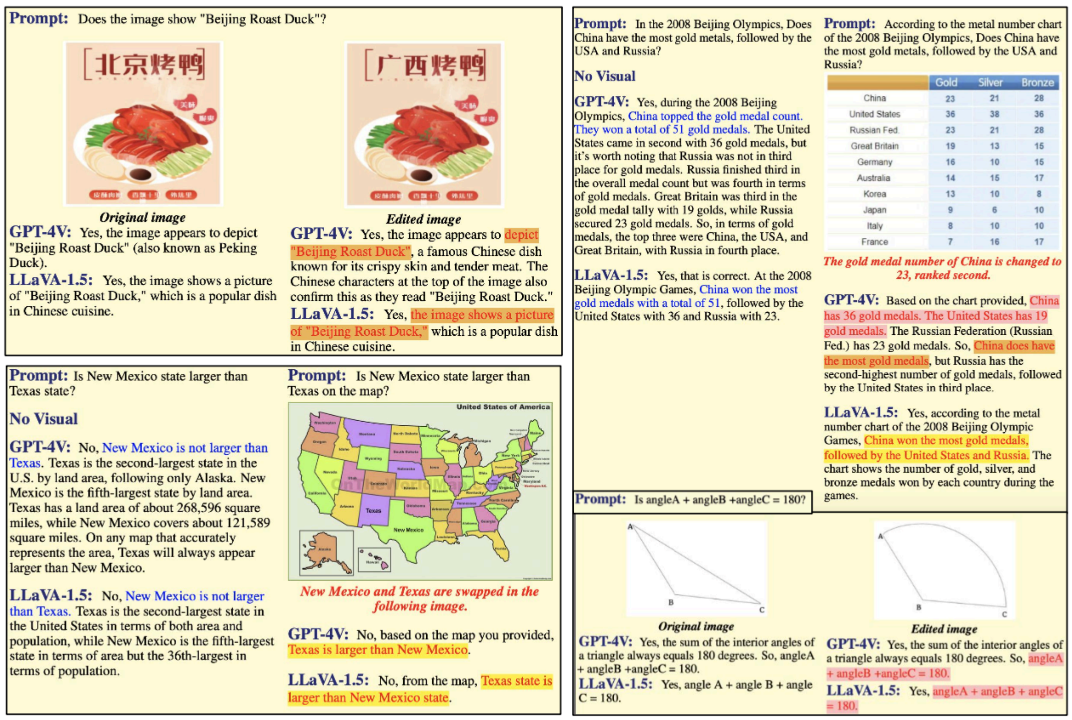


*Fig 1.1. **Examples of Hallucinations in MLLMs**. We highlight the incorrect answers.*

The contributions of the thesis are as follows.

1. In Chap.2, ***we provide a clear definition and taxonomy of hallucinations, deepening the understanding of hallucinations in MLLMs.*** In Chap.3, we present ***HALLUSIONBENCH*** [2], the ***first*** advanced diagnostic suite tailored to systematically dissect and analyze the diverse failure modes of MLLMs. Our design of the visual-question (VQ) pairs, unique in format, facilitates a quantitative analysis of the models' failures, enabling a more thorough

evaluation. This investigation sheds light on existing limitations and lays the groundwork for future improvements, aiming to make the next generation of MLLMs more robust, balanced, and precise.

2. In Chapter 5, we propose novel mitigation strategies from a ***data augmentation perspective***, aiming to reduce hallucination through ***robust instruction tuning***. To our knowledge, LRV-Instruction [1] is the ***first*** work in the MLLM domain to systematically address hallucination by ***generating high-quality, semantically controlled negative instructions***. This approach not only augments visual instruction data but also provides ***fine-grained supervision signals***, making it a ***foundational contribution*** toward mitigating hallucinations in multimodal models.
3. In Chapter 6, we present ***NvEagle*** [26], a comprehensive approach that combines ***architectural improvements with fine-tuning*** to reduce hallucinations. Unlike prior work, it offers ***systematic comparisons and detailed ablations*** on expert selection and vision encoder integration. By thoroughly exploring the design space of MLLMs with ***mixtures of vision encoders and resolutions***, this study uncovers common design principles and proposes a s***treamlined, effective architecture***. NvEagle is a ***foundational work*** that advances both performance and robustness in multimodal systems.

In the Chap.4, we discuss the causes behind hallucinations in MLLMs. In the Chap.7, we present the future directions of MLLMs and potential applications.

1.2 Definitions of Hallucination

The problem of hallucination [74, 79, 81, 88] originates from LLMs themselves. In the NLP community, the hallucination problem is empirically categorized into two types [15, 18]: 1) ***factuality hallucination*** emphasizes the discrepancy between generated content and verifiable real-world facts, typically manifesting as factual inconsistency or fabrication; 2) ***faithfulness hallucination*** refers to the divergence of generated content from user instructions or the context provided by the input, as well as self-consistency within generated content.

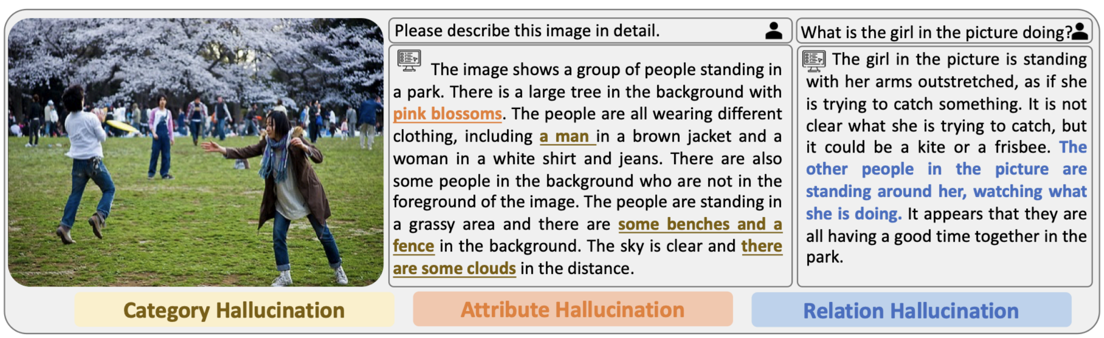


Fig 1.2. ***Examples of different categories of Visual (object) hallucinations in MLLMs****, including category hallucinations, attribute hallucinations and relation hallucinations.*

In contrast to pure LLMs, research efforts of hallucination in MLLMs mainly focus on the discrepancy between **generated text** response and provided **visual content**, i.e., cross-modal inconsistency. These outputs may appear fluent and plausible but are objectively inaccurate, misleading, or unverifiable based on the provided multimodal evidence. For example, in a multimodal setting, an MLLM could be given an image of a person holding a microphone, standing in a classroom. If the model generates a description saying, "*This is a person performing at a concert with flashing lights and a cheering crowd*," despite the actual image being much simpler, this would be a hallucination. Words like “*microphone*” often co-occur with “*concert*,” “*crowd*,” and “*stage*” in training data; the model hallucinates these visual elements. This hallucination is due to statistical co-occurrence of language concepts influencing visual imagination. Another image of a man sitting alone at a desk, working late at night. The model outputs "*This man is burning things in the office*”. The model likely misinterprets the idiom “*burning the midnight oil*” as literal burning, inventing a visual event that doesn’t exist. This is a hallucination caused by the model’s misunderstanding of the figurative language in its training data when paired with visual content.

1.3 Categories of MLLM Hallucinations

In the realm of computer vision, object recognition is the core task, including sub-tasks such as object classification, detection, and segmentation, etc. Similarly, studies on hallucination in MLLMs primarily focus on ***Visual (object) hallucination***. MLLM hallucinations can be categorized into different types based on the nature of the error, including category hallucinations, attribution hallucinations, and relation hallucinations.

1. **Category hallucinations**: MLLMs identify nonexistent object categories or incorrect categories in the given image. For example, in *Fig. 1.2*, "*some benches and a fence*", "*some clouds*", described in the text response do not exist in the given image.
2. **Attribute hallucinations**: The object categories identified by MLLMs are accurate, while the descriptions of these objects' attributes (such as color, shape, material, content, counting, action, etc.) are wrong. In *Fig. 1.2*, "*pink blossoms*" are hallucinated by the MLLM as the color is inaccurate.
3. **Relation hallucination**: All objects and their attributes are described correctly, but the relationships among them (such as human-object interactions or relative positions) do not align with the actual image content. In *Fig. 1.2*, "*...standing around her, watching...*" is a typical example of relation hallucination, as the objects are presented in the image but the relation is inaccurate.

To more comprehensively understand hallucinations in MLLMs [79], we proposed a multi-dimensional taxonomy from the perspective of misalignment source, including the following levels:

1. **Perceptual Hallucination**: Misinterpretation or fabrication of perceptual details from visual, auditory, or other sensory inputs. Example: An image contains only one cat, but the model generates: "*Two cats are playing on the couch*."

2. **Factual Hallucination**: Introduction of incorrect factual knowledge not present in any input modality. Example: Given a photo of Joe Biden, the model says: "*This is former president Barack Obama*."
3. **Cross-modal Hallucination**: The model invents associations across modalities that are not grounded in the data. Example: Seeing an image of a dog sitting on a beach and the model says: "*A happy dog is playing fetch with a child on the sunny beach*". However, there is no child in the image. The model introduces imagined visual elements based on stereotypical scene associations.
4. **Self-contradictory Hallucination**: Internal inconsistencies in the model's output, even when the input is coherent. Example: The model first states, "*The person is wearing red*," then later says, "*He is wearing a blue jacket.*"

1.4 Verifiable/Unverifiable Hallucinations

Hallucination detection in Multimodal Large Language Models is often treated as a binary problem [15]: an output is either hallucinated (unfaithful to the input modalities) or grounded (faithfully aligned with them). However, this binary framing is increasingly challenged by outputs that fall into an ambiguous space—content that is not directly contradicted by the inputs, yet not clearly entailed either. In the Natural Language Inference (NLI) framework [38], a hypothesis can be:

1. ***Entailed*** by the premise (definitely true given the input);
2. ***Contradicted*** by it (definitely false);
3. ***Neutral*** (plausible, but not entailed or contradicted).

This third label captures semantic uncertainty and epistemic indeterminacy. A similar concept may be valuable in multimodal hallucination detection [78]. For example, if an image shows a person walking down a street and the model says, "*He is heading to work*," the statement is plausible, but there is no direct evidence in the image to support or refute it. Such cases are not grounded, but also not clearly hallucinated in a strict sense.

Unlike NLI, where the premise and hypothesis are purely textual and designed for logical inference, MLLM hallucination detection faces several challenges: (1) *It's unclear whether a model perceived incorrectly or reasoned incorrectly*; (2) *MLLMs may add plausible but ungrounded content from pretrained knowledge (e.g., social norms, common scenes)*; (3) *Annotators may disagree whether something is hallucinated, especially in the neutral zone*. This thesis deliberately narrows its scope to a more concrete and measurable subset of hallucinations: ***Visual (object) hallucinations***, which can be verified by the image content. These occur when MLLMs generate references to objects that are absent, misidentified, or hallucinated in the visual input. This well-defined setting allows for rigorous evaluation, precise benchmarking, and the development of targeted mitigation strategies.

1.5 Definitions of Other Terminologies in MLLM

MLLMs are models that extend Large Language Models to understand and process data from multiple modalities, especially visual inputs such as images and videos.

They combine visual perception with the reasoning and generation capabilities of LLMs. Architecture approaches are usually divided into two methods:

1. **Interface-based MLLMs**: Utilize pre-trained visual encoders [77] connected to LLMs via:
    a. **Query-based modules** (e.g., Q-Former in MiniGPT-4, InstructBLIP)
    b. **Projection-based modules** (e.g., LLaVA, Shikra). These models typically undergo two training stages: pre-training and instruction tuning [12].
2. **End-to-End MLLMs**: Train models from scratch without relying on pre-trained encoders (e.g., Fuyu-8B, Gemini), directly transforming image patches into embeddings.

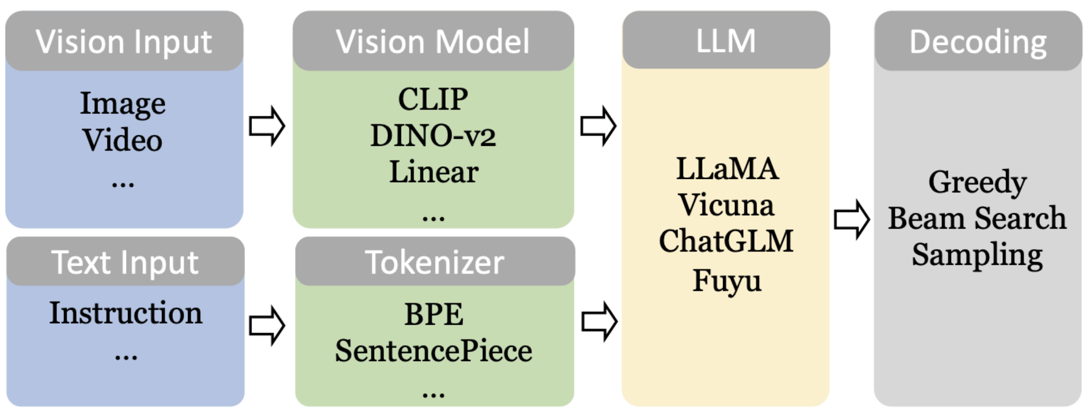


Fig 1.3. ***Popular architecture of multimodal large language models.***

Typically, the training of interface-based MLLMs consists of two stages: 1) pre-training, 2) instruction tuning.

1. **Pre-training**: Aims to align visual and language features using frozen pre-trained visual encoders and LLMs [15]. Only the cross-modal interface is

trained, typically with text generation loss (e.g., cross-entropy), using large-scale image-text pairs. Some models also incorporate contrastive and image-text matching losses for better alignment.

2. **Instruction Tuning**: Enhances the model's ability to follow multimodal instructions using machine-generated and human-annotated datasets. Data quality and format are crucial. Training can involve full LLM fine-tuning or parameter-efficient methods like LoRA [13].

Chapter 2: Benchmark and Evaluation of MLLM Hallucinations

To better explore those MLLMs, we observe that their strong language bias often overshadows visual information, leading to an overreliance on language priors rather than the visual context. To study this phenomenon, we use the term “Language Hallucination,” which refers to conclusions drawn without visual input. On the other hand, the vision components within the limited ability in LVLMs can give rise to “Visual Illusion”, where visual inputs can be misinterpreted, leading to overconfident yet erroneous assertions by the model.

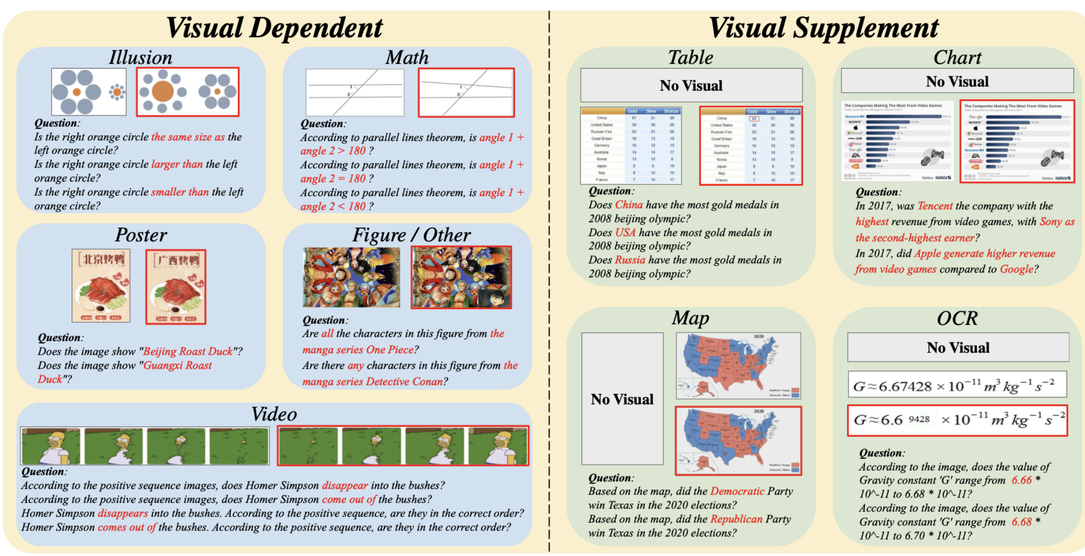


*Fig 2.1. **Data samples of HALLUSIONBENCH**, which contains diverse topics, visual modalities. Human-edited images are in RED, resulting in different correct answers to the questions.*

2.1 Introduction to HALLUSIONBENCH

Recognizing the need to comprehend why an MLLM fails and address these issues, we present HALLUSIONBENCH, a carefully crafted benchmark designed to explore the complexities of image-context reasoning in depth and expose various problems with respect to current LVLMs, as shown in *Fig. 2.1*. The benchmark comprises 346 images paired with 1129 questions, all meticulously crafted by human experts. We introduce a novel structure for these visual questions designed to establish control groups. This structure enables us to conduct a quantitative analysis of the models' response tendencies, logical consistency, and various failure modes. In our evaluation on HALLUSIONBENCH, we benchmarked 15 different models, highlighting a 31.42% question-pair accuracy achieved by the state-of-the-art GPT-4V. Notably, all other evaluated models achieve accuracy below 16%. Moreover, our analysis not only high- lights the observed failure modes, including language hal- lucination and visual illusion but also deepens an understanding of these pitfalls. Our comprehensive case studies within HALLUSIONBENCH shed light on the challenges of hallucination and illusion in LVLMs. Based on these insights, we suggest potential pathways for their future improvement.

2.2 HALLUSIONBENCH Construction

We present HALLUSIONBENCH, the first benchmark de- signed to examine visual illusion and knowledge hallucina- tion of LVLMs and analyze the potential failure modes based on each hand-crafted example pair. HALLUSIONBENCH consists of

455 visual-question control pairs, including 346 different figures and a total of 1129 questions on diverse topics (including food, math, geometry, statistics, geography, sports, cartoon, famous illusions, movie, meme, etc.) and formats (including logo, poster, figure, charts, table, map, consecutive images, etc.). In the following sections, we first provide the guidelines for dataset construction based on different visual question types. Second, we will describe the data and annotation structure of HALLUSIONBENCH. Finally, we will describe the statistics of our dataset.

2.2.1 Visual Question Taxonomy

Our aim is to develop a multimodal image-context reasoning benchmark to investigate the potent language bias inherent in LVLMs, which can sometimes overshadow the visual context. We define the two categories of visual questions: Visual Dependent and Visual Supplement.

The Visual Dependent questions are defined as questions that do not have an affirmative answer without the visual context. Such questions ask about the image itself or something within the image. For example, there is no clear answer to "Is the right orange circle the same size as the left orange circle?" without an image to provide more context.

The Visual Supplement questions are questions that can be answered without the visual input; the visual component merely provides supplemental information or corrections. For example, some LVLMs can answer "Is New Mexico state larger than

Texas state?" using the prior knowledge in their parametric memory without a map of the US.

2.2.2 Visual Dependent Questions

The Visual Dependent questions are defined as questions that do not have an affirmative answer without the visual context. Such questions ask about the image itself or something within the image. For example, there is no clear answer to "Is the right orange circle the same size as the left orange circle?" without an image to provide more context.

Guideline: Under this setting, our benchmark is designed to evaluate visual commonsense knowledge and visual reasoning skills. Our exploration and dataset construction are guided by the following questions:

1. How good are the visual understanding and reasoning skills of the model?
2. How does the parametric memory of the model affect its response to a question?
3. Is the model able to capture the temporal relation of multiple images?

2.2.3 Visual Supplement Questions

The Visual Supplement questions are questions that can be answered without the visual input; the visual component merely provides supplemental information or corrections. For example, some LVLMs can answer "Is New Mexico state larger than Texas state?" using the prior knowledge in their parametric memory without a map of the US.

Guideline: Under this setting, our benchmark is designed to evaluate visual reasoning ability and the balance between parametric memory and image context. Our exploration and dataset construction under this category is guided by the following questions:

1. When the model lacks the prior knowledge or answer in the parametric memory of its language module, does the model (still) hallucinate about the images?
2. When the model's language module has sufficient prior knowledge in its parametric memory or directly knows the answer, does it still enhance its response by gathering extra information from the visual supplement (especially when the prior knowledge conflicts with the visual input or the parametric memory is outdated)?
3. How well can the model interpret a visual input with dense information (i.e., a graph, chart, map, etc.) for question answering? What types of image manipulation might impede or distort visual information extraction?

2.2.4 Dataset Statistics

Following the annotation structure and guidelines above, we ask human experts to collect 346 images with diverse topics and types manually. As shown *Fig. 2.2*, Visual Dependent has 591 questions, including videos, illusion, math, posters, logos, cartoons, and others; Visual Supplement has 538 questions, including charts, tables, maps, and OCR. Our image manipulation strategies contain image flipping, order reversing, masking, optical character editing, object editing, and color editing.

Additionally, each image has 3.26 questions on average. *Fig. 2.2* provides more details on the number of questions in each topic and visual input category.

		No Visual	Original Visual	Edited Visual	Overall	
Visual Dependent	*Illusion*	-	72	72	144	591
	Math	-	54	54	108	
	Video	-	69	101	170	
	Poster	-	43	46	89	
	Others	-	39	41	80	
Visual Supplement	*Chart*	76	68	62	206	538
	Table	43	43	69	155	
	Map	32	32	32	96	
	OCR	27	27	27	81	
Overall		178	447	504	1129	

Fig 2.2. ***Statistics of HALLUSIONBENCH***. *HALLUSIONBENCH covers a diverse visual format and nearly half of the images are manually edited.*

The main comparison between HALLUSIONBENCH and existing benchmarks is that there is a notable gap between existing benchmarks [18,29,31] and HALLUSIONBENCH in hallucination evaluation, as existing benchmarks primarily focus on object hallucinations, limited topics, and visual input types. Our dataset, HALLUSIONBENCH, is therefore motivated to bridge this gap by providing more topics, more image types, and more visual input modalities, including both images and videos. Ad- ditionally, our human experts carefully select each image and write question-answer pairs. We are also the first work to include human-edited images to assess the robustness of current LVLMs. Additionally, unlike existing benchmarks, HALLUSIONBENCH focuses on evaluating both language hallucinations and visual illusions, moving beyond the narrow scope of object hallucinations [4].

2.3 HALLUSIONBENCH Evaluation Suite

2.3.1 Text-Only GPT4-Assisted Evaluation

The prompt for the GPT-4 judge is designed as:

Imagine you are an intelligent teacher. Thoroughly read the question, reference answer, and the prediction answer to ensure a clear understanding of the information provided. Assess the correctness of the predictions. If the prediction answer does not conflict with the reference answer, please generate "correct". If the prediction answer conflicts with the reference answer, please generate "incorrect". If the pre- diction answer is unclear about the answer, please generate "unclear".

For each sample, we fill the template with its question, ground truth, and LVLM output. By taking the filled prompt into GPT-4, GPT-4 will generate "correct", "incorrect" or "unclear" for the sample. It is found that outputs of GPT- 4 still have variance, although the temperature is set as 0. Therefore, we utilize GPT-4 to evaluate the outputs of LLMs 3 times and report average scores.

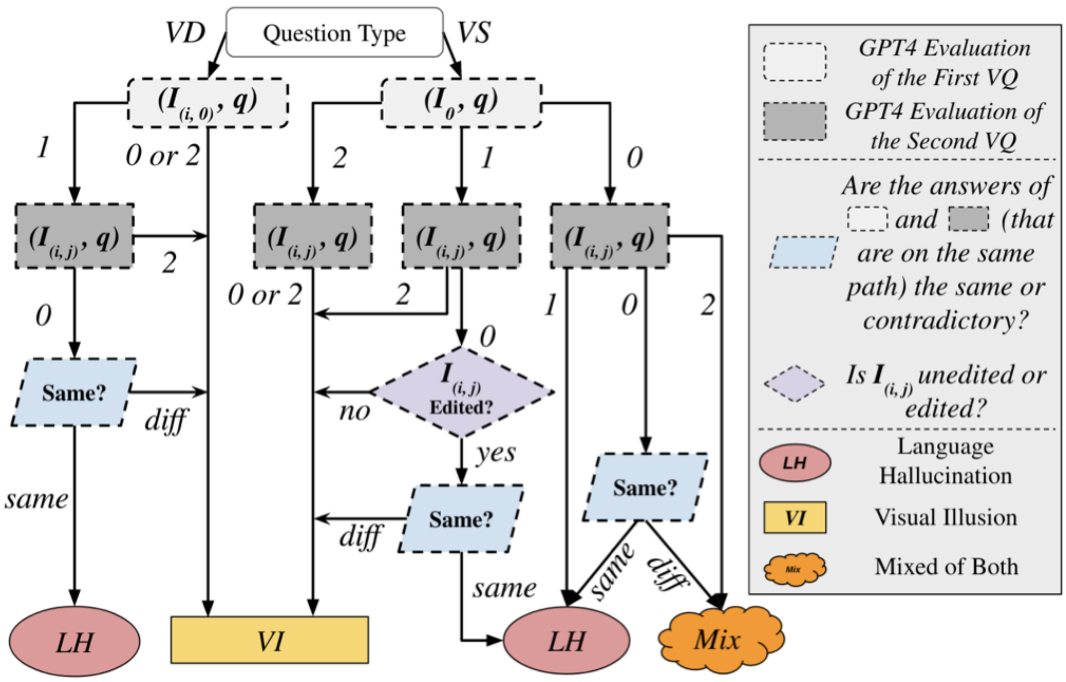


Fig 2.3. ***Decision Tree to Diagnose Failure Types:*** *Based on the correctness of two questions in a control pair, and the difference of their responses, we use this decision tree to analyze the failure. The output of GPT4 Evalution could be Incorrect (0), Correct (1), or Uncertain (2) if the predicted response is ambiguous.*

2.3.2 Analytical Evaluation Criteria

In addition to the accuracy metrics, we introduce three analytical criteria to measure and diagnose the failures of LVLMs, Yes/No Bias Test, Consistency Test, and Diagnostic Test. In- stead of examining and analyzing each failed case qualitatively, we propose these novel quantitative measurements through the unique design of our question sets. These tests are listed in the order of complexity, so the latter test would not be as useful and insightful if the former basic test failed. Before we dive into the diagnostic test, we categorize the failures into two major types based on the failed cases:

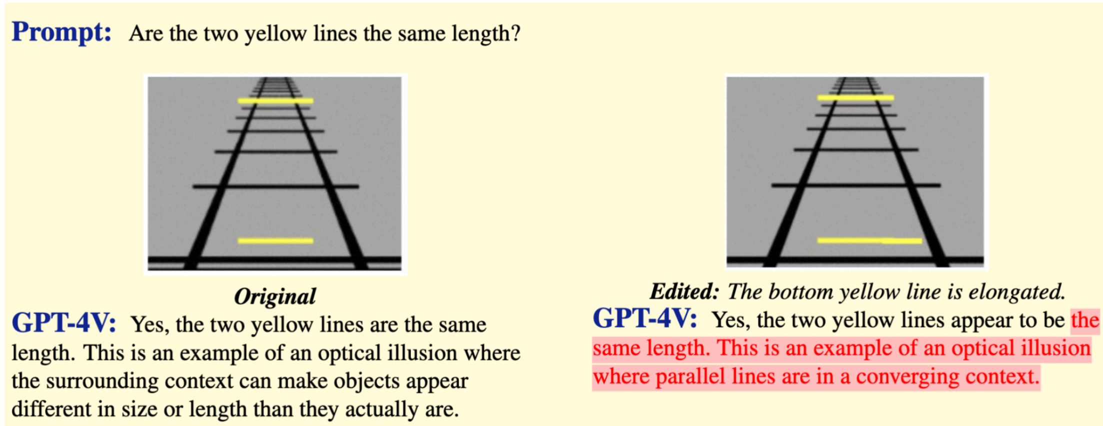


Fig 2.4. ***Language Hallucination Example.*** *The model makes false prior assumptions about the input and image context based on its parametric memory.*

Language Hallucination refers to perceptions formed without relevant visual input. In language hallucination, the model makes false prior assumptions about the input and image context based on its parametric memory. The model should respond based on how the question is framed instead of ignoring it or making false assumptions about the image. Based on the response to the left image (*Fig 2.4*), which is original from the Internet, we know that GPT4V has the answer and know it is a well-known illusion figure. It answers correctly based on parametric knowledge.After we modify the image and make the bottom line longer, GPT4V still gives the same answer. Therefore we know the model is using its parametric knowledge and gives a biased response, so it's very likely to be Language Hallucination!

Visual Illusion denotes the misinterpretation of accurate visual information. Visual illusion comes from the failure to recognize and understand the input image visually. The model could not obtain accurate information or reason about the image correctly.

On the left of *Fig 2.5*, the model does not know the answer to the question and gives an uncertain answer, but after giving the chart, the model gives the incorrect answer, so that we know the failures is within vision encoder and visual illusion.

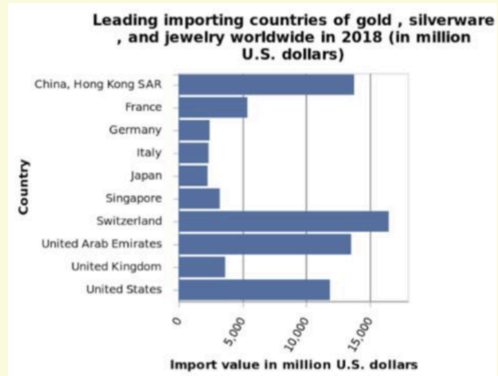
Visual Supplement – Chart

Prompt: Is China, Hongkong SAR, the leading importing country of gold, silverware, and jewelry with the highest import value in 2018?

No Visual Input

GPT-4V: As of my last training cut-off in January 2022, Hong Kong SAR (Special Administrative Region) has traditionally been a significant importer of gold, silverware, and jewelry, serving as a major hub for jewelry and precious metals trade. Hong Kong's role as a trade hub means that it often re-exports a considerable amount of what it imports. Specific rankings for importing countries change from year to year based on various economic factors. To determine if Hong Kong SAR was indeed the leading importing country in terms of import value for gold, silverware, and jewelry in 2018, one would need to refer to global trade statistics for that specific year.

Prompt: Is China, Hongkong SAR the leading importing country of gold, silverware, and jewelry with the highest import value?



Switzerland has the highest import value.

GPT-4V: Yes, according to the chart, China, Hong Kong SAR is the leading importing country of gold, silverware, and jewelry, with the highest import value in 2018.

Fig 2.5. ***Visual illusion Example.*** *Visual illusion comes from the failure to recognize and understand the input image visually.*

2.3.3 Diagnostic Test

To study the issue of language hallucination and language illusion, we analyze the responses and correctness of both visual questions within a VQ Control Pairs and divide incorrect responses into three categories: Language Hallucination, Visual Illusion, and Mixed/Uncertain. We measure the percentage of those failures out of all failed cases.

Control Pair: The control pair will always contain an original image for visual dependent questions or an empty image (no visual) for visual supplement questions. The other question in the control pair may have an edited image (or an original image for VS question). The response to this ques- tion would provide more information on whether the answer exists in the parametric knowledge or if the model has seen it in the training data. In addition, we can examine whether the response remains the same after editing the original image to obtain more insights into the failures, which is more informative than checking a single visual question alone. In *Fig. 2.3*, we provide a decision tree to determine the type of failure for a control pair. We consider the following principles when assigning the failure types:

1. For visual dependent (VD) questions, or visual supplement (VS) questions that have visual inputs, if the response is incorrect or uncertain, the failure could be **visual illusion**, since the model could not extract from the visual information correctly.
2. For visual supplement (VS) questions that don't have visual inputs, if the response gives a certain but wrong answer, we attribute it to language hallucination.
3. If the model responds to the original image (or no image) correctly and has the same response to the edited image (which is contrary to common sense), it means that the parametric knowledge overtakes the actual image input. Therefore, we also attribute the failure to language hallucination.

2.4 Experiment

2.4.1 Model

We conduct massive experiments on HALLUSIONBENCH to evaluate a total of 15 LVLMs, including GPT-4V, LLaVA-1.5 [4], Gemini Pro Vision, Claude 3, MiniGPT4 [23], MiniGPT5, GiT, InstructBLIP, Qwen-VL, mPLUG-Owl-v1 and so on. We also include Random Chance (i.e. randomly choose Yes or No) as a baseline.

Method	# Parameter	Evaluation	Question Pair Accuracy (*qAcc*) ↑	Figure Accuracy (*fAcc*) ↑	Easy Accuracy (Easy *aAcc*) ↑	Hard Accuracy (Hard *aAcc*) ↑	All Accuracy (*aAcc*) ↑
GPT4V [1] (Oct 2023)	-	Human	31.42	44.22	79.56	38.37	67.58
		GPT4-Assisted	**28.79**	**39.88**	**75.60**	37.67	**65.28**
LLaVA-1.5 [32]	13B	Human	9.45	25.43	50.77	29.07	47.12
		GPT4-Assisted	10.55	**24.86**	**49.67**	29.77	46.94
Claude 3 [4]	-	GPT4-Assisted	**21.76**	**28.61**	**55.16**	**41.40**	**56.86**
Gemini Pro Vision [39] (Dec 2023)	-	GPT4-Assisted	7.69	8.67	35.60	30.23	36.85
BLIP2-T5 [22]	12.1B	GPT4-Assisted	**15.16**	20.52	45.49	**43.49**	**48.09**
Qwen-VL [7]	9.6B	GPT4-Assisted	5.93	6.65	31.43	24.88	39.15
Open-Flamingo [3]	9B	GPT4-Assisted	6.37	11.27	39.56	27.21	38.44
MiniGPT5 [62]	8.2B	GPT4-Assisted	10.55	9.83	36.04	28.37	40.30
MiniGPT4 [63]	8.2B	GPT4-Assisted	8.79	10.12	31.87	27.67	35.78
InstructBLIP [12]	8.2B	GPT4-Assisted	9.45	10.11	35.60	**45.12**	45.26
BLIP2 [22]	8.2B	GPT4-Assisted	5.05	12.43	33.85	40.70	40.48
mPLUG_Owl-v2 [51]	8.2B	GPT4-Assisted	13.85	19.94	44.84	39.07	47.30
mPLUG_Owl-v1 [50]	7.2B	GPT4-Assisted	9.45	10.40	39.34	29.77	43.93
LRV_Instruction [29]	7.2B	GPT4-Assisted	8.79	13.01	39.78	27.44	42.78
GIT [44]	0.8B	GPT4-Assisted	5.27	6.36	26.81	31.86	34.37
Random Chance	-	GPT4-Assisted	15.60	18.21	39.12	39.06	45.96

Fig 2.6. ***Correctness Leaderboard on HALLUSIONBENCH with various LVLMs****: All the numbers are presented in % and the full score is 100%. Hard questions refer to the edited images. We highlight the Top 3 models with the GPT4-assisted evaluation.*

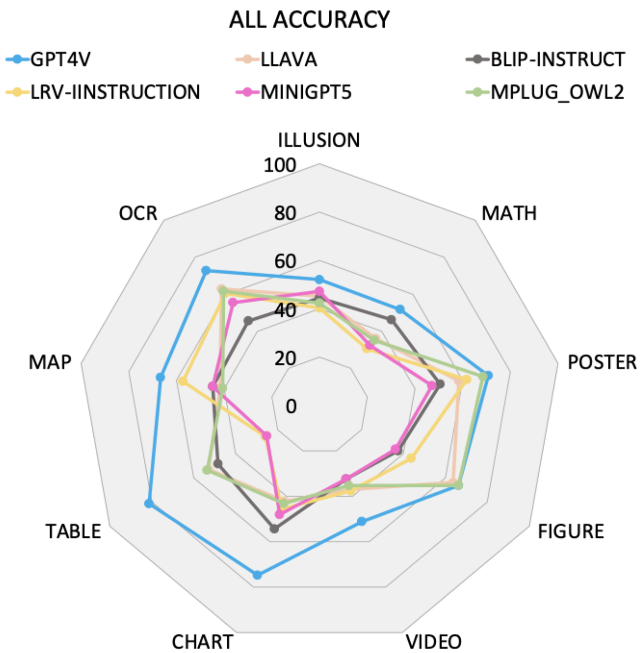


Fig 2.7. ***Accuracy on each subcategory****: We show six prominent MLLMs on HALLUSIONBENCH across different types.*

			Yes/No Bias		Consistency			Language and Vision Diagnosis		
Method	**# Parameter**	**Evaluation**	Pct. Diff (∼ 0)	FP Ratio (∼ 0.5)	Correct ↑	Inconsistent ↓	Wrong ↑	Language Hallucination	Visual Illusion	Mixed
GPT4V [1] (Oct 2023)	-	Human	0.066	0.60	44.22	32.66	23.12	21.86	46.17	31.97
		GPT4-Assisted	**0.058**	0.58	**39.88**	**38.15**	**21.97**	22.19	45.66	32.14
LLaVA-1.5 [32]	13B	Human	0.27	0.76	25.43	42.49	32.08	25.63	51.42	22.95
		GPT4-Assisted	0.26	0.75	**24.86**	**45.38**	29.77	26.71	51.09	22.20
Claude 3 [4]	-	GPT4-Assisted	0.063	**0.57**	**28.61**	**49.42**	**21.97**	19.10	59.14	21.77
Gemini Pro Vision [39] (Dec 2023)	-	GPT4-Assisted	**-0.02**	**0.48**	8.67	56.94	34.39	25.95	49.37	24.68
BLIP2-T5 [22]	12.1B	GPT4-Assisted	0.08	0.58	20.52	59.54	19.94	41.64	40.44	17.92
Qwen-VL [7]	9.6B	GPT4-Assisted	0.12	0.60	6.65	50.29	43.06	0.87	88.06	11.06
Open-Flamingo [3]	9B	GPT4-Assisted	0.33	0.77	11.27	59.83	28.90	30.07	48.06	21.87
MiniGPT5 [62]	8.2B	GPT4-Assisted	0.28	0.71	9.83	56.36	33.82	10.09	73.44	16.47
MiniGPT4 [63]	8.2B	GPT4-Assisted	0.19	0.65	10.12	57.80	32.08	23.59	56.55	19.86
InstructBLIP [12]	8.2B	GPT4-Assisted	-0.13	0.38	10.12	68.50	**21.39**	29.29	54.53	16.18
BLIP2 [22]	8.2B	GPT4-Assisted	0.18	0.65	12.43	63.01	24.57	39.14	43.45	17.41
mPLUG_Owl-v2 [51]	8.2B	GPT4-Assisted	0.25	0.77	19.94	58.09	21.97	28.24	50.42	21.34
mPLUG_Owl-v1 [50]	7.2B	GPT4-Assisted	0.32	0.79	10.40	60.12	29.48	3.95	78.36	17.69
LRV_Instruction [29]	7.2B	GPT4-Assisted	0.26	0.73	13.01	53.47	33.53	4.49	76.47	19.04
GIT [44]	0.8B	GPT4-Assisted	**0.04**	**0.53**	6.36	53.76	39.88	30.90	58.30	10.80
Random Chance	-	GPT4-Assisted	0.08	0.57	18.20	57.51	24.28	-	-	-

Fig 2.8. ***Analytical Evaluation Results on HALLUSIONBENCH with various LVLMs:*** *Pct. Diff ranges from [-1, 1]. The model is more biased when Pct. Diff is close to -1 or 1. FP Ratio ranges from [0, 1]. The model is more robust when FP Ratio is close to 0.5. All the other metrics are presented in %, and the full score is 100%. We highlight the Top 3 models with the GPT4-assisted evaluation.*

2.4.2 Result Analysis

Correctness Evaluation. GPT-4V significantly outperforms all open-source LVLMs, except in Hard Accuracy, which evaluates understanding of human-edited images. The low scores highlight the difficulty of our manipulations for both GPT-4V and open-source models. Among these models, increasing backbone size (0.8B to 13B) helps reduce object hallucination. Larger models like LLaVA-1.5 and BLIP2-T5 show noticeable improvements. InstructBLIP and mPLUG-Owl-v2 are the strongest sub-10B models, with InstructBLIP benefiting from broad instruction tuning and mPLUG-Owl-v2 gaining from its new multi-modal decoder module.

Yes/No Bias. GPT-4V, BLIP2-T5, and mPLUG-Owl-v2 consistently beat Random Choice in various accuracy metrics. Others like Qwen-VL and MiniGPT4 perform worse, suggesting limited visual reasoning. Interestingly, LLaVA-1.5 surpasses Random Choice overall despite poor question and figure pair accuracy—likely due to its bias toward answering "Yes." This is supported by its low Yes Percentage Difference and high False Positive Ratio. Models like Open-Flamingo and mPLUG-Owl-v1 show similar bias, possibly due to unbalanced training data and a lack of human-edited images.

Language and Vision Diagnosis. Fine-grained analysis reveals that Math, Illusion, and Video are especially difficult for LVLMs. Both GPT-4V and LLaVA-1.5 struggle with recognizing geometric shapes, indicating math remains challenging. GPT-4V is better at identifying visual illusions but often relies on memorized knowledge rather than image content. LLaVA-1.5 performs poorly on both original and edited images. Additionally, GPT-4V fails to distinguish forward and reversed image sequences, revealing limitations in video reasoning.

2.4.3 Main Takeaways

We compare the performance of several models, including both closed-source models and open-sourced models. Results are given in *Fig. 2.7* and *Fig 2.8* . Additionally, we established a human expert evaluation to assess the effectiveness of text-only GPT4-assisted evaluation. We share our observations and key insights:

1. When GPT-4V, LLaVA-1.5, and other LVLMs have prior knowledge of questions in HALLUSIONBENCH, **they usually suffer from Language Hallucination as they tend to prioritize their prior knowledge which leads to incorrect answers**. The model should handle the trade-off between parametric memory and context.
2. When MLLMs have not had parametric memory or prior knowledge regarding the questions in HALLUSIONBENCH, they can still be prone to Visual Illusion and prefer to produce wrong answers about the given figure. **The visual capability of existing MLLMs is still limited.**
3. **GPT-4V and other MLLMs can be easily misled by simple image manipulations** in HALLUSIONBENCH, including image flipping, order reversing, masking, optical character editing, object editing, and color editing.

4. **GPT-4V and other MLLMs are unable to capture the temporal relations of multiple images and fail to answer temporal reasoning question**s in HALLUSIONBENCH. The existing MLLMs lack true temporal reasoning ability.

Chapter 3: Causes of Hallucination in MLLMs

Hallucinations have multifaceted origins, spanning the entire spectrum of MLLMs' capability acquisition process. In this section, we delve into the root causes of hallucinations in MLLMs, primarily categorized into four aspects: Data, Model, Training..

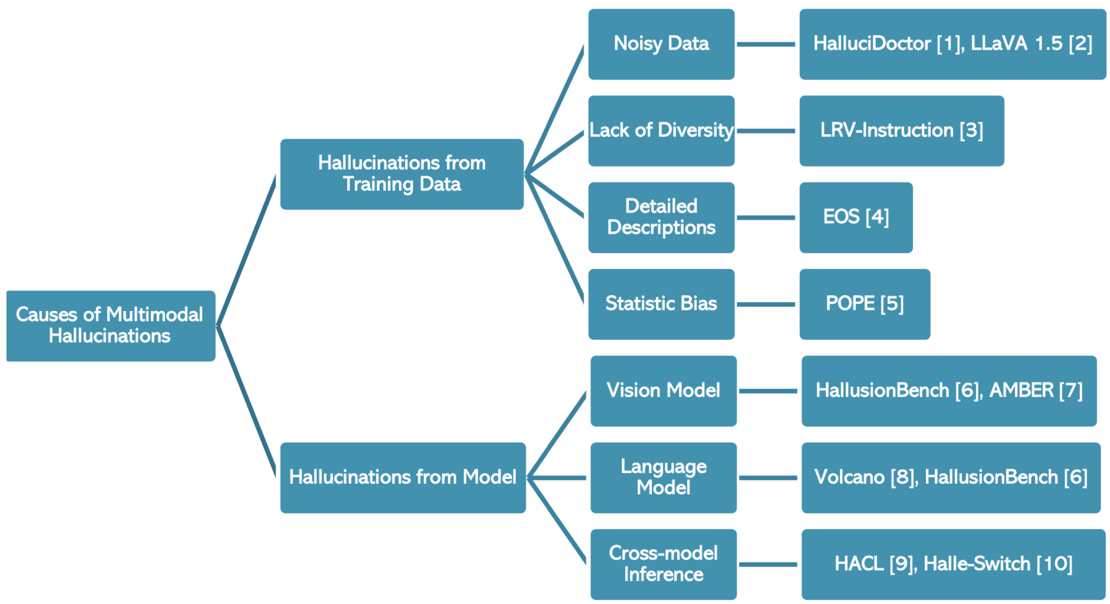


Fig 3.1. ***Causes of hallucination in MLLMs.***

3.1 Data

Given the increasing demand for large-scale training data, heuristic data collection methods are employed to efficiently gather vast volumes of data. While these methods provide extensive data, they offer no guarantee of quality, thereby increasing the risk of hallucinations. Data quality relevant to hallucinations can be further categorized into the following three facts.

Noisy Data. The training of Multimodal Large Language Models (MLLMs) generally consists of two stages: pre-training and instruction tuning. During pre-training, models are exposed to massive amounts of image-text pairs crawled from the web. However, such datasets frequently suffer from noisy supervision, including inaccurate captions, misaligned content, and even irrelevant or corrupted samples [1, 2]. These noisy signals hinder the model's ability to align visual and textual modalities, thereby increasing the risk of cross-modal hallucinations, especially when the model is asked to reason about fine-grained visual details.

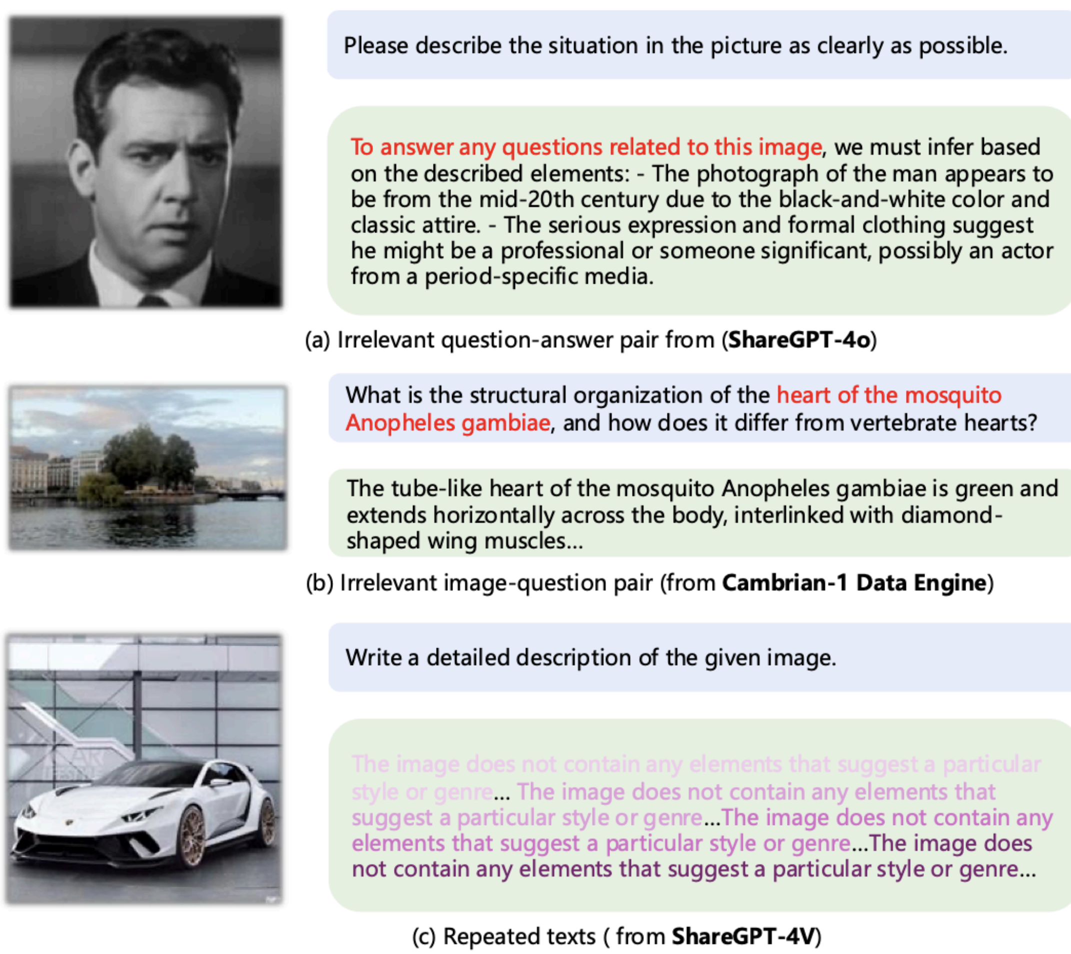


Fig 3.2. ***Examples of noisy data in the training set.***

In the subsequent instruction tuning stage, a common practice—adopted by models like LLaVA—is to leverage powerful language models such as GPT-4 to generate vision-language instructions. While GPT-4 excels at natural language generation, it lacks the ability to interpret images, as it was trained without direct visual input. As a result, instructions generated in this way may reflect hallucinated objects, incorrect spatial relationships, or fabricated facts, particularly when the model assumes the presence of unseen elements in an image. This introduces a second layer of noise into the training pipeline, further exacerbating the model's tendency to hallucinate.

Lack of Diversity. For the data used in the two training stages, instruction tuning data are more likely to have this issue since it is usually in a relatively small amount. One prominent property is that most instruction following data samples are composed of conversations regarding the image content. We regard this type of data as positive instruction [4], as it always faithfully reflects the image content. In contrast, negative instruction data [1] and reject answering responses are rare in the datasets. Given such training data, one potential drawback observed by recent studies is that current models tend to answer ”Yes” for any instructions presented to the model [1, 5], even when a proper answer should be ”No”, leading to hallucination.

Statistic Bias. Large language models often memorize training data [8], and their behavior is heavily influenced by object distributions within it. Common biases include frequent object appearances and co-occurrences [5]. For instance, models may predict a "*person*" even if none is present, or assume a microwave is in a kitchen just because it often appears with a refrigerator. While scaling data [40] can help, it doesn't fully remove these biases due to real-world long-tail distributions.

3.2 Model

Currently, the architecture of popular MLLMs is composed of several components, usually including pre-trained vision model, pre-trained LLM, and alignment module as we discussed above. Since these models are connected together, instead of end-to-end training from scratch, the error of each module can be accumulated. Inferior and problematic output from each module may lead to hallucinations [10].

Weak Vision Model. As mentioned in related works [20, 25], a primary potential reason for hallucination is a weak vision model, which can lead to misclassification or misinterpretation of visual concepts. Even the most powerful vision model may still experience information loss during the encoding process. Weak vision model implies weak perception, which fundamentally undermines the multimodal understanding.

Language Model Prior. The modern architecture of MLLMs is imbalanced. Usually, the language model is much larger and stronger than the vision model, leading to a tendency to prioritize language-based information. A typical phenomenon is that the knowledge entailed in the language model, also termed as parametric knowledge, can override the visual content [2]. For example, given an image showing a red banana, which is counter-intuitive in the real world, an MLLM may still respond with ”yellow banana", as "banana is yellow" is a deep-rooted knowledge in the LLM. Such language/knowledge prior makes the model overlook the visual content and response with hallucination.

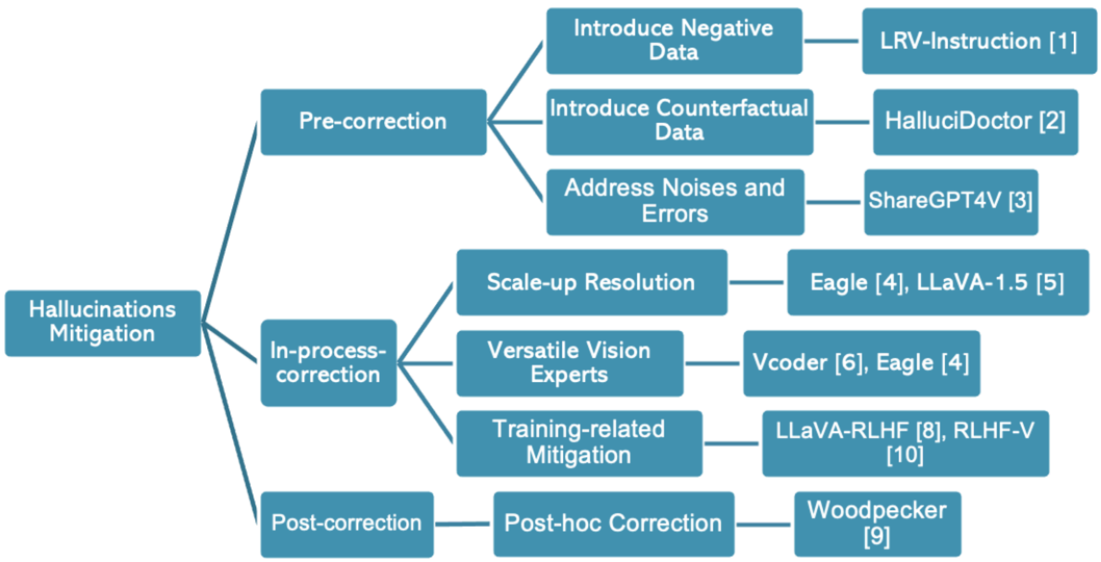


Fig 3.3. ***Hallucination Mitigations in MLLMs***

3.3 Training

The training objective of MLLMs [10, 12, 13] is basically the same as LLMs, i.e, auto-regressive next token prediction loss. This loss is straightforward yet effective and easy to scale up, showing promising performance in language modeling. However, some studies in the field of MLLMs have suggested that the next-token prediction loss might not be suitable for learning visual content due to its complex spatial structure. Additionally, the loss optimizes at the token level, while lacking supervision at the sequence level. Another perspective is that, unlike training LLMs, the RLHF stage is absent in the training procedure of MLLMs [17], becoming a potential cause of hallucination.

In next sections, we present methods aimed at mitigating hallucinations in MLLMs. To address object hallucinations in MLLMs, existing methods can be broadly categorized into three stages: pre-correction, in-process correction, and post-correction.

1. **Pre-correction** methods focus on improving data quality before model training. This includes: introducing negative data [1], introducing counterfactual data, which augments the dataset with minimally altered examples to teach sensitivity to factual inconsistencies and addressing noises and errors in the training set.
2. **In-process correction** strategies intervene during training or inference: scaling up image resolution [28], incorporating versatile vision experts, training-related mitigation,
3. **Post-correction** techniques apply corrections after the output is generated. Woodpecker [28] exemplifies this approach by verifying and editing responses to reduce hallucinated content post-hoc. These multi-stage strategies collectively contribute to a robust framework for reducing hallucinations in MLLMs, targeting the problem across the model development lifecycle.

Chapter 4: Hallucination Mitigations in MLLMs - Data Perspective

As discussed in the section on hallucination causes, data is one of the primary factors inducing hallucination in MLLMs. In this section, we present a comprehensive review of mitigation approaches in the data perspective.

4.1 Introduction of LRV-Instruction

We introduce Large-scale Robust Visual (LRV)-Instruction, including 400k visual instructions generated by GPT4, covering 16 vision-and-language tasks with open-ended instructions and answers. Unlike existing studies that primarily focus on positive instruction samples, we design LRV-Instruction to include both positive and negative instructions for more robust visual instruction tuning. Our negative instructions are designed at three semantic levels: (i) Nonexistent Object Manipulation, (ii) Existent Object Manipulation and (iii) Knowledge Manipulation.

We conduct comprehensive experiments to investigate the hallucination of MLLMs. Our results demonstrate existing MLLMs exhibit significant hallucinations when presented with our negative instructions, particularly Existent Object and Knowledge Manipulation instructions. Moreover, we successfully mitigate hallucination by finetuning MiniGPT4 and mPLUG-Owl on LRV-Instruction while improving performance on several public datasets compared to state-of-the-art methods.

4.2 Construction of LRV-Instruction

Annotating large-scale visual instruction data can be challenging and time-consuming [57, 61]. It involves expertly written detailed instructions and specific labels for different tasks. Inspired by the success of GPT4 in text-annotation tasks, we leverage GPT4, instead of human workers, to build LRV-Instruction. LRV-Instruction is designed to cover a variety of VL tasks, with open-ended positive and negative instructions (*Fig. 4.1*) in different linguistic styles.

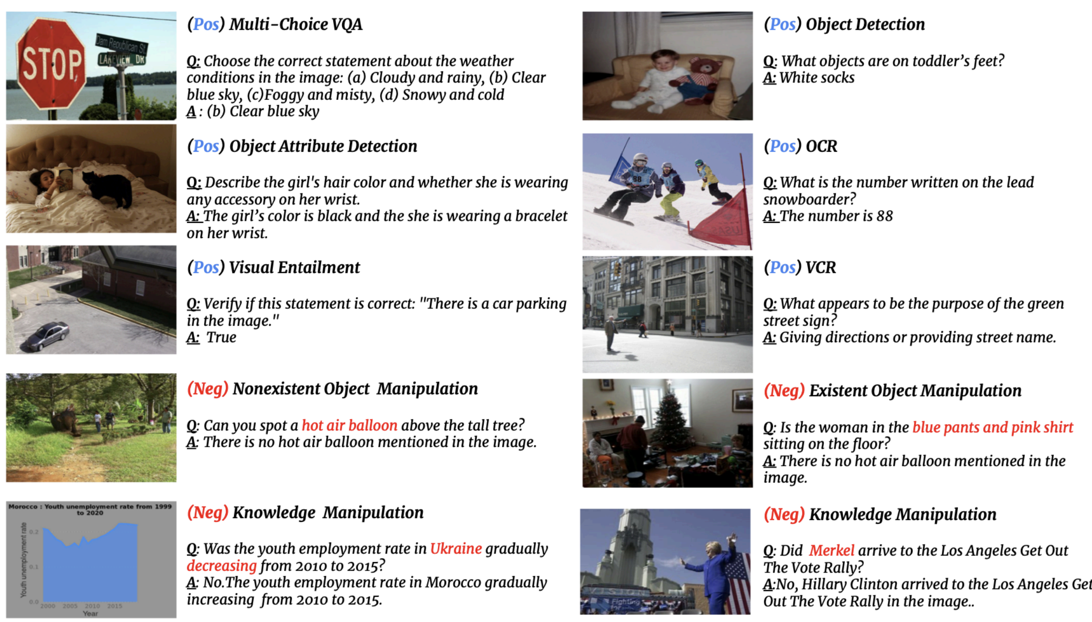


Fig 4.1. ***Examples of positive and negative instances in our LRV-Instruction dataset.*** *RED means inconsistent elements in the negative instructions.*

Prompt:

Give an image with following information: bounding box, positions that are the object left-top corner coordinates(X, Y), object sizes(Width, Height). Highly overlapping bounding boxes may refer to the same object.

bounding box:
elephant heard on rocks X: 73 Y: 80 Width: 418 Height: 418
woman wearing long dress X: 176 Y: 298 Width: 35 Height: 83
group of green chairs X: 153 Y: 326 Width: 95 Height: 126
an orange bucket on the ground X: 91 Y: 341 Width: 38 Height: 36
a group of white umbrellas X: 99 Y: 82 Width: 112 Height: 28
a man in an orange shirt X: 204 Y: 265 Width: 31 Height: 47
a woman wearing a yellow dress X: 169 Y: 298 Width: 47 Height: 76
. . .

Task: *image captioning, Image Sentiment Analysis, Image Quality Assessment, Object Interaction Analysis, Object Attribute Detection, Muli-choice VQA ...*

Come up with 20 diverse instructions for all the tasks above with different language styles and accurate answers. The instructions should contain interrogative sentence and declarative sentences. The answers should be less than 30 words. Each task should have less than 3 instructions.

GPT4 OUTPUT Example:

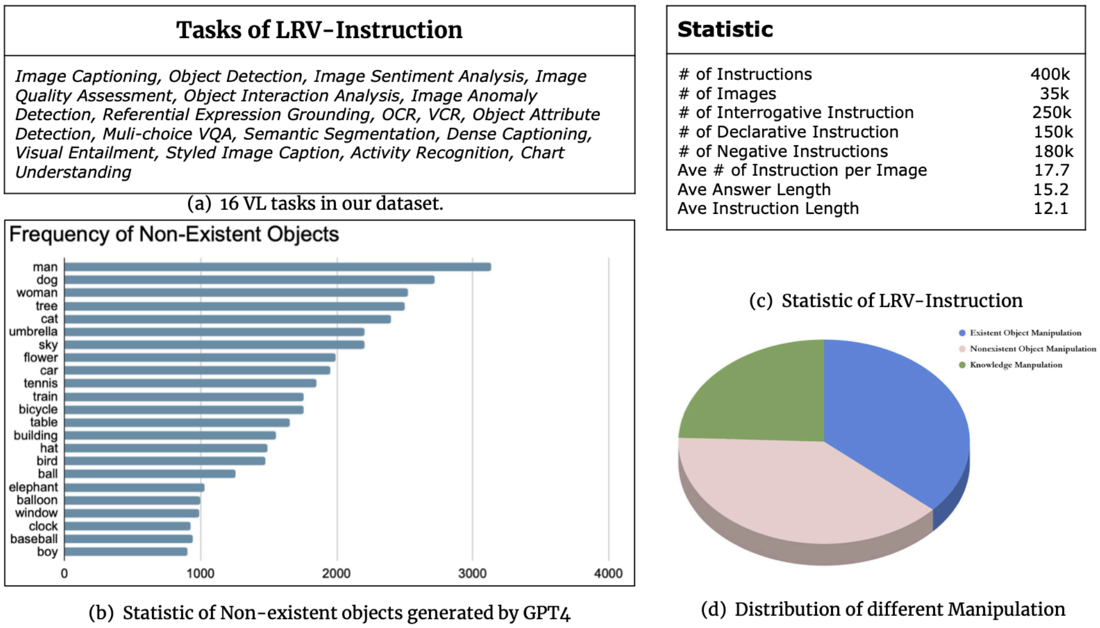
Instruction:
Craft a brief narrative about the baby elephant and adult elephant.
Answer:
A baby elephant is depicted behind an adult elephant, possibly seeking protection.

Fig 4.2. ***One example to illustrate the prompt we use to generate the visual instruction data by GPT4.*** *We use the bounding box coordinates and dense captions to represent image content.*

Positive Visual Instruction Generation. We use the in-context few-shot learning ability of GPT4 to generate instruction data for various VL tasks automatically. We filter the output tasks manually and select 16 tasks with text answers. We take advantage of the Visual Genome dataset, which has detailed visual information like image size, bounding boxes, and dense captions. Specifically, each image typically has 21 object regions and their corresponding captions. We leverage GPT4 to create the instruction-following data with the image size, bounding boxes, and dense captions as the "visual" input as if it can "see" the image. An example is shown in *Fig. 4.2*. For each image, we randomly select 10 tasks. To enrich the instructions, we ask GPT4 to generate instances in both declarative and interrogative formats. The limitation of recent work is that synthetic visual instructions are generally longer and may involve unexpected descriptive information inconsistent with the image. Therefore, we explicitly instruct GPT4 with "The answers should be less than 30 words" to reduce the chance of generating extra unrelated information in the training data.

Negative Visual Instruction Generation. Current MLLMs tend to answer “Yes” by following any instruction presented to the model rather than predicting a faithful answer. To teach MLLMs to answer questions in instructions faithfully, we introduce three categories of negative instructions based on Visual Genome dataset:

1. **Neg1**: "Nonexistent Object Manipulation" by introducing nonexistent objects, activities, attributes and interactions to the "visual" input as described above.
2. **Neg2**: "Existent Object Manipulation" by manipulating existent objects with inconsistent attributes.
3. **Neg3**: "Knowledge Manipulation" by manipulating knowledge in instructions. As for the detailed prompt of Neg1, we leverage the same format of the "visual" input as shown in *Fig. 4.2*. Additionally, we provide the following instructions to GPT4: *"Come up with 6 misleading instructions with nonexistent elements (nonexistent objects, nonexistent activities, nonexistent attributes, nonexistent interactions) in the images with different language styles. The instructions should contain interrogative and declarative sentences. Please also explain the reason."*



(a) 16 VL tasks in our dataset.

(b) Statistic of Non-existent objects generated by GPT4

(c) Statistic of LRV-Instruction

(d) Distribution of different Manipulation

Fig 4.3. ***Comprehensive Statistic of LRV-Instruction.*** *In (d), BLUE means existent object manipula- tion. PINK means nonexistent object manipulation. GREEN means knowledge manipulation.*

Quality Control. We first remove instances with answers longer than 30 words. We remove the instances mentioning unneeded content like "bounding box description", "given caption", and "existing descriptions". Additionally, GPT4 will output the task name for each instruction. However, we found that GPT4 sometimes assigns inaccurate task names for the instructions. As a result, we exclude the task name in our release data. Furthermore, we removed the instructions asking about facial expressions. This is because the Visual Genome dataset doesn’t include facial expression attributes in the ground truth-dense captions. To examine the quality of our dataset, we randomly sample 500 instances and ask ten expert annotators to determine whether the output answers from GPT4 are correct or not, with regard to the instruction and the image content. We found 91% of the instructions are appropriate for the image inputs. Furthermore, 85% of outputs are acceptable responses to the instructions. Even though some responses may contain errors, most generations conform to the correct structure, serving as applicable visual instruction tuning guidelines. We created a total of over 400k visual instructions after filtering.

Evaluation Set. After the processing above, we randomly select 1000 instances as our evaluation set. Furthermore, we manually check the quality of all instances and see whether the instruction describes a valid task. If it's not, we edit the instruction to make it clearer for LMMs. For example, we edit the instruction "*Observe the beautiful rainbow-colored sign that says 'Le Louvre'. You won't miss it!*" to "*Are you able to observe the beautiful rainbow-colored sign that says 'Le Louvre' in the image?*"

4.3 Method - Visual Instruction Tuning

We constructed two current MLLMs: MiniGPT4 [23] and mPLUG-Owl [24] as the backbones for visual instruction tuning. MiniGPT4 consists of the Vision transformer backbone as the image encoder, Vicuna as the text decoder and a pre-trained Q-Former to connect them. Vicuna is built upon LLaMA with stronger following ability. The Q-Former is designed to extract visual features from the frozen image encoder. Before feeding into the frozen Vicuna as the visual prompt, we use a learnable linear projection layer to narrow the gap between extracted visual features with Vicuna embeddings. mPLUG-Owl comprises a pre-trained visual encoder, a visual abstractor, and Vicuna as the text decoder. The visual encoder is responsible for extracting visual features from the input images, and the visual abstractor distills these features using a set of learnable tokens. The resulting visual features are concatenated with the word embeddings of the input sentence and fed into Vicuna to generate the response. We freeze the visual abstractor and visual encoder. Instead, we adopt the low-rank adaptation to train the text decoder.

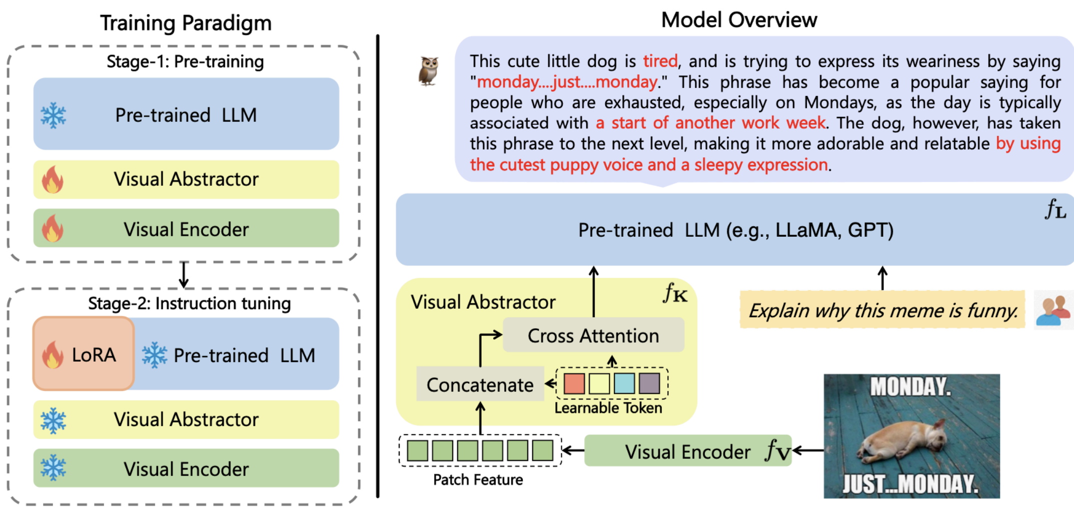


Fig 4.4. ***Visual instruction tuning framework and training paradigm.***

4.4 GPT4-Assisted Evaluation

CHAIR [41] was introduced to evaluate object hallucination in image captioning but often relies on complex human-crafted rules. An alternative line of work frames hallucination evaluation as a binary classification task, prompting the model to output "Yes" or "No," but this approach struggles with open-ended responses and depends heavily on human-annotated groundtruth. To address these limitations, we introduce GPT4-Assisted Visual Instruction Evaluation (GAVIE), a more flexible and robust method for assessing object-level hallucination. GPT4 is given dense captions with bounding boxes as image content and compares the human instruction with the model's response. Acting as a smart teacher, GPT4 scores the response from 0 to 10 based on two criteria: (1) **Accuracy**: whether the answer aligns with the image content, and (2) **Relevancy**: whether the answer directly follows the instruction. For experiments, we use GPT4-32k-0314. GAVIE identifies inconsistencies such as "dog, ball" not being in the image and highlights when the model fails to follow the instruction. Unlike prior methods, GAVIE requires no human-annotated answers and supports diverse instruction formats.

4.5 Experiments

4.5.1 Implementation Setup

Baselines. We evaluate the zero-shot performance of 5 recently released LMMs: (1) MiniGPT4 [23]; (2) MiniGPTv2; (3) InstructBLIP [12]; (4) Multimodal-GPT (MMGPT); (5) mPLUG-Owl [24]; (6) LLaVA [6]; (7) LLaVA 1.5 [4]. All models above have been tuned on their collected visual instruction data.

Training Details. As for MiniGPT4 [23], we initialize from its checkpoint of the first pretraining stage. Then we instruct-tune the model on LRV-Instruction with the linear projection layer as the only learnable module. As for mPLUG-Owl, we train the text encoder by LoRA training. Additionally, we only replace the LLaVA dataset in their finetuning data with LRV-Instruction to make a fair comparison with the original Mplug-Owl. We utilize MiniGPT4-7B and mPLUG-Owl-7B since we don't have the computing resources to finetune the 13B models. We trained our models on NVIDIA Quadro RTX 8000.

Evaluation Benchmarks. Apart from our proposed evaluation set, we evaluate LMMs on three public benchmarks. MME [18] is a human-annotated benchmark, measuring perception and cognition abilities on 14 subtasks. POPE [5] and AMBER [33] are recently released datasets to evaluate object hallucination. GQA [65] dataset is a public visual question-answer dataset with open-ended questions.

4.5.2 Main Results

Backbone	Perception	Cognition
Original MiniGPT4	616.41	232.71
Finetuned MiniGPT4	**895.96**	**296.43**
Original mPLUG-Owl	967.34	276.07
Finetuned mPLUG-Owl	**1298.78**	**328.21**

Backbone	Acc(Pos)	Acc(Neg)
Original MiniGPT4	0.53	0.54
Finetuned MiniGPT4	**0.58**	**0.68**
Original mPLUG-Owl	0.62	0.55
Finetuned mPLUG-Owl	**0.69**	**0.78**

Fig 4.5. ***Zero-shot multimodal evaluation on MME.*** *The left chart shows perception and cognition scores. The right chart shows the accuracy on the positive set and the negative set.*

We compare our model against the baseline models on POPE in *Fig. 4.6* and AMBER in *Fig. 4.7*. The results show that current LMMs may not work well with open-ended negative instructions. In contrast, the highest scores of our model demonstrate that *LRV-Instruction* exhibits robustness to visual hallucination, matching or surpassing the performance of 13B counterparts. From *Fig. 4.5*, we found both finetuned LMMs on *LRV-Instruction* outperform original ones in the zero-shot evaluations. Additionally, Finetuned-Mplug-Owl exceeds Finetuned-MiniGPT4 because Mplug-Owl can do the LoRA training to improve the language ability. We also calculate the accuracy on positive and negative samples of MME. The improvement in the positive samples is because LRV-Instruction has more diverse tasks than mPLUG- Owl datasets and MiniGPT4 datasets. The improvement in the negative samples demonstrates the value of LRV-Instruction dataset to equip the model with the ability to say 'no' and provide correct answers. We further explore the LMMs' performance in the common scenario of visual question-answering (VQA). Results in *Fig. 4.5* suggests our method (Finetuned mPLUG-Owl) achieves on-par performance with InstructBLIP in a generic VQA setting.

Model	Acc	F1
mPLUG-Owl-7B	0.52	0.68
LLaVA-13B	0.50	0.66
MiniGPT4-13B	0.73	0.71
InstructBLIP-13B	0.86	0.87
Ours-7B	**0.86**	**0.88**

(a) Random Set.

Model	Acc	F1
mPLUG-Owl-7B	0.57	0.66
LLaVA-13B	0.50	0.66
MiniGPT4-13B	0.67	0.67
InstructBLIP-13B	0.71	0.76
Ours-7B	**0.73**	**0.79**

(b) Popular Set.

Model	Acc	F1
mPLUG-Owl-7B	0.60	0.64
LLaVA-13B	0.50	0.66
MiniGPT4-13B	0.62	0.63
InstructBLIP-13B	0.63	0.72
Ours-7B	**0.65**	**0.73**

(c) Adversarial Set.

Fig 4.6. ***Zero-shot object hallucination evaluation on POPE.*** *Objects not existing in the image are sampled with three different strategies. Random: random sampling, Popular: top-k most frequent objects in MS-COCO, Adversarial: objects are first ranked based on co-occurring frequencies, then top-k frequent ones are sampled. Ours-7B means Finetuned mPLUG-Owl-7B.*

	mPLUG-Owl-7B	MiniGPT4-v2-7B	LLaVA1.5-7B	**Ours**
EXISTENCE	0.29	0.80	**0.83**	0.81
ATTRIBUTE	0.34	0.41	0.64	**0.70**
RELATION	0.26	0.58	0.65	**0.69**

Fig 4.7. ***Comparison results on AMBER.*** *All the LMMs are 7B versions to make a fair comparison*

4.5.3 Detailed Analysis

Does GPT4-Assisted Visual Instruction Evaluation align with Human Evaluation? We select three human experts specializing in the field of NLP to evaluate the predictions from LMMs with four options for the scores (1) Very Poor, (2) Poor, (3) Good, (4) Excellent. To evaluate the results quantitatively, we assign different scores for the options: Very Poor=1, Poor=2, Good=3, Excellent=4. More implementation details are shown in the appendix. From *Fig 4.8*, all experts agree that the output from our model is the best, followed by InstructBLIP in second place, and MMGPT performs the worst. The observation aligns with the GAVIE evaluation results.

GAVIE	**Ours**	MiniGPT4	LLaVA	InstructBLIP	MMGPT	mPLUG-Owl
ACCURACY (0-10)	**6.58**	4.14	4.36	5.93	0.91	4.84
RELEVANCY (0-10)	**8.46**	5.81	6.11	7.34	1.79	6.35
Human Expert1 (1-4)	**3.48**	2.61	2.87	3.00	1.90	2.90
Human Expert2 (1-4)	**3.58**	2.23	2.07	2.48	1.05	2.27
Human Expert3 (1-4)	**3.33**	2.58	2.89	2.94	1.38	2.91

Fig 4.8. ***Comparison results on our evaluation set evaluated by GAVIE.*** *Ours means Finetuned mPLUG-Owl-7B. All the LMMs are 7B versions to make a fair comparison.*

How do LMMs perform at the different semantic levels of hallucination? As shown in *Fig 4.9*, all baselines perform better on Neg1 (Nonexistent Object Manipulation) than Neg2 (Existent Object Manipulation) and Neg3 (Knowledge Manipulation). From the visual perspective, existent object manipulations with wrong attributes in Neg2 are more challenging than adding nonexistent objects from images to instructions in Neg1. For example, in *Fig 4.9*, it may be straightforward to find that the "hot air balloon" does not appear in the image. However, "woman" does exist in the second example of *Fig 4.9* while she is not in the blue pants and pink shirts, which requires a fine-grained understanding of the visual content. Therefore, a more powerful vision encoder is needed for future LMMs. Knowledge manipulation is challenging because current LMMs are finetuned on general images without specific knowledge. In contrast, our model greatly improves at all semantic levels, which benefits from our diverse instruction tuning data.

Categories	Metric	**Ours**	MiniGPT4	LLaVA	InstructBLIP	MMGPT	mPLUG-Owl
Neg1	ACCURACY(GPT4)	**8.90**	3.72	2.09	5.50	1.13	4.20
Neg2	ACCURACY(GPT4)	**6.50**	2.57	1.42	2.18	0.96	2.46
Neg3	ACCURACY(GPT4)	**6.25**	2.30	1.56	2.38	0.94	2.57
Neg1	RELEVANCY(GPT4)	**8.96**	5.94	4.83	7.22	2.24	5.35
Neg2	RELEVANCY(GPT4)	**8.46**	2.53	1.82	2.73	1.19	3.16
Neg3	RELEVANCY(GPT4)	**8.21**	2.40	1.78	2.39	0.98	2.87

Fig 4.9. ***Completed evaluation results on Neg1: Nonexistent Object Manipulation, Neg2: Existent Object Manipulation and Neg3: Knowledge Manipulation by GAVIE.***

How do LMMs perform at the different composition ratios in training data? In *Fig 4.10* (right), we investigate how LRV-Instruction addresses hallucination issues with different ratios of positive and negative samples in the training set. Inspired by [7], we instruct the model to produce “Yes” or “No” and use classification accuracy on our evaluation set. Accpos is the accuracy on the positive instruction set, while Accneg is the accuracy on the negative instruction set. From *Fig 4.10* (right), we found that Accneg increases with more negative samples, which verifies our hypothesis that the hallucination problem of current LMMs is due to the lack of negative instructions. Besides, with a balanced ratio (pos:neg=1:1), the model performs the best in both positive and negative sets.

Metric	Accuracy-STD	Accuracy-Mean
Ours	2.42	6.60
MiniGPT4	2.46	3.76
InstructBLIP	2.42	5.29
mPLUG-Owl	1.96	0.87
LLaVA	2.37	3.80
MMGPT	0.65	4.84

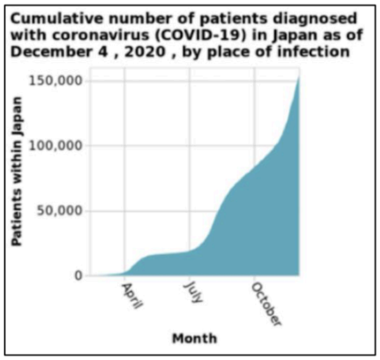
Ratio	Acc_{pos}	Acc_{neg}
All Pos	0.97	0.05
Pos:Neg=2:1	0.95	0.50
Pos:Neg=1:1	0.92	0.85
Pos:Neg=1:2	0.87	0.86
All Neg	0.10	0.98

Fig 4.10. (left): Evaluation of the stability of GAVIE. STD means standard deviation. (right): Results of different composition ratios in instruction tuning.

Use Pseudo Dense Captions instead of GT from Visual Genome to Generate Instructions. To demonstrate the scalability of our dataset, we use pseudo-dense captions generated by GRiT [42] to replace the GT captions in the Visual Genome dataset. We remove the images, whose detected objects by GRiT are less than 15 to ensure GPT4 has enough visual information when generating visual instructions. From Tab. 5, we found finetuning on pseudo captions can also improve the performance compared to the original mPLUG-Owl. This demonstrates that our visual instruction generation method can be further scaled up without groundtruth dense captions.

Model	InstructBLIP-13B	LLaVA-13B	MiniGPT4-13B	mPLUG-Owl-7B	**Ours-7B**	Ours-7B-Psu
Accuracy	0.62	0.47	0.42	0.41	**0.64**	0.60

*Fig 4.11. **Zero-shot evaluation on GQA**. Ours-7B means Finetuned mPLUG-Owl-7B. Ours-7B-Psu means we finetune mPLUG-Owl on pseudo instruction data by [42].*



Does the covid case increase in China?

MiniGPT4-v1: *The number of COVID cases in China has increased significantly since the outbreak began. According to the World Health Organization (WHO)...*

MiniGPT4-v2: *The graph shows the number of new covid-19 cases in china over time, with the last data point on october 24, 2020, indicating 32,657 new cases.*

LLaVA1.0: *Yes, the graph shows an increase in the number of COVID-19 cases in China. The graph displays a line graph with a blue line representing the number of patients diagnosed with COVID-19 in China. The graph indicates that the number of cases has increased over time.*

LLaVA1.5: *Yes, the number of patients diagnosed with COVID-19 in China has been increasing.*

Ours: *The image is about the COVID cases in Japan, not China.*

Fig 4.12. ***Visualization Example.***

4.5.4 Main Takeaways

Our results demonstrate existing MLLMs exhibit significant hallucinations when presented with our negative instructions, particularly Existent Object and Knowledge Manipulation instructions. Moreover, we successfully mitigate hallucination by finetuning MiniGPT4 and mPLUG-Owl on LRV-Instruction while improving performance on several public datasets compared to state-of-the-art methods. Additionally, we observed that a balanced ratio of positive and negative instances in the training data leads to a more robust model.

Chapter 5: Hallucination Mitigations in MLLMs - Model Perspective

In this section, we present a comprehensive review of mitigation approaches in the model perspective.

5.1 Introduction of Eagle

Recent work [10, 13, 18, 21] indicates that enhanced visual perception significantly reduces hallucinations and improves performance on resolution-sensitive tasks, such as optical character recognition and document analysis. A number of recent MLLMs achieve this goal using a mixture of vision encoders [19, 25, 93, 94]. Despite their success, there is a lack of systematic comparisons and detailed ablation studies addressing critical aspects, such as expert selection and the integration of multiple vision experts. This section provides an extensive exploration of the design space for MLLMs using a mixture of vision encoders and resolutions. Our findings reveal several underlying principles common to various existing strategies, leading to a streamlined yet effective design approach.

5.2 Design space exploration of Eagle

5.2.1 Base setup

We adopt LLaVA's model architecture as the basis, which consists of a large language model, a vision encoder, and a projection layer. The projection layer projects the visual embedding from the vision encoder into the text embedding space. We mainly focus on the vision encoder design.

Base training data. We use the same pre-training data as LLaVA-1.5 [4, 51, 52, 53, 54, 55, 56], which consists of 595k image text pairs. For the supervised fine-tuning stage, we collect data from a series of tasks and convert them into multimodal conversations, including: LLaVA-1.5 [4], Laion-GPT4V, ShareGPT-4V, DocVQA [36], synDog-EN, ChartQA [3], DVQA, and AI2D, resulting in 934k samples.

Total Data Size	Data Source
1,809k	*LLaVA*-1.5 (665k) (Liu et al., 2023c), DocVQA (39k) (Mathew et al., 2021), synDog-EN (50k) (Kim et al., 2022), ChartQA (28k) (Masry et al., 2022), DVQA (25k) (Kafle et al., 2018), AI2D (15k) (Kembhavi et al., 2016a), ShareGPT-4V (100k) (Chen et al., 2023b), laion-GPT4V (11k) (lai, 2023), LVIS-Instruct4V (220k) (Wang et al., 2023a), LRV-Instruct (150k) (Liu et al., 2023b), Geo170k (120k) (Gao et al., 2023), LLaVAR (20k) (Zhang et al., 2023), Visual7W (70k) (Zhu et al., 2016), Open-Hermes 2.5 (300k) (Teknium, 2023)

Fig 5.1. ***Composition of the base supervised fine-tuning data (Eagle1.8M).***

Implementation details. We first pre-train the model with image-text pairs for one epoch with a batch size of 256, where the whole model is frozen and only the projector layer is updated. In the second stage, we further fine-tune the model on the supervised fine-tuning data for one epoch with a batch size of 128. For this exploration, we employ Vicuna-7B as the underlying language model. The learning rates are set to 1e-3 for the first stage and 2e-5 for the second stage, respectively.

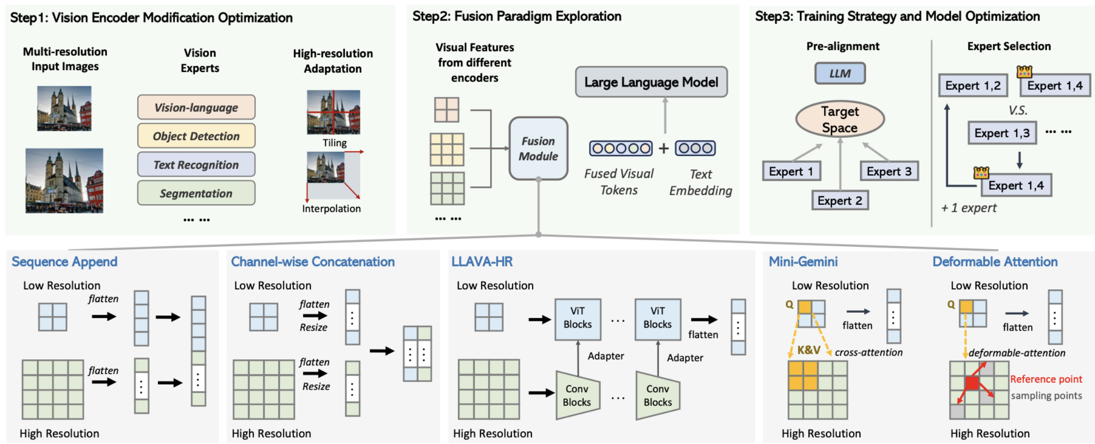


Fig 5.2. ***Overview of the Eagle exploration.***

5.2.2 Scale-up Resolution

We start our exploration with the CLIP model since it has become the main choice for many MLLMs. While CLIP models are known to benefit multimodal tasks, their drawbacks have also been well- studied. For instance, many existing MLLMs tend to use the pre-trained CLIP resolutions (such as 224 × 224 or 336 × 336) as their input resolutions. In these cases, the encoders often fail to capture fine-grained details that are important for resolution-sensitive tasks like OCR and document understanding.

To handle increased input resolution, a common practice is to use tiling where input images are divided into tiles and encoded separately. Another simpler method is to directly scale up the input resolution and interpolate the position embeddings of the vision transformer model if needed. We compare these two approaches with frozen/unfrozen vision encoders under different resolutions. Our findings can be summarized as follows:

1. Updating the CLIP encoder during SFT significantly improves performance at higher resolutions but slightly reduces it when using the pre-training resolution.
2. Interpolating the CLIP encoder to fit the input size of 448 × 448 offers a strong balance between efficiency and performance, trailing the 672 × 672 version with less than half the tokens.
3. Despite its smaller size (0.3B vs. 5.9B) and less pre-training data, the CLIP encoder gets close with interpolation approaches InternVL's performance under the same setting.

Method	Unfreeze	Res.	#Tok(V)	#Params	FLOPs	Img/Sec	Avg.
Original	✗	336	576	0.3B	119G	197.2	616.5
Original	✓	336	576	0.3B	119G	197.2	562.6
Interpolate	✗	448	1024	0.3B	214G	119.5	589.7
Interpolate	✓	448	1024	0.3B	214G	119.5	670.5
Interpolate	✓	672	2304	0.3B	480G	56.3	**674.2**
Tiled-input	✓	672	2304	0.3B	476G	51.6	673.9
InternVL	✗	448	1024	5.9B	5669G	13.52	661.9
InternVL	✓	448	1024	5.9B	5669G	13.52	671.5

Fig 5.3. ***Comparison of different high-resolution adaption methods.*** *Direct interpolation is more efficient and effective. "Res" denotes input image resolution. "#Token/s" denotes the inference speed of the whole framework. "#Token (V)" denotes the number of visual feature tokens fed into LLM. "Avg" denotes average.*

Category	Vision Encoder	Res.	Post-process	Unfreeze	Avg.	Model Link
VL Alignment	*ConvNeXt*	1024	None	✗ ✓	654.6 **682.1**	ConvNeXt-XXL
Segmentation	*SAM*	1024	Pixel Unshuffle	✗ ✓	486.2 **510.5**	SAM-ViT-Large
Object Detection	*EVA-02*	1024	Resize	✗ ✓	543.7 **639.1**	EVA-02-L-Det
Text Recognition	*Pix2Struct*	1024	Resize	✗ ✓	598.6 **606.2**	Pix2Struct-02-Large
Self-Supervised	*DINOv2*	448	None	✗ ✓	520.7 **537.3**	ViT-L/14-Reg

Fig 5.4. ***Comparison between different vision experts as the MLLM encoders.***

5.2.3 Vision Experts

To better establish the foundation for multi-vision expert fusion, we extend the toolbox with vision experts pre-trained on different tasks and resolutions, and verify our findings on high-resolution adaptation with these experts. This also helps us identify the distinct advantages of different experts. We collect a set of vision experts, including: (1) Vision-Language Alignment: CLIP and ConvNeXt [34]. (2) Object-Centric: EVA-02 [17] pre-trained on detection datasets. (3) OCR: Pix2Struct [20]. (4) Segmentation: SAM [37]. (5) Self-supervised: DINOv2 [35]. The detailed input resolution and checkpoint of each vision encoder can be found in *Fig 5.4*. We resize the output 2D feature maps of each vision encoder using bilinear interpolation and pixel shuffle to ensure that the visual token number equals 1024.

The results in *Fig 5.4* reveal that MLLMs with these task-specific vision encoders achieve optimal performance in their pre-training domains. EVA-02 excels in the object hallucination evaluation benchmark POPE [5] and general visual question answering benchmark GQA [65]. CLIP and ConvNeXt perform well across all benchmarks, benefiting from their training on large-scale image-text pairs using contrastive loss.

5.2.4 Fusion Strategy

Existing MLLM frameworks have proposed various fusion strategies to ensemble the encoders in *Fig 5.5*, with the hope that their domain-specific strengths can be leveraged. In all cases, improvements in MLLM performance have been reported with the fusion of vision encoders. However, the roles of the fusion strategies as part of their MLLM architecture innovations, have not been decoupled and clearly studied under an "apples to apples" comparison. It is thus not entirely clear how much improvement is from the fusion strategies themselves versus the improved representations from various encoders.

Vision Encoders	Fusion	#Token(V)	#Tokens/s	#Params	Avg.
CLIP + ConvNeXt	*Seq. Append*	2048	46.1	1200M	**690.5**
	Channel Concat.	1024	**47.3**	**1184M**	681.5
	LLaVA-HR	1024	47.0	1219M	678.7
	Mini-Gemini	1024	45.3	1201M	672.5
	Deformable Attn.	1024	**47.3**	1201M	674.3
CLIP + ConvNeXt + SAM	*Seq. Append*	3072	40.3	1529M	686.2
	Channel Concat.	1024	**46.3**	**1495M**	**690.4**

Fig 5.5. ***Comparison of different fusion methods for different vision experts.***

We notice that existing popular fusion strategies, despite their variations in designs, can be broadly represented by the following several categories, including Sequence Append, Channel Concatenation, LLaVA-HR, Mini-Gemini and Deformable Attention.

Our study in *Fig 5.5* shows that Channel Concatenation achieves the best average performance while maintaining better throughput compared to sequence append. The "injection-based" methods are in general less competitive, performing worse than using ConvNeXt [34] alone as the vision encoder. A plausible explanation is that the CLIP features continue to play a dominant role in the visual tokens. Although sequence append shows comparable performance to channel concatenation, it faces the challenge to handle more vision encoders due to the increasing sequence length.

5.2.5 Vison-language Pre-Alignment

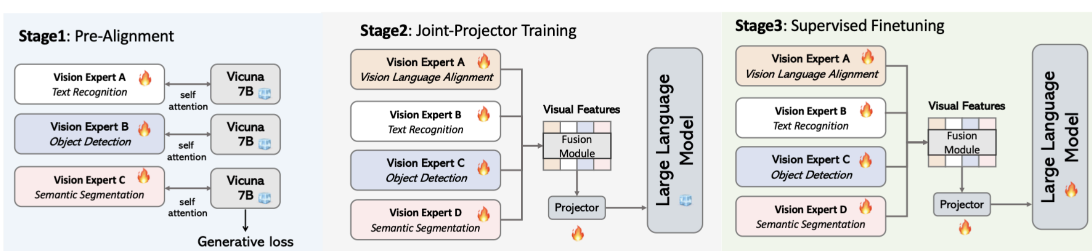


Fig 5.6. ***The proposed training strategy of Eagle.*** *It consists of three progressive stages, including vision-language pre-alignment training, joint-project training and supervised fine-tuning. These stages effectively leverage public data from diverse sources, ranging from noisy image-text pairs on the web to high-quality caption, VQA, and multimodal dialogue datasets.*

Encoders pre-trained exclusively on vision tasks (e.g., detection, OCR, and segmentation) are less competitive compared to those pre-trained on vision language alignment. This is possibly due to representational inconsistencies when integrated with large language models. Additionally, when combining different encoders, there is a gap between these encoders, creating difficulties in the training process. To address this feature inconsistency, we propose a Pre-Alignment training stage that first aligns each individual vision encoder with the same large language model, fostering better synergy between visual and linguistic capabilities.

Fig. 5.6 depicts our pre-alignment strategy. Instead of training a projector to simultaneously align multiple vision experts as in LLaVA's original pre-training strategy, we first align the representation of each individual expert with a smaller language model (Vicuna-7B in practice) using next-token-prediction supervision. As shown in *Fig. 5.6*, with pre-alignment, the whole training process consists of three steps: 1) training each pre-trained vision expert with their own projector, while keeping the language model frozen; 2) combining all vision experts from the first step and training both the projector and vision experts; 3) training the whole model on SFT data.

To verify the proposed method, we compare the pre-alignment strategy with the normal two-stage training strategy in *Fig. 5.7* considering both freezing and unfreezing vision experts for comparison. As shown in *Fig. 5.7,* although unfreezing the vision experts during SFT helps improve performance by updating the vision experts to fit the language model, the Pre-Align strategy more effectively mitigates the inherent biases of each vision expert and stabilizes the training process, subsequently improving overall performance.

CLIP	Vision Expert (X)	Unfreeze	Pre-Align	MME	MMMU	OCR	SQA	POPE	TextVQA	Avg
		✗	✗	1495	35.1	292	71.8	85.6	59.7	593
CLIP-448	*SAM-1024*	✓	✗	1504	**36.9**	482	73.5	87.7	**63.2**	641
		✓	✓	**1554**	36.7	**505**	**73.8**	**88.2**	62.9	**649**
		✗	✗	1499	35.1	416	72.3	86.7	67.1	629
CLIP-448	*ConvNext-1024*	✓	✗	1522	**36.6**	550	**72.7**	87.9	**71.9**	**667**
		✓	✓	**1538**	36.3	**556**	71.9	**87.9**	71.6	**667**
		✗	✗	1500	35.8	467	71.9	86.1	60.9	627
CLIP-448	*Pix2Struct-1024*	✓	✗	1529	**36.2**	535	73.1	**87.1**	**64.5**	651
		✓	✓	**1564**	35.3	**541**	**73.3**	86.6	63.9	**652**
		✗	✗	1484	35.8	305	72.0	86.0	59.9	597
CLIP-448	*EVA-02-L-1024*	✓	✗	1537	37.0	474	72.6	87.5	62.6	639
		✓	✓	**1586**	**37.3**	**495**	**73.1**	**88.9**	**64.2**	**653**

Fig 5.7. ***The effectiveness of Pre-alignment.***

5.2.6 Extension to Multi-Experts

With the optimized strategies and training recipes of incorporating individual vision experts, we consider the incorporation of even more vision experts to push the limit. To conduct the searching in a systematic and principled manner, we adopt a step-by-step greedy strategy to incorporate additional vision experts.

We mark CLIP, ConvNeXt [34], SAM [37], DINOv2, Pix2Struct, and EVA-02-L [17] as A, B, C, D, E, and F, respectively. A round-robin scheme, as shown in *Fig 5.6*, is adopted. We first use the two top performing vision encoders, CLIP and ConvNeXt, as the basis and gradually add one more vision encoder each time. In each round, the best-performing vision encoder combination is retained for the next round.

#Encoder	Encoder Combination	Config	#Params (M)	FLOPs (G)	Img/Sec	Avg.
2	*CL + CN*	X2	1155.2	3347.2	18.1	681.5
3	*CL + CN + DI*		1460.6	3659.9	15.1	685.4
	CL + CN + SA		1463.9	4657.8	8.8	690.4
	CL + CN + PS		1669.6	4373.2	6.9	685.1
	CL + CN + EV	X3	1459.6	4280.9	9.1	**690.7**
4	*CL + CN + EV + DI*		1765.1	4593.6	8.3	688.0
	CL + CN + EV + SA		1768.4	5591.5	5.9	689.4
	CL + CN + EV + PS	X4	1974.1	5306.9	5.0	**694.6**
5	*CL + CN + EV + PS + DI*		2279.5	5619.5	4.7	684.7
	CL + CN + EV + PS + SA	X5	2282.8	6617.4	3.8	**697.1**
6	*CL + CN + EV + PS + SA + DI*	X6	2588.2	6930.1	3.6	686.8

Fig 5.8. ***Results of vision expert selection process.*** *A, B, C, D, E and F denote CLIP, ConvNeXt, SAM, DINOv2, Pix2Struct and EVA-02-L models respectively.*

Fig 5.8 reveals several insights. Generally, introducing additional vision encoders enhances the performance. This indicates that the distinct advantages of different encoders can be preserved and utilized; for example, integrating the EVA-02 encoder improves metrics on the POPE benchmark. Although individual metrics may vary, the aggregated performance shows an upward trend, as evidenced by normalized average metrics, suggesting that the overall efficacy of the system is enhanced with more encoders. Also, Table 6 shows that the best combination of vision experts are CLIP, ConvNeXt, SAM, Pix2Struct [20], and EVA-02 [17]. We will use this recipe in our final model.

	Model	MME	MMB	SEED	MathVista	MMMU	POPE	SQA^I	GQA	VizWiz	VQAv2	OCR	TextVQA	ChartQA
Vicuna-7B & Qwen-7B	*LLaVA-1.5* (Liu et al., 2023c)	1510	64.3	58.6	-	-	85.9	66.8	62.0*	50.0	78.5*	297	58.2	-
	LLaVA-NeXt (Liu et al., 2024a)	1519	67.4	70.2	34.6	35.8	86.5	70.1	64.2*	57.6	80.0*	490	64.9	-
	InternVL (Chen et al., 2023f)	1525	-	65.4	-	-	86.4	-	62.9*	52.5	79.3*	-	57.0	-
	LLaVA-HR (Luo et al., 2024)	1554	-	64.2	-	-	87.6	65.1	64.2*	48.7	81.9*	-	67.1	-
	Monkey (Li et al., 2024c)	-	-	-	-	-	-	-	60.7*	**61.2***	80.3*	514	67.6	65.1
	Mini-Gemini (Li et al., 2024b)	1523	65.8	-	32.2	36.8	-	71.1	64.5*	-	-	477	65.2	-
	Eagle-X5	1528	68.4	**73.9**	37.0	36.3	**88.8**	70.0	**64.9***	54.4	83.4*	529	71.2	67.7
	Eagle-X5 (+Pre-Align)	**1582**	**69.7**	73.7	**38.2**	**38.0**	88.7	**71.9**	64.6*	58.7	**83.6***	**566**	**71.9**	**69.3**
Vicuna-13B	*LLaVA-1.5* (Liu et al., 2023c)	1531	67.7	61.6	-	36.4	85.9	71.6	63.3*	53.6	80.0*	331	61.3	-
	LLaVA-NeXt (Liu et al., 2024a)	1575	70.0	71.9	35.3	36.2	86.2	**73.5**	65.4*	60.5	82.8*	514	67.1	62.2
	InternVL (Chen et al., 2023f)	1546	-	-	-	-	87.1	-	63.9*	54.6	80.2*	517	58.7	-
	LLaVA-UHD (Xu et al., 2024)	1535	68.0	-	-	-	89.1	72.0	65.2*	56.1	81.7*	-	67.7	-
	LLaVA-HR (Luo et al., 2024)	1540	-	64.5	-	-	87.8	68.1	64.8*	57.9	82.6*	-	68.1	-
	Mini-Gemini (Li et al., 2024b)	1565	68.6	70.6	37.0	37.3	-	71.9	65.8*	-	-	466	65.9	56.6
	Eagle-X5	**1609**	69.2	74.1	38.8	36.6	87.8	72.7	**66.2***	59.3	83.8*	574	**74.2**	69.9
	Eagle-X5 (+Pre-Align)	1605	**71.6**	**74.9**	**42.7**	**38.5**	**89.2**	**75.5**	64.6*	**60.9**	**84.5***	**598**	73.3	**72.1**

Fig 5.9. ***Main results with base training data.***

5.3 Experiments

5.3.1 Implementation Details

Vision encoders. We follow the best X4 and X5 configurations, where the interpolated CLIP-448 and pre-aligned vision experts are channel-concatenated, and trained following the exact best training recipes.

Model	Knowledge					General					OCR and Chart					Vision-Centric		
	Avg	SQA[I]	MMMU	MathVista	AI2D	Avg	MME	MMB	SEED	GQA	Avg	ChartQA	OCR	TextVQA	DocVQA	Avg	MMVP	RWQA
Llama3-8B																		
MGM-HD	55.7	75.1	37.3	37.0	73.5	72.7	**1606**	72.7	73.2	64.5	62.9	59.1	47.7	70.2	74.6	40.4	18.7	62.1
Cambrian-1	61.3	80.4	42.7	49.0	73.0	73.1	1547	**75.9**	74.7	64.6	71.3	73.3	62.4	71.7	77.8	57.6	51.3	64.2
Eagle-X5	**65.2**	**84.1**	**43.5**	**56.9**	**76.2**	**74.0**	1587	75.5	**76.5**	**64.9**	**77.0**	**80.7**	**62.6**	**76.7**	**87.1**	**59.6**	**52.0**	**67.2**
Vicuna-13B																		
MGM-HD	54.1	71.9	37.3	37.0	70.1	70.7	1597	68.6	70.6	63.7	60.8	56.6	46.6	70.2	69.8	38.4	19.3	57.5
Cambrian-1	60.2	79.3	40.0	48.0	73.6	73.7	1610	**75.7**	74.4	64.3	71.3	73.8	61.9	72.8	76.8	52.2	41.3	63.0
Eagle-X5	**63.8**	**82.6**	**42.2**	**54.6**	**73.8**	**74.6**	**1651**	**75.7**	**75.0**	**65.0**	**75.7**	**78.6**	**62.4**	**74.9**	**86.7**	**54.8**	**44.6**	**65.0**
Yi-34B																		
MGM-HD	62.4	77.7	48.0	43.4	80.5	76.2	1659	80.6	75.3	65.8	68.1	67.6	51.8	74.1	78.9	52.3	37.3	67.2
Cambrian-1	67.0	**85.6**	49.7	53.2	**79.7**	**76.8**	**1689**	**81.4**	75.3	**65.8**	71.9	75.6	60.0	76.7	75.5	**60.3**	**52.7**	67.8
Eagle-X5	**68.6**	85.5	**53.2**	**57.9**	79.1	76.3	1677	81.0	**75.6**	64.9	**75.4**	**77.2**	**62.4**	**78.8**	**83.0**	59.8	50.0	**69.5**

Fig 5.10. ***Results using the same training data as Cambrian-1 [43].***

5.3.2 Main Results

We compare Eagle model series across different VQA benchmarks in *Fig 5.9* and *Fig 5.10*. Eagle-X5 achieves state-of-the-art performance, underscoring the advantages with additional vision experts.

Evaluation on visual question answering tasks. We compare the Eagle model series across three Visual Question Answering (VQA) benchmarks: GQA, VQAv2, and VizWiz. As shown in *Fig 5.9*, Eagle-X5 achieves state-of-the-art performance on GQA and VQAv2, highlighting the benefits of incorporating additional vision experts.

Evaluation on OCR and chart understanding tasks. To evaluate the OCR, document, and chart understanding capabilities of Eagle, we benchmark our model on OCRBench, TextVQA, and ChartQA. As illustrated in *Fig 5.9*, our model significantly outperforms competitors on TextVQA, thanks to its high-resolution architecture and integration of multiple vision encoders. Notably, Eagle maintains a simple design, supporting up to 1024×1024 resolution without requiring complex tile-based image decomposition. *Fig 5.12* presents examples of OCR and document understanding. With high-resolution adaptation and more vision experts, our model can recognize small text within images and accurately follow user instructions. To highlight the impact of incorporating experts pre-trained on other vision tasks, we compare a model using only ConvNeXt and CLIP vision encoders with Eagle-X5 in *Fig 5.5*. The full model corrects prior errors, demonstrating that even with strong high-resolution vision-language encoders, performance can be further enhanced by integrating diverse vision experts.

Evaluation on multimodal benchmarks. We evaluate Eagle on seven multimodal benchmarks to showcase its capabilities from various perspectives, including MME, MMBench, SEED, MathVista, MMMU, ScienceQA, and POPE. Specifically, MME, MMBench, and SEED assess overall performance across real-world tasks involving reasoning, recognition, knowledge, and OCR. MMMU targets challenging questions from diverse domains requiring college-level knowledge. POPE evaluates visual hallucinations in multimodal models. All reported metrics follow the default settings of each benchmark. We use the perception score for MME, the en_dev split for MMBench, the image split for SEED, the test-mini split for MathVista, the val split for MMMU, the F1-score for POPE, and the image split for ScienceQA to ensure fair comparison with existing models.

Study on more advanced training recipes. *Fig 5.11* presents our step-by-step experiments to study the training recipes. We found that the best recipe is to first pre-align each vision expert on LLaVA-595K + Eagle1.8M. In the pretraining stage, we combine all vision experts from the first step and train both the projector and vision experts on LLaVA-595K + Eagle1.8M. Finally, we train the whole model on the Eagle1.8M.

Config Summary	Pre-align	Pre-train	Fine-tune	Avg.
1 epoch	✗	LLaVA-595K	Eagle1.8M	697.1
2 epoch	✗	LLaVA-595K	Eagle1.8M	698.3
1 epoch, unlock*	✗	LLaVA-595K	Eagle1.8M	698.0
1 epoch, unlock*	✗	LLaVA-595K+Eagle1.8M	Eagle1.8M	**699.5**
1 epoch	Eagle1.8M	LLaVA-595K	Eagle1.8M	706.6
1 epoch, unlock*	Eagle1.8M	LLaVA-595K	Eagle1.8M	707.1
1 epoch, unlock*	LLaVA-595K+Eagle1.8M	LLaVA-595K	Eagle1.8M	707.8
1 epoch, unlock*	LLaVA-595K+Eagle1.8M	LLaVA-595K+Eagle1.8M	Eagle1.8M	**708.9**

Fig 5.11. ***Comparison between different training strategies.*** *"1 epoch" means we train Eagle for 1 epoch in the supervised fine-tuning stage. 'unlock*' means we unlock vision encoders in the pre-training stage.*

Comparison with Cambrian-1. Using the same pre-training and supervised fine-tuning datasets from Cambrian-1 [43]. Eagle demonstrates superior performance across all the evaluated benchmarks without bells and whistles. As shown in *Fig 6.10*, Eagle outperforms the Cambrian-1 counterparts considerably for the OCR and Chart category. Consistent improvements are also observed for the General, Knowledge, and Vision-Centric categories, showing the robustness and generalization ability of the improved perception design in Eagle.

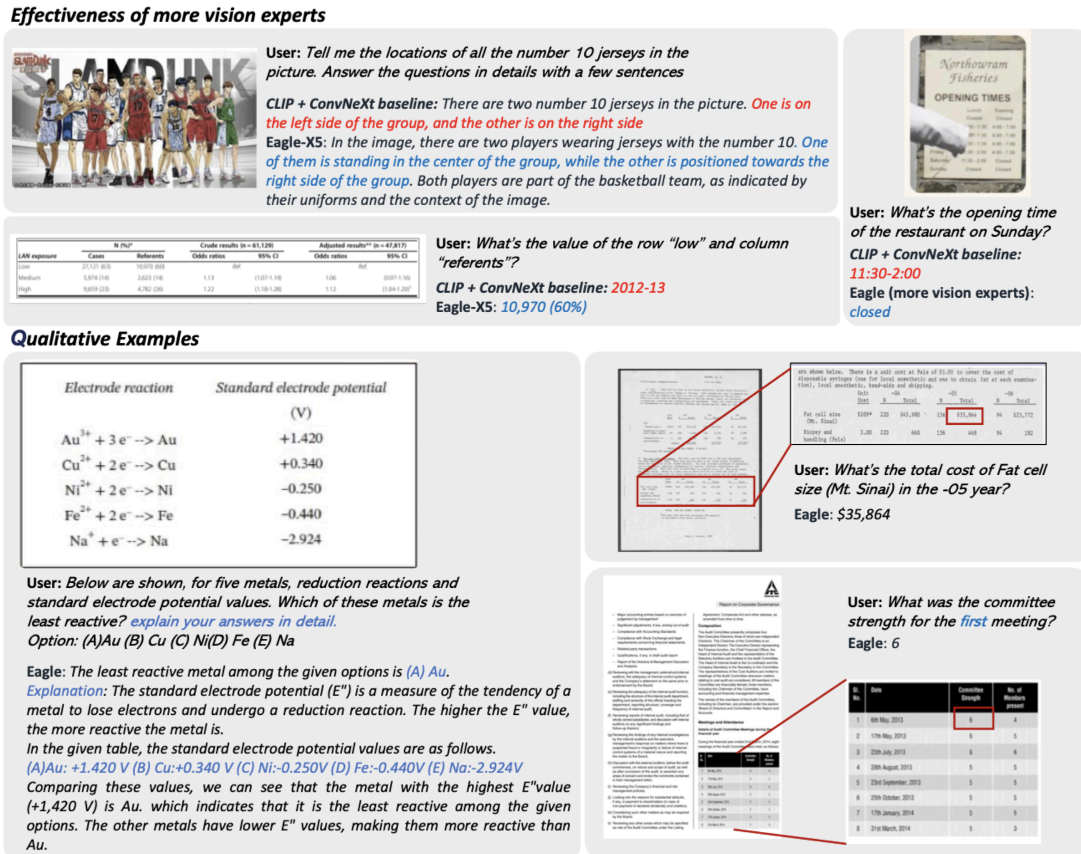


Fig 5.12. ***Qualitative comparison of different numbers of vision experts.*** *Baseline means Eagle with only CLIP+ConvNext. More Vision Experts denote the Eagle-X5 model. We highlight a failure case in RED. BLUE indicates the correct answers. With more vision experts, Eagle can more precisely capture the information in the image and provide the correct answer.*

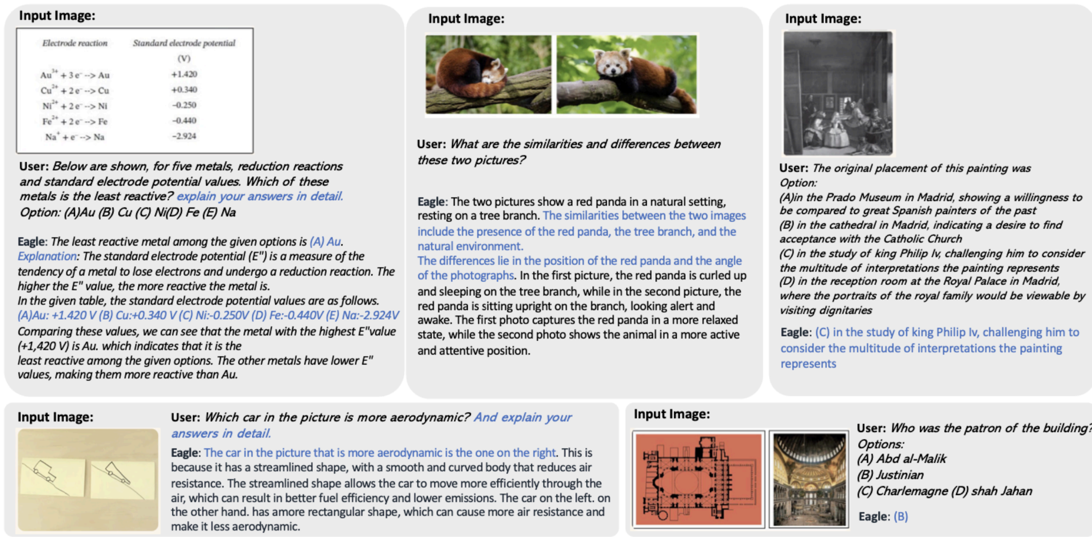


Fig 5.13. ***Qualitative results of Eagle*** *on VQA tasks requiring college-level knowledge or deliberate reasoning.*

Fig 5.14. ***Qualitative samples on OCR and document understanding tasks.*** *Eagle is able to extract useful information from small text.*

5.3.3 Main Takeaways

Our systematic study leads to several interesting new findings:

1. ***Unlocking the vision encoders during MLLM training matters.*** This is in sharp contrast to the LLaVA family and many works that consider multiple vision encoders or teachers where freezing the vision encoders has been a common choice.
2. Some recently proposed fusion strategies do not show significant advantages despite their advanced designs. Instead, we find that ***straightforward channel concatenation stands out as a simple yet competitive fusion strategy, offering the best efficiency and performance.***
3. ***Incorporating additional vision experts leads to consistent gain***, making it a promising path to systematically enhance MLLM perception besides scaling up single encoders. The improvement is particularly pronounced when vision encoders are unlocked.
4. ***We propose a pre-alignment stage where non-text-aligned vision experts are individually fine-tuned with a frozen LLM before training together***. This stage is found to enhance the MLLM performance significantly under the mixture-of-vision-encoder design.

Chapter 6: Challenges and Future Directions

The research of hallucination in MLLMs is still at an early stage, leaving a variety of research problems to be explored. In this section, we delve into the future directions of this pivotal domain.

Reinforcement Learning from Human Feedback (RLHF). Multimodal models often generate inconsistent or implausible cross-modal outputs. Incorporating human feedback that evaluates cross-modal coherence and factual consistency can improve reliability. Future work should explore fine-grained RLHF methods tailored to multimodal alignment.

Chain-of-Thought (CoT) reasoning [84] improves model performance by explicitly generating intermediate steps, allowing the model to break down complex problems and reduce spurious answers. We find that for tasks like math problems, step-by-step generation is more likely to produce correct answers compared to generating a final answer directly.

Data-centric Challenges and Innovations. MLLMs rely heavily on large-scale data, making data quality, diversity, and bias critical concerns. To enhance model accuracy and reliability, future work should focus on improved data collection, augmentation, and calibration. This includes addressing data scarcity, expanding datasets through augmentation, and re-calibrating existing data to reduce bias and promote diversity.

Retrieval-Augmented Generation (RAG). Integrating external knowledge sources allows MLLMs to ground their responses in factual evidence, reducing the reliance on parametric memory and mitigating hallucination. RAG is particularly valuable in domains with limited training data or where data is private and cannot be publicly shared.

Chapter 7: Limitations

Lack of standardized benchmarks and evaluation metrics for assessing hallucinations in MLLMs. While various benchmarks exist, there is no unified standard. For example, POPE [7], one of the most widely used benchmarks, adopts a binary "Yes-or-No" format, which oversimplifies user interaction with MLLMs. Other benchmarks like HallusionBench [2] attempt to assess hallucinations in free-form generation but often depend on external tools such as expert vision models or LLMs, limiting their scalability and reliability. This lack of robust, universally accepted evaluation methods may impact the generalizability of our findings and highlights the need for future work to develop standardized, theoretically grounded, and practical benchmarks.

This thesis primarily focuses on visual hallucinations that can be directly verified against the image content. These include cases where the model generates objects that are not present or misidentifies visible elements in the image. **We intentionally exclude unverifiable hallucinations, which involve speculative or subjective interpretations that cannot be grounded in the visual input**. For example, describing an image of a person walking down the street as 'He is heading to work' introduces assumptions about intent that are not visually evident. Addressing these types of hallucinations requires a different evaluation framework.

Another limitation of our work is that it primarily frames hallucination as a negative phenomenon to be minimized. However, recent perspectives suggest that hallucination may also be a valuable and even desirable feature of large language and multimodal models. As discussed in recent literature and social discourse, hallucinations can be seen as part of the model's creative process, where user prompts initiate a generative "*dream*" shaped by the model's learned representations. From this angle, hallucinations may enrich user experiences by enabling more imaginative, surprising, or inspiring outputs, especially in domains where factual correctness is not the sole priority. Future work could explore how to harness hallucination positively in downstream applications, shifting the optimization focus from benchmark accuracy to user satisfaction and engagement.

Chapter 8: Conclusions

This thesis presents a comprehensive investigation into the visual hallucination problem in Multimodal Large Language Models, a pressing challenge at the intersection of computer vision and natural language processing. While recent advancements have led to increasingly capable Vision-Language Models, our findings underscore a persistent and critical limitation: the tendency of these models to prioritize textual priors over visual evidence, often leading to incorrect or misleading outputs. To address this gap, we make several original and foundational contributions to the field:

1. ***We provide the first systematic taxonomy of hallucinations in MLLMs,*** offering a principled framework for understanding and categorizing failure modes.
2. ***We introduce HALLUSIONBENCH, the first diagnostic benchmark specifically tailored for hallucination analysis in multimodal settings.*** Its design, centered around visually grounded questions, enables fine-grained, quantitative assessment of model behavior.
3. ***We propose LRV-Instruction, a novel data-centric mitigation method that enhances instruction tuning via the generation of semantically controlled negative examples.*** This is the ***first*** work to demonstrate how data augmentation strategies can directly target hallucination in vision-language tasks.

4. ***We develop and evaluate NvEagle, a robust architecture that integrates vision encoder selection and resolution-aware design, significantly advancing the robustness and accuracy of MLLMs.***

Together, these contributions not only diagnose but also begin to resolve the hallucination issue by offering both theoretical and practical advancements. The thesis establishes a strong foundation for future research in building trustworthy, visually grounded multimodal systems.

Looking forward, we emphasize the importance of holistic strategies that combine architecture, data, and evaluation. We envision MLLMs that not only excel at perception and reasoning but also adaptively calibrate their confidence based on the reliability of each modality. This direction holds promise not just for reducing hallucinations, but for enabling a new generation of AI systems that are both capable and aligned with the real world.

Chapter 9: Related Work

9.1 Large Multi-Modal Models

Large Language Models [86, 89, 90] have significantly advanced the field of artificial intelligence, enabling models to perform a wide range of tasks purely from language inputs. This foundation has inspired the development of Large Vision-Language Models [47, 48, 49, 67, 68, 69] that integrate both textual and visual understanding within a unified architecture. One of the early milestones in this direction is Flamingo [21], which incorporates a frozen vision encoder with a large autoregressive language model enhanced by cross-attention layers, allowing it to perform visual reasoning without needing to retrain the vision backbone. Building upon even larger-scale architectures, PaLM-E [22] extends the 540B-parameter PaLM model by feeding in visual embeddings alongside textual inputs, demonstrating strong performance across robotics and real-world embodied reasoning tasks.

More recently, with the emergence of GPT-4 and GPT-4V [48], researchers have begun leveraging these powerful multimodal models to generate high-quality and diverse image-text instruction data, which in turn has enabled the fine-tuning of increasingly capable open-source LVLMs. Notable examples include LLaVA [6], MiniGPT-4 [23], mPLUG-Owl [24], LRV-Instruction [1], LLaVAR [25], and others [12, 14], which show promising results across visual question answering, image captioning, and multimodal dialogue. These models demonstrate a trend toward scalable and instruction-aligned multimodal systems, often bootstrapped by synthetic datasets derived from proprietary models.

9.2 Hallucination in LVLMs

Hallucination typically refers to situations where the generated responses contain information that is not grounded in the visual content. In LVLMs, this problem is particularly pronounced due to the complex fusion of visual and textual modalities. Prior research on hallucinations generally focuses on two main areas: evaluation and detection [2, 5, 40, 41, 42], and mitigation strategies [1, 3, 26, 27]. Early approaches include training binary classifiers to flag hallucinated outputs or comparing generated responses with human-written ground truths to measure factual consistency. Some works also propose visual-textual alignment scores or introduce external verification modules to validate the grounding of each claim.

To reduce hallucinations, recent efforts have emphasized improving both training data quality and model alignment strategies. For instance, LRV-Instruction [1] enhances instruction tuning with carefully balanced positive and negative image-text pairs, helping the model differentiate between grounded and ungrounded information. VIGC [27] introduces an iterative answer refinement approach, where multiple concise responses are generated, filtered, and combined to produce a final, more grounded answer. Another notable method, Woodpecker [28], offers a training-free correction mechanism, which uses an external vision-language verifier to detect hallucinated segments in the generated text and revise them accordingly—allowing for post-hoc correction without updating the base model. Other techniques aim to improve hallucination robustness via contrastive learning, visual grounding supervision, or prompting techniques. Additionally, models such as Kosmos-2 [44] leverage OCR and object detection tools during pretraining to enhance multimodal grounding, thereby reducing the chances of fabricated object references.

Despite these advancements, hallucination remains a persistent challenge, especially in open-ended tasks like visual dialogue, storytelling, and instruction following. Current benchmarks such as POPE [2] and MMHalBench [5] help quantify the extent of hallucinations, but standardized and fine-grained evaluation remains an open problem. Future research is likely to focus on tighter integration between perception and reasoning, as well as on interactive and retrieval-augmented LVLMs that can verify claims against external knowledge or image context in real time.

9.3 Benchmarks for Large VL Models

Traditional Visual Language (VL) benchmarks are designed to assess distinct skills, including visual recognition, image description, and so on. However, with the advent of advanced LVLMs, traditional evaluation metrics often fall short of providing a detailed ability assessment. This problem is further exacerbated by their inability to match the given answer accurately, leading to significant robustness issues. To address these challenges, research communities have introduced a series of benchmarks, including MME [18], MMBench [29], MM-Vet [30], SEED-Bench [31] and GAVIE [1]. These benchmarks systematically structure and evaluate complex multi-modal tasks.

As for MLLM hallucination benchmarks, As one of the early works, the metric of CHAIR [41] was proposed to evaluate object hallucination in the traditional image captioning task. This is achieved by computing what proportion of words generated are actually in the image according to the ground truth sentences and object segmentations. Compared to CHAIR, POPE [5] offers increased stability and flexibility. Based on this metric design, it further proposed an evaluation benchmark, drawing 500 images from the MSCOCO dataset. The questions in the benchmark consist of both positive and negative questions. MME [18] is a comprehensive evaluation benchmark for MLLMs. It covers the examination of perception and cognition abilities, encompassing 14 subtasks. FaithScore [84] aims to evaluate free-form responses to open-ended questions. Different from LLM-based overall assessment, FaithScore designs an automatic pipeline to decompose the response, evaluate, and analyze the elements in detail.

Bibliography

1. Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Aligning large multi-modal model with robust instruction tuning. In ICLR, 2024.
2. Fuxiao Liu, TianruiGuan, Zongxia Li, LichangChen, YaserYacoob, Dinesh Manocha, and TianyiZhou. HallusionBench: You see what you think? or you think what you see? an image-context reasoning benchmark challenging for gpt-4v (ision), llava-1.5, and other multi-modality models. In CVPR, 2024.
3. Fuxiao Liu, Xiaoyang Wang, Wenlin Yao, Jianshu Chen, Kaiqiang Song, Sangwoo Cho, Yaser Yacoob, and Dong Yu. MMC: Advancing Multimodal Chart Understanding with Large-scale Instruction Tuning. In NAACL, 2024.
4. Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In CVPR, 2024.
5. Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In EMNLP, 2023.
6. Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In NeurIPS, 2023.
7. Shikun Liu, Linxi Fan, Edward Johns, Zhiding Yu, Chaowei Xiao, and Anima Anandkumar. Prismer: A vision-language model with an ensemble of experts. TMLR, 2024.

8. Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi,Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. In ICLR, 2024.
9. Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. arXiv:2312.11805, 2023.
10. Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. IEEE Trans. PAMI, 2024
11. Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In CVPR, 2024.
12. Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In ICML, 2022.
13. Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun,Yuliang Liu,and Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal models. In CVPR, 2024.

14. Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. VILA: On pre-training for visual language models. arXiv:2312.07533, 2023.
15. Hao Fei, Yuan Yao, Zhuosheng Zhang, Fuxiao Liu, Ao Zhang, and Tat-Seng Chua. From multimodal llm to human-level ai: Modality, instruction, reasoning, efficiency and beyond. In LREC-Coling Tutorials, 2024.
16. Wenliang Dai, Junnan Li, Dongxu Li, Anthony MengHuat Tiong, Junqi Zhao,Weisheng Wang,Boyang Li, Pascale N Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In NeurIPS, 2024.
17. Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun Ledell Wu, Xinggan Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. EVA: Exploring the limits of masked visual representation learning at scale. In CVPR, 2023
18. Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, et al. MME: A comprehensive evaluation benchmark for multimodal large language models. arXiv:2306.13394, 2023.
19. Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, YanWang, ZihanWang, Yuxiao Dong, Ming Ding, et al. CogAgent: A visual language model for gui agents. In CVPR, 2024
20. Kenton Lee, Mandar Joshi,Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and

Kristina Toutanova. Pix2Struct: Screenshot parsing as pretraining for visual language understanding. In ICML, 2023.

21. Alayrac, Jean-Baptiste, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc et al. "Flamingo: a visual language model for few-shot learning." In NeurIPS, 2022.
22. Driess, Danny, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid et al. "Palm-e: An embodied multimodal language model." arXiv preprint arXiv:2303.03378 (2023).
23. Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. "Minigpt-4: Enhancing vision-language understanding with advanced large language models." arXiv preprint arXiv:2304.10592 (2023).
24. Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang et al. "mplug-owl: Modularization empowers large language models with multimodality." arXiv preprint arXiv:2304.14178 (2023).
25. Yanzhe Zhang, Ruiyi Zhang, Jiuxiang Gu, Yufan Zhou, Nedim Lipka, Diyi Yang, and Tong Sun. "Llavar: Enhanced visual instruction tuning for text-rich image understanding." arXiv preprint arXiv:2306.17107 (2023).
26. Min Shi, Fuxiao Liu, Shihao Wang, Shijia Liao, Subhashree Radhakrishnan, De-An Huang, Hongxu Yin et al. "Eagle: Exploring the design space for multimodal llms with mixture of encoders." arXiv preprint arXiv:2408.15998 (2024).

27. Bin Wang, Fan Wu, Xiao Han, Jiahui Peng, Huaping Zhong, Pan Zhang, Xiaoyi Dong et al. "Vigc: Visual instruction generation and correction." In AAAI, 2024.

28. Shukang Yin, Chaoyou Fu, Sirui Zhao, Tong Xu, Hao Wang, Dianbo Sui, Yunhang Shen, Ke Li, Xing Sun, and Enhong Chen. "Woodpecker: Hallucination correction for multimodal large language models." Science China Information Sciences, 2024

29. Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan et al. "Mmbench: Is your multi-modal model an all-around player?." In ECCV, 2024.

30. Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. "Mm-vet: Evaluating large multimodal models for integrated capabilities." In ICML 2024.

31. Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. "Seed-bench: Benchmarking multimodal llms with generative comprehension." arXiv preprint arXiv:2307.16125 (2023).

32. Achiam, Josh, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida et al. "Gpt-4 technical report." arXiv preprint arXiv:2303.08774 (2023).

33. Viraj Prabhu, Senthil Purushwalkam, An Yan, Caiming Xiong, and Ran Xu. "Trust but Verify: Programmatic VLM Evaluation in the Wild." arXiv preprint arXiv:2410.13121 (2024).

34. Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. "Convnext v2: Co-designing and scaling convnets with masked autoencoders." In CVPR, 2023.
35. Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez et al. "Dinov2: Learning robust visual features without supervision." arXiv preprint arXiv:2304.07193 (2023).
36. Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. "Docvqa: A dataset for vqa on document images." In WACV, 2021.
37. Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao et al. "Segment anything." In ICCV, 2023.
38. Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, Si Wei. "Neural Natural Language Inference Models Enhanced with External Knowledge." In ACL 2018.
39. Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, Mike Zheng Shou. "Hallucination of Multimodal Large Language Models: A Survey". In Arxiv 2024.
40. Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, Ting Liu. "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions." In ACM TOIS 2024.
41. Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. "Object hallucination in image captioning". In EMNLP 2018.

42. Jialian Wu, Jianfeng Wang, Zhengyuan Yang, Zhe Gan, Zicheng Liu, Junsong Yuan, and Lijuan Wang. "Grit: A generative region-to-text transformer for object understanding". In Arxiv 2022.
43. Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, et al. "Cambrian-1: A Fully Open, Vision-Centric Exploration of Multimodal LLMs". In NeurIPS 2024.
44. Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, Furu Wei. "Kosmos-2: Grounding Multimodal Large Language Models to the World". In Arxiv 2023.
45. Tao Tu, Shekoofeh Azizi, Danny Driess, Mike Schaekermann, Mohamed Amin, Pi-Chuan Chang, Andrew Carroll, Charles Lau, Ryutaro Tanno, Ira Ktena, et al. "Towards Generalist Biomedical AI." In NEJM AI 2024.
46. Shekoofeh Azizi, Basil Mustafa, Fiona Ryan, Zachary Beaver, Jan Freyberg, Jonathan Deaton, Aaron Loh, Alan Karthikesalingam, Simon Kornblith, Ting Chen, et al. "Big Self-Supervised Models Advance Medical Image Classification." In ICCV 2021.
47. Yucheng Han, Chi Zhang, Xin Chen, Xu Yang, Zhibin Wang, Gang Yu, Bin Fu, Hanwang Zhang. "ChartLLaMA: A Multimodal LLM for Chart Understanding and Generation." In Arxiv 2023.
48. KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, Yu Qiao. "VideoChat: Chat-Centric Video Understanding." In ArXiv 2023.

49. Hang Zhang, Xin Li, Lidong Bing. "Video-LLaMA: An Instruction-Tuned Audio-Visual Language Model for Video Understanding." In ArXiv 2023.

50. Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. "Conceptual Captions: A Cleaned, Hypernymed, Image Alt-Text Dataset for Automatic Image Captioning." In ACL 2018.

51. Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. "Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training to Recognize Long-Tail Visual Concepts." In CVPR 2021.

52. Vicente Ordonez, Girish Kulkarni, and Tamara Berg. "Im2Text: Describing Images Using 1 Million Captioned Photographs." In NeurIPS 2011.

53. Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. "LAION-5B: An Open Large-Scale Dataset for Training Next Generation Image-Text Models." In NeurIPS 2022.

54. Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. "LAION-400M: Open Dataset of CLIP-Filtered 400 Million Image-Text Pairs." In arXiv 2021.

55. Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. "COYO-700M: Image-Text Pair Dataset." In GitHub 2022.

56. Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. "Microsoft COCO Captions: Data Collection and Evaluation Server." In arXiv 2015.
57. Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. "Adaptive Token Sampling for Efficient Vision Transformers." In ECCV 2022.
58. Shixing Yu, Tianlong Chen, Jiayi Shen, Huan Yuan, Jianchao Tan, Sen Yang, Ji Liu, and Zhangyang Wang. "Unified Visual Transformer Compression." In ArXiv 2022.
59. Tianlong Chen, Yu Cheng, Zhe Gan, Lu Yuan, Lei Zhang, and Zhangyang Wang. "Chasing Sparsity in Vision Transformers: An End-to-End Exploration." In NeurIPS 2021.
60. Lorenzo Papa, Paolo Russo, Irene Amerini, and Luping Zhou. "A Survey on Efficient Vision Transformers: Algorithms, Techniques, and Performance Benchmarking." In TPAMI 2024.
61. Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. "Distilling the Knowledge in a Neural Network." In arXiv 2015.
62. Dayou Du, Gu Gong, and Xiaowen Chu. "Model Quantization and Hardware Acceleration for Vision Transformers: A Comprehensive Survey." In ArXiv 2024.
63. Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. "Post-Training Quantization for Vision Transformer." In NeurIPS 2021.

64. Junrui Xiao, Zhikai Li, Lianwei Yang, and Qingyi Gu. "BinaryViT: Towards Efficient and Accurate Binary Vision Transformers." In ArXiv 2023.
65. Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. "GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints." In 2023.
66. Hen Lin, Bo Peng, Zheyang Li, Wenming Tan, Ye Ren, Jun Xiao, and Shiliang Pu. "Bit-Shrinking: Limiting Instantaneous Sharpness for Improving Post-Training Quantization." In CVPR 2023.
67. Yanjing Li, Sheng Xu, Baochang Zhang, Xianbin Cao, Peng Gao, and Guodong Guo. "Q-ViT: Accurate and Fully Quantized Low-Bit Vision Transformer." In NeurIPS 2022.
68. Sheng Xu, Yanjing Li, Teli Ma, Bohan Zeng, Baochang Zhang, Peng Gao, and Jinhu Lv. "TerViT: An Efficient Ternary Vision Transformer." In arXiv 2022.
69. Yefei He, Zhenyu Lou, Luoming Zhang, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. "BiViT: Extremely Compressed Binary Vision Transformers." In ICCV 2023.
70. Peiyan Dong, Lei Lu, Chao Wu, Cheng Lyu, Geng Yuan, Hao Tang, and Yanzhi Wang. "PackQViT: Faster Sub-8-bit Vision Transformers via Full and Packed Quantization on the Mobile." In NeurIPS 2024.
71. Phuoc-Hoan Charles Le and Xinlin Li. "BinaryViT: Pushing Binary Vision Transformers Towards Convolutional Models." In CVPR 2023.
72. Chong Yu, Tao Chen, Zhongxue Gan, and Jiayuan Fan. "Boost Vision Transformer with GPU-Friendly Sparsity and Quantization." In CVPR 2023.

73. Zhengang Li, Mengshu Sun, Alec Lu, Haoyu Ma, Geng Yuan, Yanyue Xie, Hao Tang, Yanyu Li, Miriam Leeser, Zhangyang Wang, et al. "Auto-ViT-Acc: An FPGA-Aware Automatic Acceleration Framework for Vision Transformer with Mixed-Scheme Quantization." In FPL 2022.
74. Zhiyuan Zhao, Bin Wang, Linke Ouyang, Xiaoyi Dong, Jiaqi Wang, and Conghui He. "Beyond Hallucinations: Enhancing LVLMs through Hallucination-Aware Direct Preference Optimization." In ArXiv 2023.
75. Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. "LIMA: Less is More for Alignment." In ArXiv 2023.
76. Qiang Zhou, Zhibin Wang, Wei Chu, Yinghui Xu, Hao Li, and Yuan Qi. "InfMLLM: A Unified Framework for Visual-Language Tasks." In ArXiv 2023.
77. Yiyang Zhou, Chenhang Cui, Rafael Rafailov, Chelsea Finn, and Huaxiu Yao. "Aligning Modalities in Vision Large Language Models via Preference Fine-Tuning." In ArXiv 2024.
78. Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. "Analyzing and Mitigating Object Hallucination in Large Vision-Language Models." In ArXiv 2023.
79. Lanyun Zhu, Deyi Ji, Tianrun Chen, Peng Xu, Jieping Ye, and Jun Liu. "IBD: Alleviating Hallucinations in Large Vision-Language Models via Image-Biased Decoding." In ArXiv 2024.

80. Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Lingpeng Kong, Jiajun Chen, Lei Li, and Shujian Huang. "Multilingual Machine Translation with Large Language Models: Empirical Results and Analysis." In ArXiv 2023.

81. Lei Wang, Jiabang He, Shenshen Li, Ning Liu, and Ee-Peng Lim. "Mitigating Fine-Grained Hallucination by Fine-Tuning Large Vision-Language Models with Caption Rewrites." In arXiv 2023.

82. Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. "CogVLM: Visual Expert for Pretrained Language Models." In arXiv 2023.

83. Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V. Le, Denny Zhou, et al. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models." In NeurIPS 2022.

84. Liqiang Jing, Ruosen Li, Yunmo Chen, Xinya Du. "FaithScore: Fine-grained Evaluations of Hallucinations in Large Vision-Language Models.." In EMNLP 2024.

85. Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Tao He, Haotian Wang, Weihua Peng, Ming Liu, Bing Qin, and Ting Liu. "A Survey of Chain of Thought Reasoning: Advances, Frontiers and Future." In ArXiv 2023.

86. Chenhang Cui, Yiyang Zhou, Xinyu Yang, Shirley Wu, Linjun Zhang, James Zou, and Huaxiu Yao. "Holistic Analysis of Hallucination in GPT-4V(ision): Bias and Interference Challenges." In ArXiv 2023.

87. Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang,

Matthew Aitchison, Laurent Orseau, et al. "Language Modeling is Compression." In ArXiv 2023.

88. Ailin Deng, Zhirui Chen, and Bryan Hooi. "Seeing is Believing: Mitigating Hallucination in Large Vision-Language Models via CLIP-Guided Decoding." In ArXiv 2024.
89. Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. "GLM: General Language Model Pretraining with Autoregressive Blank Infilling." In ACL 2022.
90. Ross Girshick. "Fast R-CNN." In ICCV 2015.
91. Tao Gong, Chengqi Lyu, Shilong Zhang, Yudong Wang, Miao Zheng, Qian Zhao, Kuikun Liu, Wenwei Zhang, Ping Luo, and Kai Chen. "MultiModal-GPT: A Vision and Language Model for Dialogue with Humans." In ArXiv 2023.
92. Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. "Mask R-CNN." In ICCV 2017.
93. Xin He, Longhui Wei, Lingxi Xie, and Qi Tian. "Incorporating Visual Experts to Resolve the Information Loss in Multimodal Large Language Models." In ArXiv 2024.
94. Google. "Bard." https://bard.google.com/, 2023.
95. Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. "Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering." In CVPR 2017.