SCIENCE CHINA
Information Sciences

December 2024, Vol. 67, Iss. 12, 220105:1–220105:13
https://doi.org/10.1007/s11432-024-4251-x

• RESEARCH PAPER •
Special Topic: Large Multimodal Models

Woodpecker: hallucination correction for multimodal large language models

Shukang YIN[1†], Chaoyou FU[2,3*†], Sirui ZHAO[1*†], Tong XU[1*], Hao WANG[1], Dianbo SUI[4], Yunhang SHEN[5], Ke LI[5], Xing SUN[5] & Enhong CHEN[1*]

[1]*School of Artificial Intelligence and Data Science, University of Science and Technology of China, Hefei* 230026, *China;*
[2]*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing* 210023, *China;*
[3]*School of Intelligence Science and Technology, Nanjing University, Suzhou* 215163, *China;*
[4]*Institute of Automation, Chinese Academy of Sciences, Beijing* 100190, *China;*
[5]*YouTu, Shanghai* 200233, *China*

Received 8 March 2024/Revised 12 October 2024/Accepted 30 November 2024/Published online 13 December 2024

Abstract Hallucinations is a big shadow hanging over the rapidly evolving multimodal large language models (MLLMs), referring to that the generated text is inconsistent with the image content. To mitigate hallucinations, existing studies mainly resort to an instruction-tuning manner that requires retraining the models with specific data. In this paper, we pave a different way, introducing a training-free method named Woodpecker. Like woodpeckers heal trees, it picks out and corrects hallucinations from the generated text. Concretely, Woodpecker consists of five stages: key concept extraction, question formulation, visual knowledge validation, visual claim generation, and hallucination correction. Implemented in a post-remedy manner, Woodpecker can easily serve different MLLMs, while being interpretable by accessing intermediate outputs of the five stages. We evaluate Woodpecker both quantitatively and qualitatively and show the huge potential of this new paradigm. On the POPE benchmark, our method obtains a 30.66%/24.33% improvement in accuracy over the baseline MiniGPT-4/mPLUG-Owl. The source code is released at https://github.com/BradyFU/Woodpecker.

Keywords multimodal learning, multimodal large language models, hallucination correction, large language models, vision and language

1 Introduction

Multimodal large language models (MLLMs) [1] are now flourishing in the research community, working towards artificial general intelligence (AGI). By exploiting powerful large language models (LLMs), researchers align foreign modalities like vision with language, and develop MLLMs with various exciting capabilities [2–6], such as fully describing the contents of a given image.

However, as strong as these MLLMs are, they sometimes output descriptions that are inconsistent with the input image. It is called hallucination and has been found prevalent in MLLMs [7]. As exemplified by Figure 1, the MLLM claims non-existent objects and fails to describe the attribute of the object in the image accurately, which are categorized by us as object-level and attribute-level hallucinations, respectively. It is obvious that these hallucinations are huge obstacles to the practical application of MLLMs.

To mitigate the hallucinations, existing studies usually explore an instruction-tuning way [7, 8]. A common key observation is that MLLMs tend to hallucinate when generating longer text [7], which results in different problem-solving strategies. For example, LRV-Instruction [7] takes an intuitive approach by limiting the text length of instruction data. As a consequence, the tuned model usually generates less hallucinated but also less detailed descriptions. VIGC [8] takes a multi-step generation scheme and iteratively updates the visual features with the textual context, which relieves hallucinations via

* Corresponding author (email: bradyfu24@gmail.com, sirui@mail.ustc.edu.cn, tongxu@ustc.edu.cn, cheneh@ustc.edu.cn)
† Yin S K, Fu C Y, and Zhao S R have the same contribution to this work.

© Science China Press 2024

Figure 1 (Color online) Illustration of hallucinations in MLLMs. Given an image, an MLLM outputs a corresponding response with both object-level (marked in blue) and attribute-level (marked in red) hallucinations.

sacrificing generative efficiency. Moreover, both of the two methods are instruction-tuning-based and thus are data- and computation-intensive.

To break the limitation, we take a different strategy that can directly correct the hallucinations without retraining. As illustrated in Figure 2, given a text generated by MLLMs as well as the input image, our training-free framework Woodpecker corrects the text elaborately, and meanwhile, provides the corresponding evidence, i.e., the bounding boxes. It adds interpretability and reliability beyond the black-box MLLMs, providing convenient visual fact-checking. Concretely, our framework performs correction after a thorough diagnosis, which incorporates a total of five stages: (1) Key concept extraction identifies the main objects mentioned in the generated sentences. (2) Question formulation asks questions around the extracted objects, such as their number and attributes. (3) Visual knowledge validation answers the formulated questions via expert models. For example, a visual perception model can be used to determine the object number. (4) Visual claim generation converts the above question-answer (QA) pairs into a visual knowledge base, which consists of the object-level and attribute-level claims about the input image. (5) Hallucination correction modifies the hallucinations and adds the corresponding evidence under the guidance of the visual knowledge base. It is worth noting that each step in the pipeline is clear and transparent, which offers good interpretability.

We evaluate the effectiveness of our method through comprehensive quantitative and qualitative experiments on the POPE [9], MME [10], and LLaVA-QA90 [2] datasets. The results and associated analyses indicate the superiority of this new paradigm. For instance, on the POPE benchmark, our method largely boosts the accuracy of the baseline MiniGPT-4 [4]/mPLUG-Owl [3] from 54.67%/62% to 85.33%/86.33%.

In summary, the main contributions are as follows:

- We propose a training-free framework named Woodpecker to correct the hallucinations for MLLMs. To the best of our knowledge, we are the first to apply a corrective manner to tackle the visual hallucination problem.
- Our framework is designed in such a way that each step is clear and transparent, thus providing good interpretability.
- We comprehensively evaluate the effectiveness of our method, and the large improvements demonstrate its great potential in hallucination correction.

2 Related work

Hallucination in MLLM. Recently, there has been increasing attention on the hallucination phenomenon of MLLMs. This is mainly because the issue directly affects the reliability of MLLMs. Current researches on the hallucination of MLLMs mainly focus on two aspects, i.e., the evaluation/detection [9, 11, 12] and mitigation [7, 8, 13]. The previous line of work generally either trains a classification model

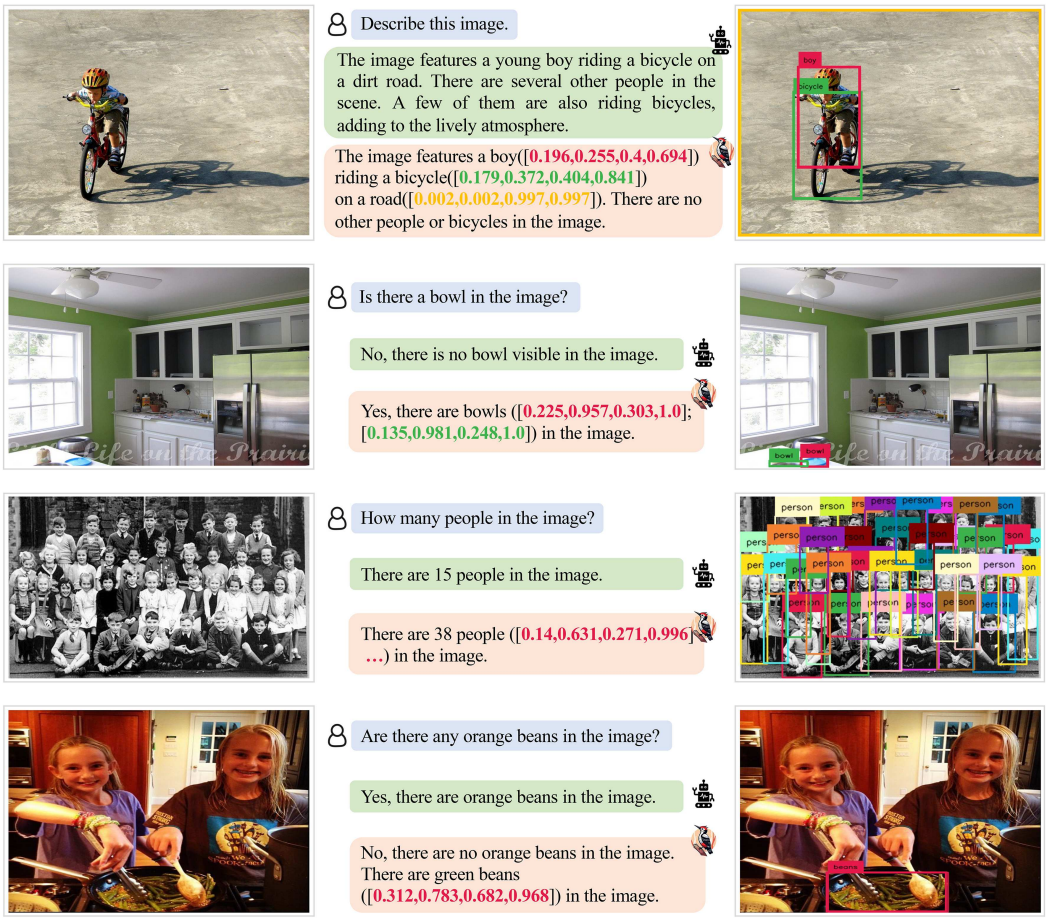

Figure 2 (Color online) Examples of our framework for hallucination correction. Given a response of an MLLM, Woodpecker corrects the hallucinated parts and incorporates grounding information for ease of verification.

to discriminate hallucination [12] or checks the output text against ground-truth answers to decide if the hallucination happens [9, 11].

For hallucination mitigation, previous studies have explored different ways to suppress hallucinations. Fundamentally, hallucinations can be attributed to two factors: objects not grounded by the vision encoders or bias in data/LLMs. The first line of work accentuates that hallucinations happen when objects fail to be grounded by vision encoders [14, 15]. Therefore, a practical way is further tuning the model parameters and improving upon this regard, such as scaling up the input resolution [14]. Another line considers bias in the training data that leads to false statistical correlations [9, 16] or model bias inherent in LLMs [17]. From this perspective, the way our work deals with hallucinations is in line with the first route, but instead resorts to external visual knowledge to augment models, thus mitigating hallucinations in MLLMs.

From a broader perspective, using vision models to help solve real-life questions is very common, covering aerial image detection [18], 3D object detection [19], and robotic grasp detection [20]. Some studies [21, 22] also addressed tasks utilizing both visual and linguistic information.

Knowledge-augmented LLM. Since LLMs are limited to the inherent knowledge gained from pre-training, various studies have been dedicated to augmenting LLMs with external knowledge sourced from a pre-defined knowledge base [23–26] or the internet [27, 28]. As a natural extension of this idea, recently, researchers have explored using knowledge as evidence to alleviate factual hallucinations in LLMs [29, 30]. Specifically, these studies use relevant knowledge as background information to refine a possibly false input claim, resulting in a higher factuality of the response. Our methods share in common with the idea that we use information relevant to the given image to correct potentially wrong claims. However, it is non-trivial to transfer the idea to the vision-language field. This is because the language-only counterpart usually deals with text only and acquires relevant knowledge through retrieval to mitigate hallucinations

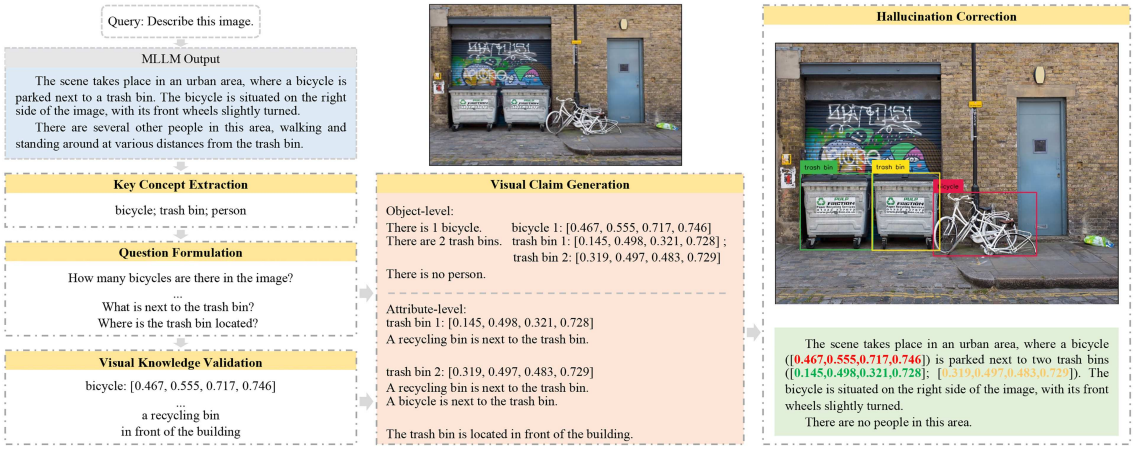

Figure 3 (Color online) Framework of Woodpecker. Given an image and a query, an MLLM outputs the corresponding response. Through the four steps, including key concept extraction, question formulation, visual knowledge validation, and visual claim generation, we get a visual knowledge base specific to the image and the original response. In the final step, the hallucinations in the response are corrected with the bounding boxes as evidence.

(factual fallacies, in this context). But when it comes to the vision-language realm, it is much more complicated to check model responses against image content and thus correct visual hallucinations. To cope with the complexity, our method proposes a unified framework to invoke appropriate tools for hallucination mitigation. Corresponding to the key differences, in this work, we devise a strategy to construct a structured visual knowledge base conditioned on the image and the query. We also explore how to address both object-level and attribute-level hallucinations in an organized way, as we will illustrate later.

LLM-aided visual reasoning. According to the taxonomy in the survey [1], our proposed framework is closely related to the LLM-aided visual reasoning model. The main idea is that we can leverage the strong reasoning and instruction-following capabilities of LLMs to help fulfill vision or multimodal tasks. Typical roles that LLMs play include the task dispatcher [31–34], the reasoner [35–37], or the language refiner [38–40]. In this work, we utilize the strong reasoning and language proficiencies of LLMs to help the processes of key concept extraction, question formulation, and hallucination correction.

3 Method

Our objective is to diagnose and correct the hallucinations in the response generated by MLLMs. The key challenges lie in locating the hallucinations and determining the facts, which can be organized in a structured way for final correction. To this end, we break down the whole process into five subtasks: key concept extraction (Subsection 3.1), question formulation (Subsection 3.2), visual knowledge validation (Subsection 3.3), visual claim generation (Subsection 3.4), and hallucination correction (Subsection 3.5). We will illustrate each step in sequence later. An overview of our framework is depicted in Figure 3.

3.1 Key concept extraction

Since descriptions usually revolve around key concepts, the first step is to extract them from the generated sentence. To this end, we identify the main objects mentioned in the sentence, which are the ones most likely to exit visual hallucinations. For instance, given a sentence "`The man is wearing a black hat.`", the objects "`man`" and "`hat`" are extracted, and will serve as the center for diagnosis in the following steps. In light of the strong generalization ability and rich world knowledge of LLMs, we prompt an LLM to fulfill this task.

The template for key concept extraction, which comprises a system message and a formatted prompt, is listed in Appendix A.1. The former sets up the basic context for the LLM, while the latter starts with some detailed descriptions of the task and some requirements, followed by several in-context examples and inputs. The in-context examples are provided so that the LLM could better understand the requirements in terms of the task.

3.2 Question formulation

After acquiring the key concepts, we ask a series of questions around them to make the hallucination diagnosis. Our questions are directed at both object-level and attribute-level hallucinations. For the former, we ask, "`Is there any {object} in the image? How many are there?`", where "`{object}`" is the key concept extracted earlier. For the latter, various questions involving the attributes of objects can be formulated, such as "`What is {object} doing?`", "`Is {object`$_1$ `on the right side of {object`$_2$`}?`", and "`What color is the {object}?`", where "`{object`$_1$`}`" and "`{object`$_2$`}`" are two different key concepts.

In fact, object-level questions can be directly validated through perceiving images, while attribute-level questions are much more diverse and dependent on the context. To facilitate such free-form formulation of questions, we prompt an LLM with some in-context examples so that meaningful questions are raised. The prompt is listed in Appendix A.2.

3.3 Visual knowledge validation

This step is responsible for solving the two types of questions above. For the object-level questions, the crux is determining the existence and the count of a certain object. In light of the strong perception capabilities of vision foundation models [41,42], we employ an open-set object detector as the solver [43]. For attribute-level questions, we apply a pre-trained VQA model [44] to answer the questions conditioned on the image. Compared with mainstream MLLMs, the VQA model tends to generate shorter answers but also with fewer hallucinations and thus can be a reasonable choice.

3.4 Visual claim generation

After questions are raised and answered, we combine QA pairs into visual claims and organize them into a visual knowledge base for reference in the following step. The visual knowledge base is structured by the following.

• Object-level claims: This part of the information mainly plays a role in mitigating object-level hallucinations. We include information about object counts of key concepts extracted from the sentences (Subsection 3.1). For existing objects, we add a claim as "`There are {counts} {name}.`", where "`{counts}`" and "`{name}`" are the counts and the name of a certain kind of object. We use a similar template, "`There is no {name}`", for nonexistent objects. The counting information comes from the open-set object detection in the previous step.

• Attribute-level claims: We include attribute information specific to each object in order to alleviate attribute-level hallucinations. Typical attributes include positions, colors, and actions. For this part, we adopt a QA-to-Claim model [29] to merge questions and answers into claims. To cope with cases involving multiple objects or the relationship between the foreground objects and the background, more global information is needed. Thus, we also include claims that involve the interaction between different objects or the objects and the background, such as "`The cat is lying next to the dog.`".

3.5 Hallucination correction

Guided by the visual claims, an LLM can act as a corrector and modify the hallucinations in the generated responses. Specifically, after combining the visual knowledge base with the original responses into a prompt, we instruct an LLM to correct the responses and output the refined ones. For better interpretability, we explicitly instruct the LLM to attach bounding boxes right behind expressions when referring to objects. This design facilitates the correspondence between the mentioned entities in the responses and object instances in the image, which provides convenient access to check the reliability of the output. The prompt template for correction is included in Appendix A.3.

4 Experiment

4.1 Experimental settings

Dataset. LLaVA-QA90 [2] is also used to evaluate MLLMs. Specifically, we sample 10 description-type queries that are paraphrased in various forms to instruct an MLLM to describe an image, such

as "`Describe the following image.`" and "`What is the photo about?`". LLaVA-QA90 uses images from COCO and adopts text-only GPT-4 [45] to compose queries and reference answers. We discard the reference answers, directly feed the image to GPT-4V [45], and prompt it to rate the responses regarding our designed two dimensions, i.e., accuracy and detailedness. The prompt template of evaluation is available in Appendix A.4.

VaLLu [46] is a comprehensive benchmark developed to evaluate the cognitive capabilities of MLLMs, with a diverse cover of task types. The samples are sourced from existing benchmarks and manually filtered and annotated. Rather than using binary Yes/No or multi-choice questions, the benchmark adopts open-ended evaluation and includes metrics like helpfulness, clarity, and factuality.

OVEN [47] is a benchmark that focuses on visual entity recognition in the open domain. Each text query refers to an object in the image and potentially links to one of the six million potential Wikipedia entities. Thus, the task tests models on capabilities of identifying various entities in real-world scenarios and requires abundant world knowledge.

POPE [9] is dedicated to evaluating hallucinations of MLLMs. It contains the settings of random, popular, and adversarial sampling, which mainly differ in the way negative samples are constructed. For the random setting, the objects not presented in the image are sampled randomly, while for the popular setting, non-existent objects are sampled from a pool of objects with the highest frequencies. For the adversarial setting, objects that most frequently co-occur but do not exist in the image are sampled.

In terms of the sampling setting, we sample 50 images and build 6 questions for each image. The ratio between positive and negative samples is balanced, namely 50% vs. 50%. This setup transforms object annotations into a series of "Yes-or-No" questions and focuses on evaluating the object-level hallucination, and more specifically, the existence aspect. Thereby, MLLMs are prompted to answer if an object exists in the image or not. Accordingly, evaluation metrics include accuracy, precision, recall, and F1-Score.

MME [10] is a comprehensive benchmark designed to evaluate the performance of MLLMs in various aspects. It encompasses ten subtasks for the perception ability and four subtasks for the cognition ability, respectively. In this paper, we repurpose the dataset and select existence and count subsets to measure the object-level hallucination. The position and color subsets are used to measure the attribute-level hallucination. Similar to the setup of POPE, each subset is composed of "Yes-or-No" questions. We report the score, namely the sum of accuracy and accuracy+ following the official implementation [10], in which a higher score indicates better performance and fewer hallucinations.

Baselines. We choose mainstream MLLMs as our baseline models, such as mPLUG-Owl [3], LLaVA [2], Otter [48], and MiniGPT-4 [4]. These MLLMs follow a "vision encoder-interface-language model" architecture [1] and are trained on image-text pairs. Specifically, LLaVA and MiniGPT-4 adopt a simple projection layer to align multimodal embeddings. mPLUG-Owl uses a Q-Former [44] to compress visual features into a fixed number of tokens, which can be concatenated with the language embeddings. Otter adopts a similar Perceiver [49] resampler to obtain the token compression.

Implementation details. Our pipeline is training-free and comprises three pre-trained models apart from the MLLM to be corrected. We choose the LLM, GPT-3.5-turbo [50], to fulfill the subtasks of key concept extraction, question formulation, and hallucination correction. For open-set object detection, we use Grounding DINO [43] to extract object counting information with default detection thresholds. Moreover, we utilize BLIP-2-FlanT5$_{\mathrm{XXL}}$ [44] as the VQA model to answer the attribute-related questions conditioned on the input image.

For the "Yes-or-No" questions, we find that the instruction-following ability of some MLLMs is somewhat weak, often outputting irrelevant texts such as pure emojis or URLs. This is an obstacle to our correction process. Besides, some MLLMs only output a single "Yes" or "No", which also poses a challenge to the correction. To deal with these issues, we design two simple measures: (1) We first extract keywords, i.e., "Yes" and "No" from the responses as the answers, then combine the questions with the answers into more specific claims. For example, given a question, "`Is there a dog in the image?`" and a model answer, "Yes", we compose a more specific answer as "`Yes, there is a dog in the image.`". (2) We additionally feed the questions to the LLM in the correction process so that the LLM can have a better grasp of the context and task requirements.

4.2 Experimental results

Results on LLaVA-QA90. The description-type queries of LLaVA-QA90 instruct MLLMs to fully translate the input image into language rather than merely referring to the existence or the attribute of

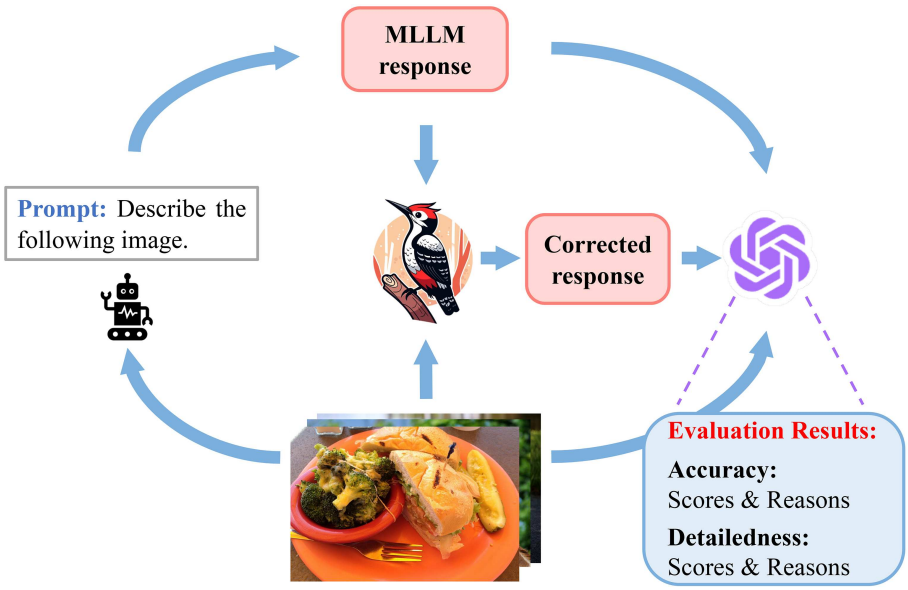

Figure 4 (Color online) Illustration of GPT-4V-aided evaluation.

Table 1 Results of GPT-4V-aided evaluation on LLaVA-QA90[a)]

Method	w/Ours	Accuracy	Detailedness
LLaVA [2]	✗	7.1	7.1
	✓	7.8	8.6
MiniGPT-4 [4]	✗	7.0	6.4
	✓	8.2	8.8
mPLUG-Owl [3]	✗	5.4	6.4
	✓	5.7	6.4
Otter [48]	✗	7.0	6.7
	✓	8.5	8.8

a) The accuracy and detailedness metrics are on a scale of 10, and a higher score indicates better performance.

an object. Therefore, a more reasonable and comprehensive manner is needed to support the evaluation of such open answers. Some existing efforts are devoted to exploring automatic evaluation with the aid of LLM [2, 7]. Specifically, a text-only GPT-4 is adopted, and the image content is fed to the language model in the form of short captions and bounding boxes of some objects. Nevertheless, the process of image-to-text translation inevitably loses a lot of information, making the evaluation process potentially inaccurate and biased.

In light of the recent release of a strong MLLM, GPT-4V, we propose to evaluate via a more straightforward approach. As shown in Figure 4, GPT-4V can directly receive the original response, the corrected ones, and most importantly, the input image. In such a case, we can prompt GPT-4V to let it give evaluation results and reasons for judgment.

Concretely, we structure the original and the corrected responses into a prompt and ask the GPT-4V to rate the responses grounded in the image. The evaluation process is implemented via the ChatGPT web interface. To meet our needs, we devise the following two metrics.

- Accuracy: whether the response is accurate with respect to the image content.
- Detailedness: whether the response is rich in details.

The two criteria focus on different aspects of responses from multimodal agents. Specifically, "Accuracy" reflects the correctness of descriptions with regard to the image. Therefore, a higher accuracy score can be interpreted as fewer hallucinations. On the other hand, "Detailedness" is included to obviate the case where answers are correct while the information is limited. The scores of the two metrics are displayed in Table 1, from which we can see that our method achieves consistent gains over the baseline MLLMs. On the one hand, the improvement in accuracy suggests that our Woodpecker can effectively correct the hallucinations in MLLM responses. On the other hand, the bounding box information introduced in our framework adds details to the response, contributing to the boost in detailedness.

Table 2 Results on the VaLLu benchmark[a)]

Method	w/Ours	Helpfulness	Clarity	Factuality
LLaVA [2]	✗	2.61	3.69	1.95
	✓	2.68	3.81	2.01
LLaVA-v1.5 [51]	✗	1.86	2.90	1.47
	✓	2.15	2.93	1.57
mPLUG-Owl2 [52]	✗	2.73	3.64	2.20
	✓	2.89	3.90	2.23
CogVLM [53]	✗	3.25	4.19	2.82
	✓	3.44	4.30	2.91

a) w/Ours denotes MLLM responses corrected by our proposed Woodpecker. The metrics are on a scale of 5, and a higher score indicates better performance.

Table 3 Results on the OVEN benchmark[a)]

Method	w/Ours	Helpfulness	Clarity	Factuality
LLaVA [2]	✗	3.12	4.41	2.44
	✓	3.21	4.49	2.46
LLaVA-v1.5 [51]	✗	3.25	4.41	2.66
	✓	3.26	4.42	2.46
mPLUG-Owl2 [52]	✗	3.23	4.42	2.86
	✓	3.27	4.50	2.90
CogVLM [53]	✗	3.58	4.51	3.35
	✓	3.70	4.58	3.45

a) The metrics are the same as the VaLLu benchmark.

Results on VaLLu & OVEN. Similar to the LLaVA-QA90 benchmark, VaLLu and OVEN evaluate open-ended responses of MLLMs. The results are listed in Tables 2 and 3 [2,51–53], respectively. As shown in Tables 2 and 3, our proposed framework brings consistent gains across different metrics on various MLLMs. Specifically, the "Factuality" gains can be attributed to the expert models introduced in the correction framework, while "Helpfulness" and "Clarity" can be attributed to the language proficiency of LLM, which organizes and presents responses in a more reasonable way. Notably, the gains in the OVEN benchmark suggest an augmentation in world knowledge, demonstrating the general effectiveness of our methods in the open domain.

Results on POPE. The results on POPE under the random, popular, and adversarial settings are summarized in Table 4. It can be seen that, in the random setting, MiniGPT-4 is relatively weak in perception capabilities, specifically in judging the existence of objects. The F1-Score for MiniGPT-4 is only 43.33%, while other baselines are all over 70%. In addition, mPLUG-Owl and Otter tend to be overconfident, as reflected by a high Yes Rate. Meanwhile, the high recall and the low precision result in a relatively low F1-Score. For all of the baselines, Woodpecker achieves consistent gains in most metrics, which indicates that our method has the ability to effectively correct object-level hallucinations. Specifically, Woodpecker obtains a relative gain of 30.66% for MiniGPT-4 and 24.33% for mPLUG-Owl in terms of accuracy.

In the more challenging popular and adversarial settings, MLLMs show performance degradation to different extents, more prominent in relatively stronger baselines, such as LLaVA. Specifically, compared with the random setting, LLaVA shows a 9.33% and 12.67% accuracy degradation in the popular and adversarial settings, respectively. This tendency suggests that MLLMs may incorrectly fit some data characteristics in the training corpus. For example, the decline in the popular setting may stem from the long-tailed data distribution [9]. In contrast, equipped with a robust expert vision model, our correction method shows strong stability, making obvious improvements in various metrics for the baselines, where all accuracies exceed 80%. Particularly, our Woodpecker largely boosts the accuracy of mPLUG-Owl from 56.33% to 81% in the adversarial setting.

Results on MME. Compared with POPE, the experiment on MME is more well-rounded since it covers not only object-level but also attribute-level hallucination evaluation. The corresponding results are listed in Table 5. We can see that, for object-level evaluation, LLaVA and Otter excel in the existence aspect, which is also verified in the POPE evaluation, while they relatively lag in answering harder count queries. In this case, our correction method is particularly effective and contributes a large score gain,

Table 4 Results on POPE[a)]

Setting	Method	w/Ours	Accuracy	Precision	Recall	F1-Score	Yes rate
Random	LLaVA [2]	✗	86.00	87.50	84.00	85.71	48.00
		✓	**87.67**	**95.93**	78.67	**86.45**	41.00
	MiniGPT-4 [4]	✗	54.67	57.78	34.67	43.33	30.00
		✓	85.33	92.06	77.33	84.06	42.00
	mPLUG-Owl [3]	✗	62.00	57.26	**94.67**	71.36	82.67
		✓	86.33	93.60	78.00	85.09	41.67
	Otter [48]	✗	72.33	66.18	91.33	76.75	69.00
		✓	86.67	93.65	78.67	85.51	42.00
Popular	LLaVA [2]	✗	76.67	72.22	86.67	78.79	60.00
		✓	80.67	83.82	76.00	79.72	45.33
	MiniGPT-4 [4]	✗	56.67	58.77	44.67	50.76	38.00
		✓	82.33	85.40	78.00	81.53	45.67
	mPLUG-Owl [3]	✗	57.33	54.20	**94.67**	68.93	87.33
		✓	83.00	84.14	81.33	82.71	48.33
	Otter [48]	✗	67.33	61.71	91.33	73.66	74.00
		✓	**84.33**	**88.15**	79.33	**83.51**	45.00
Adversarial	LLaVA [2]	✗	73.33	69.02	84.67	76.05	61.33
		✓	80.67	82.86	77.33	80.00	46.67
	MiniGPT-4 [4]	✗	55.00	56.88	41.33	47.88	36.33
		✓	82.33	83.92	80.00	81.91	47.67
	mPLUG-Owl [3]	✗	56.33	53.51	**96.67**	68.88	90.33
		✓	81.00	82.07	79.33	80.68	48.33
	Otter [48]	✗	66.67	61.16	91.33	73.26	74.67
		✓	**83.00**	**85.61**	79.33	**82.35**	46.33

a) w/Ours denotes MLLM responses corrected by our proposed Woodpecker. The best and second-to-best performance within each setting is in bold and underlined, respectively.

Table 5 Results on MME[a)]

Method	w/Ours	Object-level		Attribute-level		Total
		Existence	Count	Position	Color	
LLaVA [2]	✗	195.00	95.00	53.33	78.33	421.67
	✓	195.00	160.00	55.00	155.00	565.00
MiniGPT-4 [4]	✗	100.00	61.67	53.33	65.00	280.00
	✓	183.33	**163.33**	60.00	121.67	528.33
mPLUG-Owl [3]	✗	101.67	73.33	58.33	66.67	300.00
	✓	**200.00**	131.67	**78.33**	145.00	555.00
Otter [48]	✗	185.00	95.00	50.00	118.33	448.33
	✓	195.00	160.00	51.67	**165.00**	**571.67**

a) w/Ours denotes MLLM responses corrected by our proposed Woodpecker. The performance is measured by scores, where the best and second-to-best for each partition are in bold and underlined, respectively.

ranging from +65 over LLaVA to +101.66 over MiniGPT-4. With regard to attribute-level evaluation, baseline MLLMs tend to achieve poorer results, which suggests that they are more prone to attribute-level hallucinations. For example, MiniGPT-4 only achieves a score of 65 in the color split, and mPLUG-Owl merely attains 66.67. After introducing our correction framework, these MLLMs make consistent and remarkable gains, where the score of mPLUG-Owl goes up 78.33. In contrast, the improvements in position are relatively small, which may be caused by two factors: (1) the relatively weak ability of the VQA model BLIP-2 in position reasoning; (2) LLM fails to comprehend the given bounding boxes well enough to derive positional relationships by itself.

Comparison with other methods. We compare our methods with two other representative methods, LRV-Instruction [7] and VIGC [8]. The results shown in Table 6 suggest that both LRV-Instruction and VIGC achieve performance gains only in some benchmarks, but downgrade in others. This is because these methods are based on extra finetuning on data, which inevitably introduces bias to the models and makes it hard to generalize and perform well in all the benchmarks. In contrast, we enhance MLLMs in

Table 6 Comparison with other hallucination mitigation methods[a)]

Method	POPE			MME				LLaVA-QA90	
	Rand.	Pop.	Adv.	Exst.	Count	Pos.	Color	Acc.	Detl.
MiniGPT-4 (baseline)	43.3	50.8	47.9	100.0	61.7	53.3	65.0	7.0	6.4
LRV-Instruction	42.5(−0.8)	58.8(+8.0)	57.8(+9.9)	70.0(−30.0)	51.7(−10.0)	55.0(+1.7)	58.3(−6.7)	1.3(−5.7)	1.4(−5.0)
VIGC	66.7(+23.4)	66.9(+16.1)	66.8(+18.9)	50.0(−50.0)	48.3(−13.4)	50.0(−3.3)	50.0(−15.0)	5.8(−1.2)	4.9(−1.5)
Ours	84.1(+40.8)	81.5(+30.7)	81.9(+34.0)	183.3(+83.3)	163.3(101.6)	60.0(+6.7)	121.7(+56.7)	8.2(+1.2)	8.8(+2.4)

a) Rand., Pop., and Adv. suggest Random, Popular, and Adversarial settings, respectively, measured in F1-Score. Exst. and Pos. indicate Existence and Position. Acc. and Detl. mean Accuracy and Detailedness.

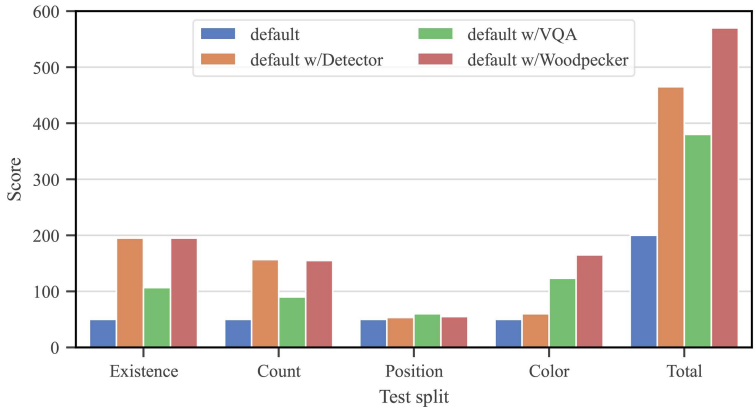

Figure 5 (Color online) Results on MME with different framework variants. "default" is a model that always answers "Yes", "default w/Detector" introduces the object detector for hallucination correction, and "default w/VQA" introduces the VQA model. "default w/Woodpecker" is our full framework.

the most foundational perception capabilities, thereby boosting performance comprehensively.

4.3 Experimental analysis

Analysis of framework modules. To understand the roles of different modules and their synergy, we take a dive into them and their ensemble. For the purpose of avoiding distractions from the variation of MLLMs, we formulate a simple test bench by casting a "default" model that always answers "Yes". Afterward, the answers and the questions are merged into more specific claims. For example, given a question, "`Is there a train in the picture? Please answer yes or no.`", we compose an answer of the default model as "`Yes, there is a train in the picture.`". Furthermore, we create two extra variants of our framework, one of which only includes the open-set detector and the other with only the VQA model, respectively dubbed "default w/Detector" and "default w/VQA":

• default w/Detector. This variant is designed to probe the contribution of the detector on mitigating object-level hallucinations, more specifically, the existence and count aspects of hallucinations.

• default w/VQA. By designing this variant, we aim to study the effectiveness of our selected VQA model in providing attribute information.

The former is implemented by only providing object-level information in the knowledge base, while the latter is realized by providing attribute-level information. We compare these two variants with our proposed full framework, i.e., "default Woodpecker", which uses both two types of information.

As shown in Figure 5, the gains in terms of existence and count splits mainly derive from the introduction of the open-set detector, and the improvement in the color part can be attributed to the application of the VQA model. This is in line with the expectation since we collect count information by means of the detector and gather information about specific attributes, i.e., position and color, via the VQA model. Consequently, the full model combines the advantages of both modules and achieves the best results.

To give an intuitive comprehension of the results of correction and the GPT-4V-aid evaluation, we offer some cases in Appendix B. Specifically, we list the query and the MLLM response before and after correction. For reference, scores and reasons given by GPT-4V are also listed.

Analysis of correction performance. In this part, we aim to probe further the performance of correction. Since there is a lack of related studies in measuring the correction behavior, we fulfill this goal by breaking down the results after correction into three sections.

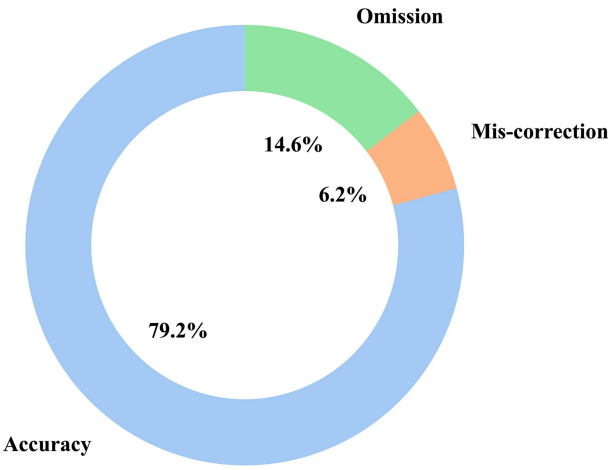

Figure 6 (Color online) Proportion of different correction results.

Table 7 Statistics of operation efficiency of the whole framework. We run 100 instances and calculate the mean value for each of the efficiency metrics

Module	Time (s)	Params	FLOPs
LLaVA	7.5	13.1B	63T
Detection	0.7	172M	464G
VQA	0.3	12.4B	1T
LLM	15.3	–	–
Total	23.8	–	–

- Accuracy: |correct answers kept and wrong answers corrected|/|problems|.
- Omission: |wrong responses that fail to be corrected|/|problems|.
- Mis-correction: |correct responses mistakenly modified|/|problems|.

Concretely, we summarize the results of the "default" model on MME and calculate the three introduced metrics. As reflected in Figure 6, our correction method reaches an accuracy of 79.2%, and meanwhile, the omission and mis-correction rates remain at a relatively low level. The results indicate that our method can cover most cases without being overconfident.

Analysis of operating efficiency. In this part, we offer an analysis of computing efficiency and complexities in terms of our methods, as shown in Table 7. We sample 100 images from the COCO2014-val dataset and use the prompt "Describe this image." to calculate efficiency metrics in an end-to-end way. We use the MLLM, LLaVA-13B, to benchmark the efficiency. The computation experiment is performed on NVIDIA A800 GPUs.

As shown in Table 7, the two introduced experts, i.e., detection and VQA expert, bring a slight latency compared with the MLLM computation. The main source of the delay is the LLM, which performs various types of NLP tasks. Overall, the average token length output of MLLM is 211.5, and the correction latency is around 16.3 s. It should be noted that just like how LLMs progress, we first explore the upper bound of performance for this new method, leaving space for efficiency optimization in the future, such as using more lightweight LLMs to perform linguistic operations and streamlining the whole pipeline.

5 Conclusion

In this work, we have proposed the first correction-based framework for mitigating hallucinations in MLLMs. As a training-free method, our approach incorporated multiple off-the-shelf models and could be easily integrated into different MLLMs. To evaluate the efficacy of the proposed framework, we conduct massive experiments on three benchmarks under different settings, including using GPT-4V for direct and automatic assessment. We hope this work can spark new thoughts on addressing the issue of hallucinations in MLLMs.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. U23A20319, 62222213, U22B2059, 61727809, 62072423) and Young Scientists Fund of the Natural Science Foundation (Grant No. 2023NSFSC1402).

Supporting information Appendixes A–C. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Yin S, Fu C, Zhao S, et al. A survey on multimodal large language models. Natl Sci Open, 2024. doi: 10.1093/nsr/nwae403
2 Liu H, Li C, Wu Q, et al. Visual instruction tuning. In: Proceedings of Conference on Neural Information Processing Systems, 2023
3 Ye Q, Xu H, Xu G, et al. mPLUG-Owl: modularization empowers large language models with multimodality. 2023. ArXiv:2304.14178
4 Zhu D, Chen J, Shen X, et al. MiniGPT-4: enhancing vision-language understanding with advanced large language models. In: Proceedings of International Conference on Learning Representations, 2024
5 Zhang A, Fei H, Yao Y, et al. Transfer visual prompt generator across LLMs. In: Proceedings of Conference on Neural Information Processing Systems, 2023
6 Bai J, Bai S, Yang S, et al. Qwen-VL: a versatile vision-language model for understanding, localization, text reading, and beyond. 2023. ArXiv:2308.12966
7 Liu F, Lin K, Li L, et al. Mitigating hallucination in large multi-modal models via robust instruction tuning. In: Proceedings of International Conference on Learning Representations, 2023
8 Wang B, Wu F, Han X, et al. VIGC: visual instruction generation and correction. In: Proceedings of AAAI, 2024
9 Li Y, Du Y, Zhou K, et al. Evaluating object hallucination in large vision-language models. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, 2023
10 Fu C, Chen P, Shen Y, et al. MME: a comprehensive evaluation benchmark for multimodal large language models. 2023. ArXiv:2306.13394
11 Wang J, Zhou Y, Xu G, et al. Evaluation and analysis of hallucination in large vision-language models. 2023. ArXiv:2308.15126
12 Gunjal A, Yin J, Bas E. Detecting and preventing hallucinations in large vision language models. In: Proceedings of AAAI, 2024
13 Lu J, Rao J, Chen K, et al. Evaluation and mitigation of agnosia in multimodal large language models. In: Proceedings of AAAI (Workshop on ReLM), 2024
14 Li Z, Yang B, Liu Q, et al. Monkey: image resolution and text label are important things for large multi-modal models. In: Proceedings of Conference on Computer Vision and Pattern Recognition, 2024
15 Zhai B, Yang S, Zhao X, et al. Halle-switch: rethinking and controlling object existence hallucinations in large vision language models for detailed caption. 2023. ArXiv:2310.01779
16 Leng S, Zhang H, Chen G, et al. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In: Proceedings of Conference on Computer Vision and Pattern Recognition, 2024
17 Huang Q, Dong X, Zhang P, et al. OPERA: alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. In: Proceedings of Conference on Computer Vision and Pattern Recognition, 2024
18 Xie X X, Cheng G, Li Q Y, et al. Fewer is more: efficient object detection in large aerial images. Sci China Inf Sci, 2024, 67: 112106
19 Zhang D Y, Liang D K, Yang H C, et al. SAM3D: zero-shot 3D object detection via the segment anything model. Sci China Inf Sci, 2024, 67: 149101
20 Song Y N, Gao L, Li X Y, et al. A novel vision-based multi-task robotic grasp detection method for multi-object scenes. Sci China Inf Sci, 2022, 65: 222104
21 Yang Y, Bao R, Guo W L, et al. Deep visual-linguistic fusion network considering cross-modal inconsistency for rumor detection. Sci China Inf Sci, 2023, 66: 222102
22 Xiao K, Zhu A N, Iwana B K, et al. Scene text recognition via dual character counting-aware visual and semantic modeling network. Sci China Inf Sci, 2024, 67: 139101
23 Dinan E, Roller S, Shuster K, et al. Wizard of Wikipedia: knowledge-powered conversational agents. In: Proceedings of International Conference on Learning Representations, 2019
24 Petroni F, Piktus A, Fan A, et al. KILT: a benchmark for knowledge intensive language tasks. In: Proceedings of North American Chapter of the Association for Computational Linguistics, 2020
25 Lewis P, Perez E, Piktus A, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Proceedings of Conference on Neural Information Processing Systems, 2020
26 Borgeaud S, Mensch A, Hoffmann J, et al. Improving language models by retrieving from trillions of tokens. In: Proceedings of International Conference on Machine Learning, 2022
27 Shuster K, Xu J, Komeili M, et al. BlenderBot 3: a deployed conversational agent that continually learns to responsibly engage. 2022. ArXiv:2208.03188
28 Piktus A, Petroni F, Karpukhin V, et al. The web is your Oyster-knowledge-intensive NLP against a very large web corpus. 2021. ArXiv:2112.09924
29 Huang K H, Chan H P, Ji H. Zero-shot faithful factual error correction. In: Proceedings of Association for Computational Linguistics, 2023
30 Peng B, Galley M, He P, et al. Check your facts and try again: improving large language models with external knowledge and automated feedback. 2023. ArXiv:2302.12813
31 Shen Y, Song K, Tan X, et al. HuggingGPT: solving AI tasks with ChatGPT and its friends in hugging face. 2023. ArXiv:2303.17580
32 Yang R, Song L, Li Y, et al. GPT4Tools: teaching large language model to use tools via self-instruction. In: Proceedings of Conference on Neural Information Processing Systems, 2023
33 Lu P, Peng B, Cheng H, et al. Chameleon: plug-and-play compositional reasoning with large language models. In: Proceedings of Conference on Neural Information Processing Systems, 2023
34 Gupta T, Kembhavi A. Visual programming: compositional visual reasoning without training. In: Proceedings of Conference on Computer Vision and Pattern Recognition, 2023

35 You H, Sun R, Wang Z, et al. IdealGPT: iteratively decomposing vision and language reasoning via large language models. In: Proceedings of Findings of the Association for Computational Linguistics, 2023

36 Zhu D, Chen J, Haydarov K, et al. ChatGPT asks, BLIP-2 answers: automatic questioning towards enriched visual descriptions. Transactions on Machine Learning Research, 2024. https://openreview.net/forum?id=1LoVwFkZNo

37 Yang Z, Li L, Wang J, et al. MM-REACT: prompting ChatGPT for multimodal reasoning and action. 2023. ArXiv:2303.11381

38 Wu C, Yin S, Qi W, et al. Visual ChatGPT: talking, drawing and editing with visual foundation models. 2023. ArXiv:2303.04671

39 Zhang R, Hu X, Li B, et al. Prompt, generate, then cache: cascade of foundation models makes strong few-shot learners. In: Proceedings of Conference on Computer Vision and Pattern Recognition, 2023

40 Zhu X, Zhang R, He B, et al. PointCLIP V2: prompting CLIP and GPT for powerful 3D open-world learning. In: Proceedings of International Conference on Computer Vision, 2023

41 Kirillov A, Mintun E, Ravi N, et al. Segment anything. In: Proceedings of International Conference on Computer Vision, 2023

42 Wang Z, Li Y, Chen X, et al. Detecting everything in the open world: towards universal object detection. In: Proceedings of Conference on Computer Vision and Pattern Recognition, 2023

43 Liu S, Zeng Z, Ren T, et al. Grounding DINO: marrying DINO with grounded pre-training for open-set object detection. In: Proceedings of European Conference on Computer Vision, 2023

44 Li J, Li D, Savarese S, et al. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In: Proceedings of International Conference on Machine Learning, 2023

45 OpenAI. GPT-4 Technical Report. 2023. ArXiv:2303.08774

46 Ghosh S, Evuru C K R, Kumar S, et al. VDGD: mitigating LVLM hallucinations in cognitive prompts by bridging the visual perception gap. 2024. ArXiv:2405.15683

47 Hu H, Luan Y, Chen Y, et al. Open-domain visual entity recognition: towards recognizing millions of Wikipedia entities. In: Proceedings of Conference on Computer Vision and Pattern Recognition, 2023

48 Li B, Zhang Y, Chen L, et al. Otter: a multi-modal model with in-context instruction tuning. 2023. ArXiv:2305.03726

49 Jaegle A, Gimeno F, Brock A, et al. Perceiver: general perception with iterative attention. In: Proceedings of International Conference on Machine Learning, 2021

50 Brown T, Mann B, Ryder N, et al. Language models are few-shot learners. In: Proceedings of Conference on Neural Information Processing Systems, 2020

51 Liu H, Li C, Li Y, et al. Improved baselines with visual instruction tuning. In: Proceedings of Conference on Computer Vision and Pattern Recognition, 2024

52 Ye Q, Xu H, Ye J, et al. mPLUG-Owl2: revolutionizing multi-modal large language model with modality collaboration. In: Proceedings of Conference on Computer Vision and Pattern Recognition, 2024

53 Wang W, Lv Q, Yu W, et al. CogVLM: visual expert for pretrained language models. In: Proceedings of Conference on Neural Information Processing Systems, 2024