Woodpecker: Hallucination Correction for Multimodal Large Language Models

Shukang Yin[1]*, **Chaoyou Fu**[2*‡†], **Sirui Zhao**[1*‡], **Tong Xu**[1‡], **Hao Wang**[1]
Dianbo Sui, **Yunhang Shen**[2], **Ke Li**[2], **Xing Sun**[2], **Enhong Chen**[1‡]
[1]School of Data Science, USTC & State Key Laboratory of Cognitive Intelligence
[2]Tencent YouTu Lab
{xjtupanda,sirui}@mail.ustc.edu.cn, {tongxu,cheneh}@ustc.edu.cn
{bradyfu24}@gmail.com

Abstract

Hallucination is a big shadow hanging over the rapidly evolving Multimodal Large Language Models (MLLMs), referring to the phenomenon that the generated text is inconsistent with the image content. In order 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 a woodpecker heals 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.

1. Introduction

Multimodal Large Language Models (MLLMs) [41] 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 [1, 21, 40, 44, 47], such as fully describe the contents of a given image.

Figure 1. Illustration of hallucinations in MLLMs. Given an image, an MLLM outputs a corresponding response with both object-level and attribute-level hallucinations.

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 [20]. As exemplified by Fig. 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.

In order to mitigate the hallucinations, existing works usually explore an instruction-tuning way [20, 33]. A common key observation is that MLLMs tend to hallucinate when generating longer text [20], which results in different problem-solving strategies. For example, LRV-Instruction [20] takes an intuitive approach by limiting the text length of instruc-

*Equal contribution.
†Project leader.
‡Corresponding author.

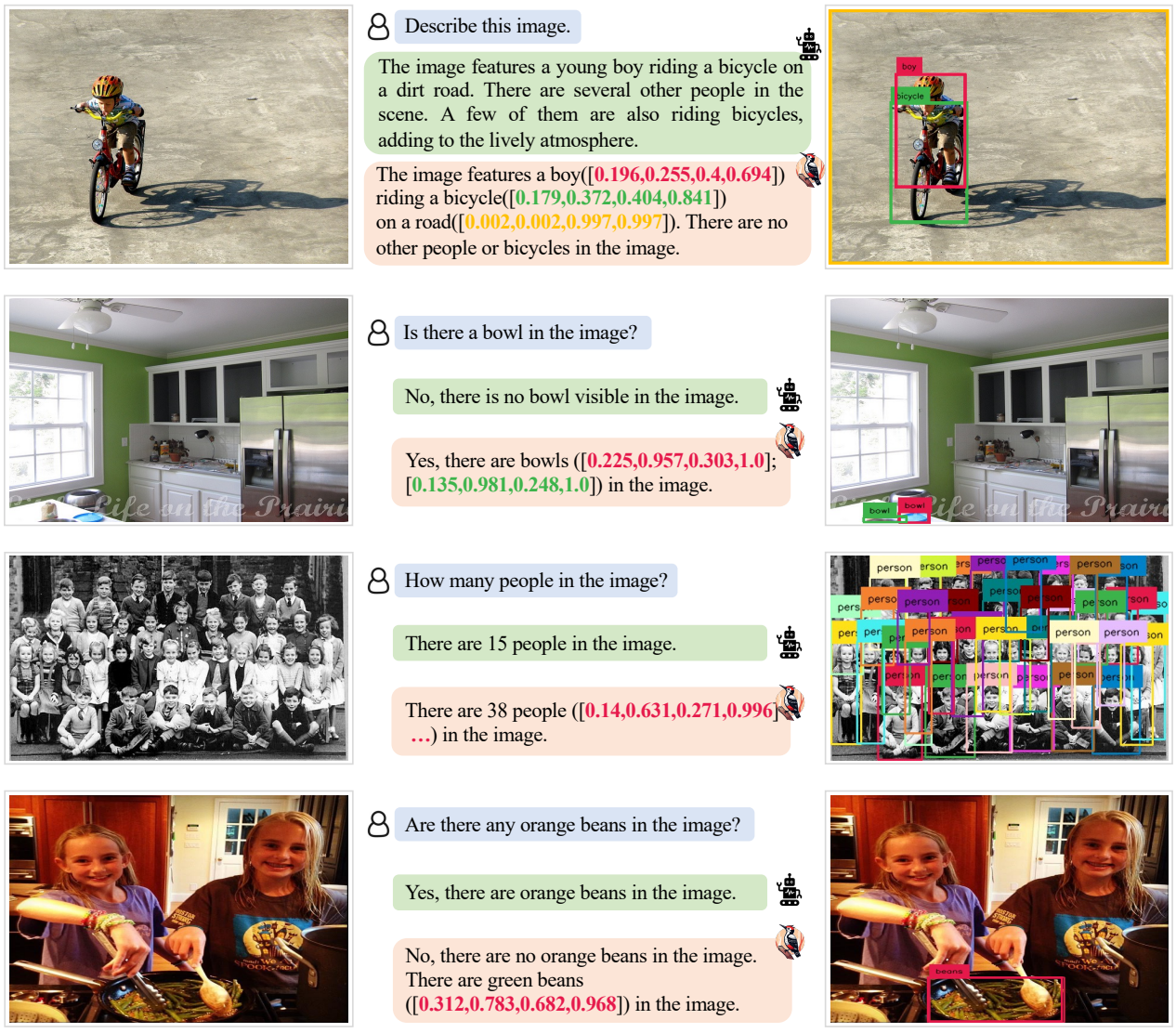

Figure 2. Examples of our framework for hallucination correction. Given a response of an MLLM, Woodpecker corrects the hallucinated parts and incorporates grounding information for ease of verification.

tion data. As a consequence, the tuned model usually generates less hallucinated but also less detailed descriptions. VIGC [33] takes a multi-step generation scheme and iteratively updates the visual features with the textual context, which relieves hallucinations via 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 Fig. 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 [18], MME [6], and LLaVA-QA90 [21] 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 [47]/mPLUG-Owl [40] 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 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

2.1. 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 [8, 18, 34] and mitigation [20, 23, 33]. The previous line of work generally either trains a classification model to discriminate hallucination [8] or checks the output text against ground-truth answers to decide if the hallucination happens [18, 34].

For hallucination mitigation, previous works focus on optimizing the data collection process and the training scheme. LRV-Instruction [20] composes negative instances to refrain from over-confidence. Moreover, the text length of Ground-Truth answers is strictly controlled, based on the observation that shorter responses are less likely to be hallucinated. Similarly, VIGC [33] takes an iterative process, where short answers are generated and concatenated each time. In this way, it tries to ensure accuracy without compromising details. While previous works try to develop MLLMs with fewer hallucinations, our main objective is to refine the responses of MLLMs by modifying the hallucinated parts. Specifically, we design a training-free framework that incorporates off-the-shelf models. This exempts the complexity of collecting instruction data and resource-intensive training. As a result, our framework can be easily integrated with various MLLMs, serving as a general plug-and-play module.

2.2. Knowledge-augmented LLM

Since LLMs are limited to the inherent knowledge gained from pretraining, various works have been dedicated to augmenting LLMs with external knowledge sourced from a predefined knowledge base [3, 5, 14, 28] or the internet [29, 31]. As a natural extension of this idea, recently, researchers have explored using knowledge as evidence to alleviate factual hallucinations in LLMs [10, 27]. Specifically, these works 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, while it is inappropriate to do so for image-text pairs. Moreover, knowledge-augmented LLMs pay more attention to alleviating factual fallacies, while we lay more stress on mitigating visual hallucinations. 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.

2.3. LLM-aided Visual Reasoning

According to the taxonomy in the survey [41], our proposed framework is closely related to the LLM-Aided Visual Reasoning model [2, 7, 13]. 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 [9, 24, 30, 38], the reasoner [37, 39, 42, 46], or the language refiner [35, 43, 45, 48]. 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 (Sec. 3.1), question formulation (Sec. 3.2), visual knowledge validation (Sec. 3.3), visual claim generation (Sec. 3.4), and hallucination correction (Sec. 3.5). We will illustrate each step in sequence later. An overview of our framework is depicted in Fig. 3.

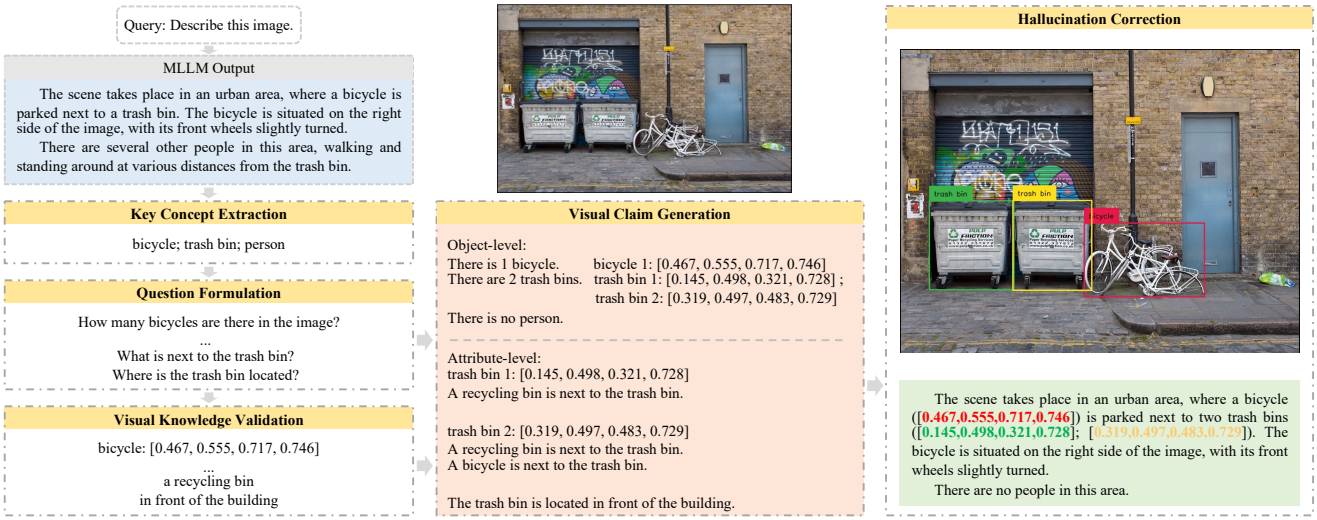

Figure 3. Framework of Woodpecker. Given an image and a query, an MLLM outputs the corresponding response. Through the four steps, including key concept extraction, question formulation, visual knowledge validation, and visual claim generation, we get a visual knowledge base specific to the image and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

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 is listed in Appendix A.1, which comprises a system message and a formatted prompt. 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 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 above two types of questions. 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 [12, 17, 25, 32, 36], we employ an open-set object detector as the solver [22]. For attribute-level questions, we apply a pre-trained VQA model [16] 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:

- 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 (Sec. 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, actions, *etc*. For this part, we adopt a QA-to-Claim model [10] to merge questions and answers into claims. In order 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. **POPE** [18] 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 [6] 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 [6], in which a higher score indicates better performance and fewer hallucinations.

LLaVA-QA90 [21] 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 [19] and adopts text-only GPT-4 [26] to compose queries and reference answers. We discard the reference answers, directly feed the image to GPT-4V [26], and prompt it to rate the responses regarding our designed two dimensions, *i.e*., accuracy and detailedness. The prompt template is available in Appendix A.4.

Baselines. We choose mainstream MLLMs as our baseline models, including mPLUG-Owl [40], LLaVA [21], MiniGPT-4 [47], and Otter [15]. These four MLLMs follow a "vision encoder-interface-language model" architecture [41] 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 [16] to compress visual features into a fixed number of tokens, which can be concatenated with the language embeddings. Otter adopts a similar Perceiver [11] 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 [4], to fulfill the subtasks of key concept extraction, question formulation, and hallucination correction. For open-set object detection, we use Grounding DINO [22] to extract object counting information with default detection thresholds. Moreover, we utilize BLIP-2-FlanT5$_{XXL}$ [16] 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

Setting	Method	w/Ours	Accuracy	Precision	Recall	F1-Score	Yes Rate
Random	LLaVA [21]	✗	86.00	87.50	84.00	85.71	48.00
		✓	**87.67**	**95.93**	78.67	**86.45**	41.00
	MiniGPT-4 [47]	✗	54.67	57.78	34.67	43.33	30.00
		✓	85.33	92.06	77.33	84.06	42.00
	mPLUG-Owl [40]	✗	62.00	57.26	**94.67**	71.36	82.67
		✓	86.33	93.60	78.00	85.09	41.67
	Otter [15]	✗	72.33	66.18	91.33	76.75	69.00
		✓	86.67	93.65	78.67	85.51	42.00
Popular	LLaVA [21]	✗	76.67	72.22	86.67	78.79	60.00
		✓	80.67	83.82	76.00	79.72	45.33
	MiniGPT-4 [47]	✗	56.67	58.77	44.67	50.76	38.00
		✓	82.33	85.40	78.00	81.53	45.67
	mPLUG-Owl [40]	✗	57.33	54.20	**94.67**	68.93	87.33
		✓	83.00	84.14	81.33	82.71	48.33
	Otter [15]	✗	67.33	61.71	91.33	73.66	74.00
		✓	**84.33**	**88.15**	79.33	**83.51**	45.00
Adversarial	LLaVA [21]	✗	73.33	69.02	84.67	76.05	61.33
		✓	80.67	82.86	77.33	80.00	46.67
	MiniGPT-4 [47]	✗	55.00	56.88	41.33	47.88	36.33
		✓	82.33	83.92	80.00	81.91	47.67
	mPLUG-Owl [40]	✗	56.33	53.51	**96.67**	68.88	90.33
		✓	81.00	82.07	79.33	80.68	48.33
	Otter [15]	✗	66.67	61.16	91.33	73.26	74.67
		✓	**83.00**	**85.61**	79.33	**82.35**	46.33

Table 1. Results on POPE. w/Ours denotes MLLM responses corrected by our proposed Woodpecker. The best and second-to-best performances within each setting are **bolded** and underlined, respectively.

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 POPE. The results on POPE under the random, popular, and adversarial settings are summarized in Tab. 1. 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 the 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 [18]. 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 experiments on MME is more well-rounded since it covers not only object-level but also attribute-level hallucination evaluation.

Method	w/Ours	Object-level		Attribute-level		Total
		Existence	*Count*	*Position*	*Color*	
LLaVA [21]	✗	195.00	95.00	53.33	78.33	421.67
	✓	195.00	160.00	55.00	155.00	565.00
MiniGPT-4 [47]	✗	100.00	61.67	53.33	65.00	280.00
	✓	183.33	**163.33**	60.00	121.67	528.33
mPLUG-Owl [40]	✗	101.67	73.33	58.33	66.67	300.00
	✓	**200.00**	131.67	**78.33**	145.00	555.00
Otter [15]	✗	185.00	95.00	50.00	118.33	448.33
	✓	195.00	160.00	51.67	**165.00**	**571.67**

Table 2. Results on MME. 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 **bolded** and underlined, respectively.

The corresponding results are listed in Tab. 2. 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, 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 may not comprehend the given bounding boxes well enough to derive position relationships by itself.

Method	w/Ours	Accuracy	Detailedness
LLaVA [21]	✗	7.1	7.1
	✓	7.8	8.6
MiniGPT-4 [47]	✗	7.0	6.4
	✓	8.2	8.8
mPLUG-Owl [40]	✗	5.4	6.4
	✓	5.7	6.4
Otter [15]	✗	7.0	6.7
	✓	8.5	8.8

Table 3. Results of GPT-4V-aided evaluation. The accuracy and detailedness metrics are on a scale of 10, and a higher score indicates the better performance.

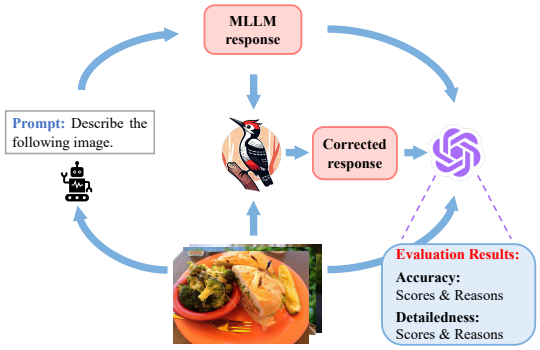

Figure 4. Illustration of GPT-4V-aided evaluation.

Results on LLaVA-QA90. Different from the above two experiments that only involve "Yes-or-No" questions, the experiment on LLaVA-QA90 is much more open. The description-type queries instruct MLLMs to fully translate the input image into language, rather than merely referring to the existence or the attribute of 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 [20, 21]. 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 Fig. 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. However, it has just opened up its web interface that only supports multimodal interaction through manual operation, and there are strict limits on the number of uses. This makes the GPT-4V-based evaluation labor-intensive, and we can only test a limited number of images, such as LLaVA-QA90. 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 scores of the two metrics are displayed in Tab. 3, 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.

4.3. Experimental Analysis

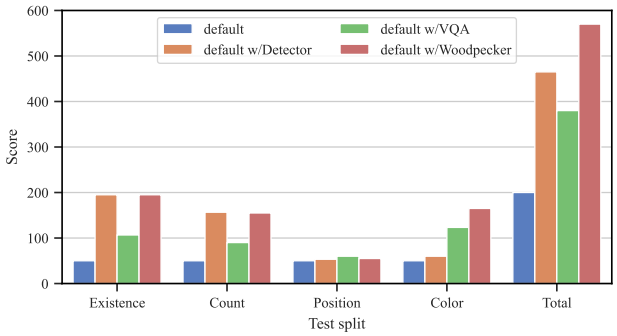

Figure 5. Results on MME with different framework variants. "default" is a model that always answer "Yes", "default w/Detector" introduces the object detector for hallucination correction, and "default w/VQA" introduces the VQA model. "default w/Woodpecker" is our full framework.

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

As shown in Fig. 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.

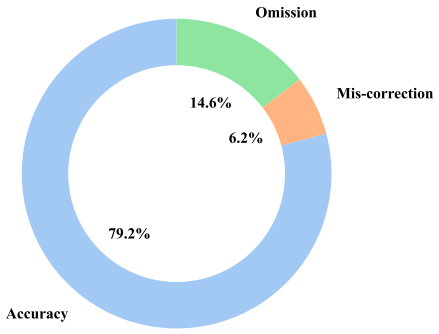

Figure 6. Proportion of different correction results.

To give an intuitive comprehension of the results of correction and the GPT-4V-aid evaluation, we offer a case 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 works in measuring the correction behavior, we fulfill this goal by breaking down the results after correction into three sections:

- 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 Fig. 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 over-confident.

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.

References

[1] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv*, 2023. 1

[2] William Berrios, Gautam Mittal, Tristan Thrush, Douwe Kiela, and Amanpreet Singh. Towards language models that can see: Computer vision through the lens of natural language. *arXiv*, 2023. 3

[3] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *ICML*, 2022. 3

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 2020. 5

[5] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv*, 2018. 3

[6] 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*, 2023. 3, 5

[7] Ran Gong, Qiuyuan Huang, Xiaojian Ma, Hoi Vo, Zane Durante, Yusuke Noda, Zilong Zheng, Song-Chun Zhu, Demetri Terzopoulos, Li Fei-Fei, et al. Mindagent: Emergent gaming interaction. *arXiv*, 2023. 3

[8] Anisha Gunjal, Jihan Yin, and Erhan Bas. Detecting and preventing hallucinations in large vision language models. *arXiv*, 2023. 3

[9] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *CVPR*, 2023. 3

[10] Kung-Hsiang Huang, Hou Pong Chan, and Heng Ji. Zero-shot faithful factual error correction. *arXiv*, 2023. 3, 5

[11] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *ICML*, 2021. 5

[12] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv*, 2023. 4

[13] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. *arXiv*, 2023. 3

[14] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *NeurIPS*, 2020. 3

[15] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning. *arXiv*, 2023. 5, 6, 7

[16] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv*, 2023. 4, 5

[17] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *CVPR*, 2022. 4

[18] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv*, 2023. 3, 5, 6

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 5

[20] Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Mitigating hallucination in large multi-modal models via robust instruction tuning. *arXiv*, 2023. 1, 3, 7

[21] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv*, 2023. 1, 3, 5, 6, 7

[22] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv*, 2023. 4, 5

[23] Jiaying Lu, Jinmeng Rao, Kezhen Chen, Xiaoyuan Guo, Yawen Zhang, Baochen Sun, Carl Yang, and Jie Yang. Evaluation and mitigation of agnosia in multimodal large language models. *arXiv*, 2023. 3

[24] Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. *arXiv*, 2023. 3

[25] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *CVPR*, 2022. 4

[26] OpenAI. Gpt-4 technical report. *arXiv*, 2023. 5

[27] Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv*, 2023. 3

[28] Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. Kilt: a benchmark for knowledge intensive language tasks. *arXiv*, 2020. 3

[29] Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Dmytro Okhonko, Samuel Broscheit, Gautier Izacard, Patrick Lewis, Barlas Oğuz, Edouard Grave, Wen-tau Yih, et al. The web is your oyster-knowledge-intensive nlp against a very large web corpus. *arXiv*, 2021. 3

[30] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface. *arXiv*, 2023. 3

[31] Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, et al. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. *arXiv*, 2022. 3

[32] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *CVPR*, 2022. 4

[33] Bin Wang, Fan Wu, Xiao Han, Jiahui Peng, Huaping Zhong, Pan Zhang, Xiaoyi Dong, Weijia Li, Wei Li, Jiaqi Wang, et al. Vigc: Visual instruction generation and correction. *arXiv*, 2023. 1, 2, 3

[34] Junyang Wang, Yiyang Zhou, Guohai Xu, Pengcheng Shi, Chenlin Zhao, Haiyang Xu, Qinghao Ye, Ming Yan, Ji Zhang, Jihua Zhu, et al. Evaluation and analysis of hallucination in large vision-language models. *arXiv*, 2023. 3

[35] Teng Wang, Jinrui Zhang, Junjie Fei, Yixiao Ge, Hao Zheng, Yunlong Tang, Zhe Li, Mingqi Gao, Shanshan Zhao, Ying Shan, et al. Caption anything: Interactive image description with diverse multimodal controls. *arXiv*, 2023. 3

[36] Zhenyu Wang, Yali Li, Xi Chen, Ser-Nam Lim, Antonio Torralba, Hengshuang Zhao, and Shengjin Wang. Detecting everything in the open world: Towards universal object detection. In *CVPR*, 2023. 4

[37] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv*, 2023. 3

[38] Rui Yang, Lin Song, Yanwei Li, Sijie Zhao, Yixiao Ge, Xiu Li, and Ying Shan. Gpt4tools: Teaching large language model to use tools via self-instruction. *arXiv*, 2023. 3

[39] Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv*, 2023. 3

[40] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv*, 2023. 1, 3, 5, 6, 7

[41] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *arXiv*, 2023. 1, 3, 5

[42] Haoxuan You, Rui Sun, Zhecan Wang, Long Chen, Gengyu Wang, Hammad A Ayyubi, Kai-Wei Chang, and Shih-Fu Chang. Idealgpt: Iteratively decomposing vision and language reasoning via large language models. *arXiv*, 2023. 3

[43] Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, et al. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv*, 2022. 3

[44] Ao Zhang, Hao Fei, Yuan Yao, Wei Ji, Li Li, Zhiyuan Liu, and Tat-Seng Chua. Transfer visual prompt generator across llms. *arXiv*, 2023. 1

[45] Renrui Zhang, Xiangfei Hu, Bohao Li, Siyuan Huang, Hanqiu Deng, Yu Qiao, Peng Gao, and Hongsheng Li. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. In *CVPR*, 2023. 3

[46] Deyao Zhu, Jun Chen, Kilichbek Haydarov, Xiaoqian Shen, Wenxuan Zhang, and Mohamed Elhoseiny. Chatgpt asks, blip-2 answers: Automatic questioning towards enriched visual descriptions. *arXiv*, 2023. 3

[47] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv*, 2023. 1, 3, 5, 6, 7

[48] Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyu Guo, Ziyao Zeng, Zipeng Qin, Shanghang Zhang, and Peng Gao. Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In *ICCV*, 2023. 3

A. Prompt Templates

In this part, we list our prompt templates for instructing LLM to fulfill various tasks, including key concept extraction, question formulation, hallucination correction, and GPT-4V-aided evaluation.

A.1. Key Concept Extraction

The template is listed in Tab. 4.

A.2. Question Formulation

The template is listed in Tab. 5.

A.3. Hallucination Correction

The template is listed in Tab. 6.

A.4. GPT-4V-aided Evaluation

The template is listed in Tab. 7.

B. GPT-4V-aided Evaluation Case

To offer a straightforward and intuitive understanding, we list an evaluation case in Fig. 7, where "Assistant 1" and "Assistant 2" in the evaluation reason part correspond to "MLLM" and "MLLM w/Woodpecker", respectively. GPT-4V gives not only respective scores for responses but also reasons for the judgment.

System message

You are a language assistant that helps to extract information from given sentences.

Prompt

Given a sentence, extract the existent entities within the sentence for me.
Extract the common objects and summarize them as general categories without repetition, merge essentially similar objects.
Avoid extracting abstract or non-specific entities. Only extract concrete, certainly existent objects that fall in general categories and are described in a certain tone in the sentence.
Extract entity in the singular form. Output all the extracted types of items in one line and separate each object type with a period. If there is nothing to output, then output a single "None".

Examples:
{In-context examples}

Sentence:
{Input sentence}

Output:

Table 4. Template for prompting LLM to perform key concept extraction. {In-context examples} are in-context examples used to better instruct the LLM to fulfill the task, and {Input sentence} is the input from which the key concept is extracted.

System message
You are a language assistant that helps to ask questions about a sentence.

Prompt
Given a sentence, extract the existent entities within the sentence for me.
Given a sentence and some entities connected by periods, you are required to ask some relevant questions about the specified entities involved in the sentence, so that the questions can help to verify the factuality of the sentence.
Questions may involve basic attributes such as colors and actions mentioned in the sentence. Do not ask questions involving object counts or the existence of objects.
When asking questions about attributes, try to ask simple questions that only involve one entity.
Ask questions that can be easily decided visually. Do not ask questions that require complex reasoning.
Do not ask semantically similar questions. Do not ask questions only about scenes or places.
Use "where" type questions to query the position information of the involved entities.
Do not ask questions about uncertain or conjecture parts of the sentence, for example, the parts described with "maybe" or "likely", etc.
It is no need to cover all the specified entities. If there is no question to ask, simply output a "None".
When asking questions, do not assume the claims in the description as true in advance. Only ask questions relevant to the information in the sentence.
Only ask questions about common, specific, and concrete entities. The entities involved in the questions are limited to the range within the given entities.
Output only one question in each line. For each line, first output the question, then a single "&", and finally entities involved in the question, still connected by periods if multiple entities are involved.

Examples:
{In-context examples}

Sentence:
{Input sentence}

Entities:
{Input entities}

Questions:

Table 5. Prompt template for question formulation. {In-context examples} are in-context examples. {Input sentence} and {Input entities} are the inputs, where the latter comes from the step of key concept extraction.

System message

You are a language assistant that helps to refine a passage according to instructions.

Prompt

Given a passage and some supplementary information, you are required to correct and output the refined passage in a fluent and natural style, following these rules:

1. The supplementary information may include some of the following parts:

"Count" information that specifies how many instances of a certain kind of entity exist, and their associated bounding boxes;

"Specific" information that describes attribute information specific to each entity instance, including bounding boxes, colors, etc. The information is arranged in the form of "entity 1: [bbox]" info of this entity. Note that the entity in "Specific" information corresponds to that in the "Count" information.

"Overall" information that may involve information about multiple entity objects.

2. Try to retain the original sentence with minimal changes.

3. The number of entitie instances should match the number in the "Count" information. Also correct the number counts if the number stated in the original sentence does not match the counting information.

4. If the original sentence is already correct, then just keep it. If you need to rewrite the original sentence, when rewriting, try to modify the original sentence as little as possible based on the original sentence, and use the supplementary information as guidance to correct or enrich the original sentence.

5. In the refined passage, when describing entities mentioned in the "Specific" supplementary information, add their associated bounding boxes in parentheses right after them, in the form of "entity([bbox])". If multiple entities of the same kind are mentioned, then separate the box with ";", in the form of "entity([bbox1];[bbox2])"

Examples:
{In-context examples}

Supplementary information:
{Input information}

Passage:
{Input passage}

Refined passage:

Table 6. Prompt template for hallucination correction. {In-context examples} are in-context examples. {Input information} is the formatted knowledge base, and {Input passage} is the original response to be corrected.

Prompt

You are required to score the performance of two AI assistants in describing a given image. You should pay extra attention to the hallucination, which refers to the part of descriptions that are inconsistent with the image content, such as claiming the existence of something not present in the image or describing incorrectly in terms of the counts, positions, or colors of objects in the image. Note that the descriptions may be accompanied by bounding boxes, indicating the position of objects in the image, which are represented as [x1, y1, x2, y2] with floating numbers ranging from 0 to 1. These values correspond to the top left x1, top left y1, bottom right x2, and bottom right y2.
Please rate the responses of the assistants on a scale of 1 to 10, where a higher score indicates better performance, according to the following criteria:
1: Accuracy: whether the response is accurate with respect to the image content. Responses with fewer hallucinations should be given higher scores.
2: Detailedness: whether the response is rich in necessary details. Note that hallucinated descriptions should not count as necessary details.
Please output a single line for each criterion, containing only two values indicating the scores for Assistant 1 and 2, respectively. The two scores are separated by a space. Following the scores, please provide an explanation of your evaluation, avoiding any potential bias and ensuring that the order in which the responses were presented does not affect your judgment.

[Assistant 1]
{Response 1}

[End of Assistant 1]

[Assistant 2]
{Response 2}

[End of Assistant 2]

Output format:

Accuracy:
Scores of the two answers:
Reason:

Detailedness:
Scores of the two answers:
Reason:

Table 7. Prompt template for GPT-4V-aided evaluation. {Response 1} and {Response 2} are the original responses and the corrected ones, respectively.

GPT-4V-aided Evaluation

MLLM:

An elephant is walking on a dirt ground with a blue bag and a tire nearby.

MLLM w/Woodpecker:

In this snapshot, an elephant([0.306, 0.4, 0.636, 0.783]) is walking on a dirt ground. There is a blue bag([0.319, 0.798, 0.449, 0.919]) nearby the elephant, and there are two tires([0.634, 0.797, 0.795, 0.873];[0.0, 0.874, 0.21, 0.987]) nearby as well.

GPT Score:

Accuracy:
8 10
Reason:
Both assistants have accurately described the presence of an elephant on a dirt ground with a blue bag and tires nearby. However, Assistant 1 mentioned a single tire, while the image shows two tires. Assistant 2 correctly identified the number of tires and their positions.

Detailedness:
6 9
Reason:
While both descriptions provide necessary details about the elephant, blue bag, and tires, Assistant 2 goes a step further by providing bounding boxes for the elephant, blue bag, and both tires, offering a more detailed description of the image content. Assistant 1's description is less detailed in comparison.

Figure 7. Example for the GPT-4V-aided evaluation.