

# Logical Closed Loop: Uncovering Object Hallucinations in Large Vision-Language Models

Junfei Wu<sup>1,2</sup>, Qiang Liu<sup>1,2</sup>, Ding Wang<sup>1,2</sup>, Jinghao Zhang<sup>1,2</sup>, Shu Wu<sup>1,2\*</sup>, Liang Wang<sup>1,2</sup>, Tieniu Tan<sup>1,2,3</sup>

<sup>1</sup> New Laboratory of Pattern Recognition (NLPR),

State Key Laboratory of Multimodal Artificial Intelligence Systems,  
Institute of Automation, Chinese Academy of Sciences

<sup>2</sup>School of Artificial Intelligence, University of Chinese Academy of Sciences

<sup>3</sup>Nanjing University

junfei.wu@cripac.ia.ac.cn, qiang.liu@nlpr.ia.ac.cn, wangding2024@ia.ac.cn  
jinghao.zhang@cripac.ia.ac.cn, {shu.wu, wangliang, tnt}@nlpr.ia.ac.cn

## Abstract

Object hallucination has been an Achilles' heel which hinders the broader applications of large vision-language models (LVLMs). Object hallucination refers to the phenomenon that the LVLMs claim non-existent objects in the image. To mitigate the object hallucinations, instruction tuning and external model-based detection methods have been proposed, which either require large-scale computational resources or depend on the detection result of external models. However, there remains an under-explored field to utilize the LVLM itself to alleviate object hallucinations. In this work, we adopt the intuition that the LVLM tends to respond logically consistently for existent objects but inconsistently for hallucinated objects. Therefore, we propose a Logical Closed Loop-based framework for Object Hallucination Detection and Mitigation, namely **LogicCheckGPT**. In specific, we devise logical consistency probing to raise questions with logical correlations, inquiring about attributes from objects and vice versa. Whether their responses can form a logical closed loop serves as an indicator of object hallucination. As a plug-and-play method, it can be seamlessly applied to all existing LVLMs. Comprehensive experiments conducted on three benchmarks across four LVLMs have demonstrated significant improvements brought by our method, indicating its effectiveness and generality<sup>1</sup>.

## 1 Introduction

With the great advancement of large language models (LLMs) (Ouyang et al., 2022; Touvron et al., 2023; Zhao et al., 2023), they have showcased impressive abilities, such as text generation, instruction following. Recent studies have been devoted to introduce the powerful capabilities of LLMs to the field of multimodal models. Empowered by LLMs,

\*To whom correspondence should be addressed.

<sup>1</sup>The source code is publicly available at: <https://github.com/Hyperwjf/LogicCheckGPT>.

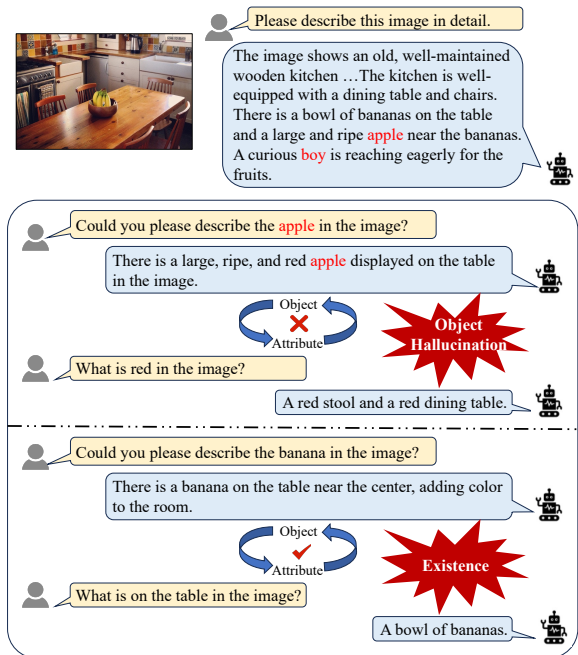


Figure 1: An example of object hallucinations. Hallucinated objects are highlighted in red. The LVLM shows different logical consistency to hallucinated object “apple” and existent object “banana”.

large vision-language models (LVLMs) (Liu et al., 2023; Ye et al., 2023; Zhu et al., 2023; Li et al., 2023; Bai et al., 2023; Dai et al., 2023b) are facilitated to perform strong multimodal understanding and reasoning.

Despite the exciting breakthrough in LVLMs, they all suffer from hallucination issues inevitably, particularly object hallucination. Object hallucination refers to the phenomenon that the LVLMs generate inconsistent descriptions of the given image, for example, making up non-existent objects. Taking Fig. 1 as an example, the LVLM hallucinates several objects which are not existent in the image, including an apple and a boy. This issue hinders more widespread application of LVLMs. In safety-related scenarios, the consequences of

hallucinations would be unbearable.

There have been several efforts made to alleviate hallucinations in LVLMs. Generally, these methods can be categorized into three groups. The first as well as the most popular approach (Liu et al., 2024; Lee et al., 2023) mainly resorts to instruction tuning or specific retraining to facilitate LVLMs to generate less hallucinated contents. The second approach (Yin et al., 2023; Zhou et al., 2024) incorporates external detection models or specific LVLMs to enhance visual understanding, thereby refining the outputs of the original LVLMs. The last approach (Huang et al., 2023; Leng et al., 2023) investigates the decoding process of LVLMs and devise novel decoding strategies to avoid hallucinations. Although these methods have achieved some effectiveness in alleviating hallucinations, there are still some drawbacks: requiring significant computational resources, depending on external models, or necessitating access to the internal parameters of the model.

However, we argue that the logical consistency of LVLM behaviors have the potential to elucidate the underlying hallucination it encapsulates. As illustrated in Fig. 1, we first obtain the attributes “red” for “apple” and “on the table” for “banana” by inquiring the LVLM. Subsequently, when we inquire about which object possesses these attributes, the LVLM can correctly respond with “banana” but fails to answer for the hallucinated object “apple”. It demonstrates that when we pose a series of logically connected questions about a particular object, the LVLM exhibits better logical consistency for existing objects, while its performance tends to degrade for hallucinated objects. It is reasonable because the described attributes of hallucinated objects primarily originate from two sources: attributes from other objects in the image, or fabricated attributes absent in the image. Consequently, the model may fail to answer the hallucinated object when we question what possesses these attributes.

Inspired by this observation, we propose a novel and effective framework called Logic Closed Loop for Object Hallucination Detection and Mitigation, namely **LogicCheckGPT**, which is training-free and only requires language interaction. Our aim is to formulate two types of questions in two stages: the first stage involves inquiring attributes based on objects, followed by inquiring objects based on attributes. Whether their responses can form a logical closed loop serves as an indicator of object

hallucination.

In specific, according to the different stages of questioning, we divide our framework into 5 steps: (1) *Object extraction* extracts objects in the responses of LVLMs. (2) *Object-to-Attribute inquiring* inquiries into the detailed attributes of the target objects. (3) *Attribute-to-Object inquiring* further formulates follow-up questions to inquire what object possesses the attributes mentioned in previous answers. (4) *Logic closed loop check* examines whether the logical relationships from objects to attributes and attributes to objects can form a closed loop. (5) *Hallucination detection and mitigation* rectifies hallucinated objects if the ratio of closed loops to the total number of questions exceeds a certain threshold. Our method is a plug-and-play approach that can be applied to various LVLMs without training or relying on external detection models. Furthermore, the question-answer process in natural language enhances its interpretability.

We evaluated the effectiveness of our framework across multiple advanced LVLMs on several benchmarks (Yifan et al., 2023; Fu et al., 2023), as well as GPT-4v assisted evaluation (Liu et al., 2024; Yin et al., 2023). Our method demonstrates significant improvements across state-of-the-art LVLMs, including a 31.33%/10.00% improvement on the POPE dataset for mPLUG-Owl (Ye et al., 2023)/MiniGPT-4 (Zhu et al., 2023).

Overall, our main contributions can be summarized as follows:

- We are the first to adopt the logical closed loop in the context of object hallucination alleviation in LVLMs.
- We propose a novel framework LogicCheckGPT for detecting and mitigating object hallucinations in LVLMs, which is training-free and offers language interaction for user-friendly interpretation.
- Comprehensive experiments are conducted to validate the effectiveness of our method, where the results demonstrate the superiority and universality.

## 2 Related Work

### 2.1 Large Vision-Language Models

With the surge in the capabilities of large language models (LLMs) (Ouyang et al., 2022; Zhao et al., 2023; Brown et al., 2020), there is currently

a growing interest in how to integrate the general artificial intelligence of LLMs into the multimodal domains. In consequence, large vision-language models (LVLMs) powered by LLMs are proposed (Ye et al., 2023; Zhu et al., 2023; Liu et al., 2023; Li et al., 2023; Dai et al., 2023a; Bai et al., 2023), which can understand multimodal contents and perform multimodal tasks under instructions. In general, existing LVLMs follow the following paradigm: leveraging a multimodal alignment module to comprehend multimodal inputs, followed by utilizing a LLM to generate responses. Therefore, the training process of LVLMs typically involves modalities alignment pre-training and instruction tuning. Specifically, mPLUG-Owl (Ye et al., 2023) pre-trains the encoder and alignment module, and then finetunes LLaMa (Touvron et al., 2023) by low-rank adaption. In contrast, LLaVA (Liu et al., 2023) only pre-trains the alignment network and finetunes the alignment network and Vicuna (Chiang et al., 2023) on constructed instructions. MiniGPT-4 (Zhu et al., 2023) only finetunes the cross-modal alignment network with other modules frozen.

## 2.2 Hallucination in LVLMs

Despite the strong capabilities of these LVLMs, they all grapple with hallucination issues unexpectedly. To tackle with this issue, several benchmarks (Fu et al., 2023; Xu et al., 2023; Yifan et al., 2023; Lovenia et al., 2023; Jing et al., 2023; Chen et al., 2024) have been proposed to provide detailed evaluations of the hallucination degree exhibited by LVLMs.

Existing hallucination mitigation strategies for LVLMs can be roughly divided into three groups. The first and most widely adopted approach (Liu et al., 2024; Gunjal et al., 2023; Lee et al., 2023; Wang et al., 2023) primarily relies on instruction tuning and retraining. LRV-Instruction (Liu et al., 2024) introduces a comprehensive instruction tuning dataset encompassing positive and negative instructions. (Wang et al., 2023) adopts an iterative instruction generation strategy to improve diversity and accuracy of instructions. Volcano (Lee et al., 2023) facilitates the model with the ability to utilize self-feedback to self-revise responses through training. However, these methods heavily depend on the quality of instruction data construction and require substantial computational resources.

The second group of approaches, exemplified by Woodpecker (Yin et al., 2023) and LURE (Zhou

et al., 2024), aim to integrate external detection models or specific LVLMs as revisors to enhance accurate visual understanding, thereby refining base LVLMs’ hallucinated generation. Nevertheless, these approaches rely on external models and fail to explore the intrinsic capabilities of the base model.

The third group of approaches aim to devise decoding strategies to mitigate hallucinations during the process of decoding. OPERA (Huang et al., 2023) propose a penalty-based decoding method along with roll-back strategy to avoid over-trust during decoding. On the other hand, VCD (Leng et al., 2023) introduces contrastive decoding to reduce over-reliance on spurious bias and learned priors. However, obtaining internal states of LVLMs during the decoding process poses a challenge for common users.

Compared to existing approaches, our proposed method is training-free and mitigates hallucinations solely through language interactions. It not only explores the potential of LVLMs to alleviate hallucinations but also offers better interpretability.

## 2.3 Consistency Checking for Hallucination Detection

There have also been some works on hallucination detection in LLMs (Manakul et al., 2023; Kuhn et al., 2022; Lin et al., 2023), which view the consistency of responses reflects the model’s uncertainty. (Kuhn et al., 2022) propose semantic entropy to measure the degree of semantic divergence among responses to accommodate semantic equivalence in free-form text. SelfCheckGPT (Manakul et al., 2023) extend the method to black-box LLMs, eliminating the need for tokens’ probability. (Lin et al., 2023) introduce and compare various uncertainty estimation metrics for black-box LLMs. In contrast to prior works that focus on consistency among responses to the same question, we propose LogicCheckGPT, a logic consistency-based method that involves logic-related questions and answers. LogicCheckGPT is more capable of delving deeper into the internal uncertainty and the degree of hallucination within the LVLMs.

## 3 Method

In this section, we first introduce the overall framework of LogicCheckGPT, and then elaborate each component. Our framework is shown in Fig. 2.

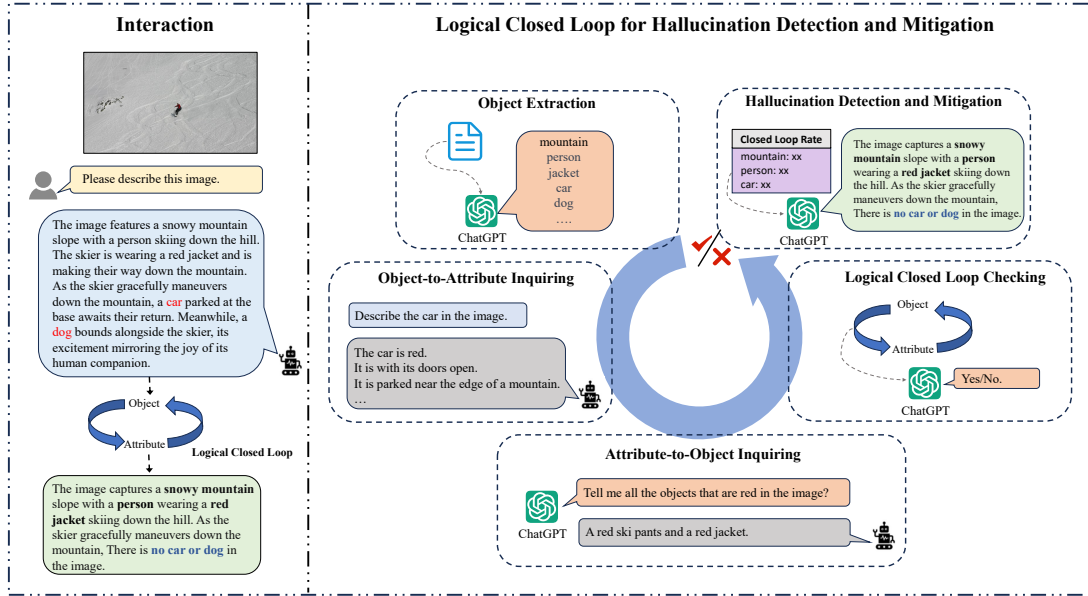


Figure 2: The proposed framework LogicCheckGPT. For LVLm responses to multimodal instructions, LogicCheckGPT employs the following five steps to alleviate object hallucinations: object extraction, object-to-attribute inquiring, attribute-to-object inquiring, logical close loop checking, and hallucination detection and mitigation.

### 3.1 Overall

To alleviate object hallucinations, we delve into the logical consistency of LVLm’s responses. Specifically, we examine whether the responses demonstrate logical coherence. For each object mentioned in a response, we pose two types of sequential questions: one regarding the attributes of the object, and another about which object possesses those attributes. The logical consistency of the LVLm’s responses, i.e., whether they form a closed loop of logical reasoning, serves as an indicator of object hallucination. Here, the “logical closed loop” refers to the subsequent object answered being consistent with the initial object.

This process is decomposed into the following steps: object extraction, object-to-attribute inquiring, attribute-to-object inquiring, logical closed loop checking, and hallucination detection and mitigation.

### 3.2 Object Extraction

To determine the object hallucination, we first need to extract candidate objects from the responses of LVLms for further querying and checking. For simplicity and versatility, we adopt an LLM to complete each sub-task, including object extraction. Specifically, we employ GPT-3.5<sup>2</sup> as our LLM because of its strong capabilities. The prompt we used

<sup>2</sup><https://platform.openai.com/docs/models/gpt-3-5-turbo>

can be referred in Appendix D.1. The extracted objects are represented as  $O = \{o_1, \dots, o_i, \dots, o_m\}$ , where  $m$  is the total number of examinee objects.

### 3.3 Object-to-Attribute Inquiring

As discussed in Section 3.1, the primary step involves constructing object-to-attribute questions to inquire about the attributes of the object. However, it is impractical to enumerate all possible attributes, given their infinite nature. Additionally, it is challenging to create generic attribute question templates applicable to all objects, as different objects typically possess their own specific attributes. For example, attributes related to material for a “dining table” differ from those related to clothing for a “person”.

For better flexibility and adaptability, we prompt the LVLm to provide a detailed description of the object  $o_i$  in free-form text. The template question is formulated as follows: “Could you please describe the {object} in the image?” We ask the LVLm to respond multiple times. This approach allows us to obtain detailed and specific attribute descriptions from the LVLm regarding the object, thereby facilitating the construction of attribute-to-object questions in subsequent steps. The average number of extracted attributes of existent or hallucinated objects has also been studied in Section C.



### 3.4 Attribute-to-Object Inquiring

In the attribute-to-object inquiring stage, our goal is to formulate questions from the attributes to the object, in contrast to Section 3.3. After obtaining the attribute descriptions of the object, follow-up questions can be raised based on them. Rather than directly prompting the LLM to formulate questions from the descriptions, we break down this task into two subtasks: attribute extraction and question formulation. This approach avoids potential issues of misleading the LLM to not follow instructions and inadvertently revealing object identity in the questions. It has also been observed that decomposing a task into several simple sub-tasks for LLMs to fulfill yields better performance (Wei et al., 2022; Zhao et al., 2023).

In specific, we prompt the LLM to extract attributes of the target object  $o_i$  from the description, wherein the target object is represented as “The object”. For instance, extracted attributes such as “The object is made of wood in the image”, “The object is red in color”. For the examinee object  $o_i$ , the extracted attributes are represented as  $A_i = \{a_{i,1}, \dots, a_{i,j}, \dots, a_{i,n_i}\}$ , where the number of extracted attributes is denoted as  $n_i$ . Detailed prompting instructions can be found in Appendix D.2.1.

Subsequently, we instruct the LLM to convert these extracted attributes into questions that inquire about what object possesses the specific attributes. However, we have noticed that asking questions like “What is/has {attribute} in the image?” often yield answers about the most obvious objects with the target attributes, potentially leading to the omission of other less conspicuous yet existent objects. To deal with this issue, we frame our questions  $Q_i = \{q_{i,1}, \dots, q_{i,j}, \dots, q_{i,n_i}\}$  in the format “Could you tell me all the objects that {attribute} in the image?”. The ablation study of this prompt design can be referred in Section 4.3. This format enables the LVLM to comprehensively cover objects that meet the specified attribute constraints. Detailed prompting instructions can be found in Appendix D.2.2.

### 3.5 Logical Closed Loop Checking

After obtaining the answers  $R_i = \{r_{i,1}, \dots, r_{i,j}, \dots, r_{i,n_i}\}$  from the attribute-to-object inquiring stage, we can assess whether each answer forms a logical closed loop, meaning that the object mentioned in the answer is consistent

with the examinee object  $o_i$ . In specific, we prompt the LLM to check whether the  $o_i$  is covered in each LVLM’s response  $r_{i,j}$ . The judgement is limited to “Yes” and “No”. For the  $i$ -th examinee object, the judgement of the  $j$ -th answer is mapped into score  $x_{i,j}$  through the mapping {Yes: 1.0, No: 0.0}. The prompt is listed in Appendix D.3.

### 3.6 Hallucination Detection and Mitigation

Finally, the logical closed loop rate for each examinee object, defined as the number of logical closed loops divided by the total number of attribute-to-object question-answer pairs  $n_i$ , can be formulated as

$$\mathcal{S}(o_i) = \frac{1}{n_i} \sum_j^{n_i} x_{i,j} \quad (1)$$

where  $\mathcal{S}(o_i)$  denotes the logical closed loop rate for the  $i$ -th object.

As discussed before, the object is likely to be hallucinated when  $\mathcal{S}(o_i)$  is low. On the other hand, when it’s close to 1, it’s more likely that the object truly exists. Therefore,  $\mathcal{S}(o_i)$  serves as an indicator of object hallucination. Through a valid hallucination threshold  $\lambda$ , our method can effectively detect objects that are likely to be hallucinated, i.e.,  $\mathcal{S}(o_i)$  is below  $\lambda$ .

After identifying hallucinated objects, we guide the LLM to eliminate contents related to hallucinated objects, aiming to minimize hallucinations. The detailed prompt can be found in Appendix D.4.

## 4 Experiment

### 4.1 Experimental Setup

#### 4.1.1 Dataset

**POPE** (Yifan et al., 2023) is proposed to provide a detailed evaluation of object hallucination in LVLMs, by querying the models about the presence of specific objects in given images. POPE adopts three sampling settings to construct negative samples: random, popular, and adversarial. The random setting selects non-present objects at random, whereas the popular setting chooses from a list of frequently occurring but absent objects, and the adversarial method selects based on common co-occurrence in contexts despite the absence in the target image. For each sampling setting, we sample 50 images and 6 questions for each image, with an even distribution of positive and negative samples (50% - 50%). Accuracy and F1-score are employed as the evaluation metrics.

Model	Method	Adversarial		Popular		Random	
		Acc	F1	Acc	F1	Acc	F1
mPLUG-Owl	vanilla	50.67	66.81	51.67	67.26	55.33	68.98
	LRV-Instruction	59.67	69.21	68.33	74.11	74.33	77.94
	SelfCheck	66.67	74.09	72.00	77.29	70.66	75.82
	LURE	76.33	76.72	79.67	78.75	81.33	80.95
	LogicCheckGPT	<b>82.00</b>	<b>82.23</b>	<b>84.66</b>	<b>84.45</b>	<b>91.00</b>	<b>90.84</b>
MiniGPT-4	vanilla	72.67	75.88	78.33	79.87	84.33	84.59
	LRV-Instruction	74.00	71.11	80.33	78.70	81.67	80.97
	SelfCheck	73.00	72.72	76.67	75.86	76.00	73.53
	LURE	77.67	79.14	80.67	80.67	83.67	84.14
	LogicCheckGPT	<b>82.67</b>	<b>80.59</b>	<b>83.67</b>	<b>81.51</b>	<b>86.67</b>	<b>85.29</b>
LLaVA-1.5	vanilla	83.33	84.84	84.67	85.89	93.00	<b>93.02</b>
	SelfCheck	88.67	88.27	88.67	88.59	90.33	89.53
	LURE	85.33	86.25	87.00	87.05	89.67	89.70
	LogicCheckGPT	<b>90.00</b>	<b>89.58</b>	<b>91.67</b>	<b>91.40</b>	<b>93.33</b>	93.00
QWEN-VL-Chat	vanilla	86.67	86.67	86.33	86.37	90.67	90.28
	SelfCheck	87.67	87.54	87.67	87.87	91.13	<b>91.45</b>
	LURE	87.00	87.62	87.33	87.16	88.67	88.28
	LogicCheckGPT	<b>89.00</b>	<b>88.00</b>	<b>89.67</b>	<b>88.64</b>	<b>91.33</b>	90.71

Table 1: The performance comparison between our proposed method LogicCheckGPT and baselines on POPE. The best result is highlighted in boldface.

**MME** (Fu et al., 2023) serves as a comprehensive benchmark for evaluating the perceptual and cognitive capabilities of LVLMs across a wide spectrum of tasks. For the purpose of this study, we only utilize the *existence* subset to evaluate the phenomenon of object-level hallucination within these models. This approach mirrors the methodology employed in the POPE framework, wherein each subset consists of binary “Yes-or-No” questions. We use accuracy and accuracy+ as metrics, where the former is calculated based on each question, while the latter is based on each image, requiring both questions to be answered correctly.

**GPT-4v Assisted Evaluation** To assess the effectiveness of our method for hallucination mitigation in open-ended generation, we adopted the GPT-4v Assisted Evaluation, inspired by Yin et al. (2023) and Liu et al. (2024). Our evaluation samples a set of 50 images from the COCO 2014 validation dataset, and asks the model to generate detailed descriptions. Subsequently, we prompt GPT-4v to score original outputs and our outputs in accuracy and relevancy, based on the image and instruction. The detailed prompt example is shown in Appendix

D.5.

#### 4.1.2 Baselines

We selected several widely used open-source LVLMs as backbones to evaluate the effectiveness of our LogicCheckGPT, including mPLUG-Owl (mplug-owl-llama-7b) (Ye et al., 2023), LLaVA (llava-1.5-7b) (Liu et al., 2023), MiniGPT-4 (vicuna-13b) (Zhu et al., 2023), QWEN-VL-Chat (Bai et al., 2023).

We also compare our method with advanced hallucination detection and mitigation methods, including LRV-Instruction (Liu et al., 2024), LURE (Zhou et al., 2024) and SelfCheckGPT (Manakul et al., 2023). IRV-Instruction constructs a comprehensive instruction dataset for instruction tuning. However, as they only released the checkpoints of fine-tuned mPLUG-Owl and MiniGPT-4, we report the results of these two models. LURE trains a LVM revisor to post-hoc rectify the hallucinated outputs of base models. SelfCheckGPT employs semantic uncertainty to detect hallucinations. We integrate it into our framework to fulfill hallucination mitigation. In addition, the base LVLMs are referred to as vanilla.

Model	Method	Acc	Acc+
mPLUG-Owl	vanilla	65.00	35.00
	LRV-Instruction	83.33	66.67
	SelfCheck	85.00	73.33
	LURE	80.00	60.00
	LogicCheckGPT	<b>96.67</b>	<b>93.33</b>
MiniGPT-4	vanilla	78.33	56.67
	LRV-Instruction	83.33	66.67
	SelfCheck	80.00	60.00
	LURE	85.00	70.00
	LogicCheckGPT	<b>86.67</b>	<b>73.33</b>
LLaVA-1.5	vanilla	96.67	93.33
	SelfCheck	96.67	93.33
	LURE	93.33	86.67
	LogicCheckGPT	96.67	93.33
	QWEN-VL-Chat	vanilla	88.33
SelfCheck		93.33	86.67
LURE		90.00	80.00
LogicCheckGPT		<b>95.00</b>	<b>90.00</b>

Table 2: The performance comparison between our proposed method LogicCheckGPT and baselines on MME Existence subset. The best result on each dataset is highlighted in boldface.

## 4.2 Experimental Results

**Results on POPE** The overall performance of our proposed method LogicCheckGPT on POPE is shown in Table 1, from which we have the following observations:

Firstly, our LogicCheckGPT consistently demonstrates significant performance improvements across various LVLMS under different settings. It can be observed that there is a significant performance decline of LVLMS as we transition from random to popular and adversarial settings, indicating that LVLMS tend to hallucinate non-existent objects that are related to the input image. In specific, mPLUG-Owl only achieved accuracies of 50.67%, 51.66%, and 55.33% across the three settings. For MiniGPT-4, its performance in adversarial settings is over 10% lower compared to that in random settings. However, with the help of LogicCheckGPT, all LVLMS outperform the vanilla ones significantly under all settings. More specifically, mPLUG-Owl equipped with LogicCheckGPT achieves an accuracy enhancement exceeding 30% across three settings. Despite the already promising performance of LLaVA-1.5, our method still brings a substantial improvement to it, achieving about a 6.67% increase in accuracy. This demonstrates the effectiveness and robustness of our approach.

Model	Method	Acc	Rel
mPLUG-Owl	vanilla	3.44	<b>8.78</b>
	LogicCheckGPT	<b>4.32</b>	8.74
MiniGPT-4	vanilla	5.00	7.96
	LogicCheckGPT	<b>6.02</b>	<b>8.38</b>
LLaVA-1.5	vanilla	5.22	7.24
	LogicCheckGPT	<b>6.50</b>	<b>7.64</b>
QWEN-VL-Chat	vanilla	8.36	9.96
	LogicCheckGPT	<b>8.58</b>	9.96

Table 3: The performance of our proposed method LogicCheckGPT over base LVLMS on GPT-4v assisted evaluation. The best result on each dataset is highlighted in boldface.

Secondly, our proposed LogicCheckGPT outperforms other competitors by a significant margin. While LRV-Instruction facilitates robust instruction tuning of LVLMS, it remains challenging to completely eliminate potential hallucination issues within the model. Due to that SelfCheckGPT relies on semantic consistency in model responses, it is difficult to detect hallucinated objects when the LVLMS is overconfident. LURE demonstrates significant improvements on models like mPLUG-Owl and MiniGPT-4, but shows only marginal enhancements on more powerful models such as LLaVA-1.5 and QWEN-VL-Chat. We attribute this to the fact that LURE is built upon MiniGPT-4, thus inheriting the limitations of the underlying model. By contrast, our LogicCheckGPT yields the most substantial improvements across all LVLMS, indicating the superiority of our method.

**Results on MME Existence Subset** As shown in Table 2, we also evaluate our method on MME existence subset, which focuses on the object existence hallucination. It can be observed that, although mPLUG-Owl performs worst, our method results in substantial increases in both accuracy and accuracy+, by 31.67% and 58.33%, respectively, achieving remarkably high levels of performance. It also demonstrates mPLUG-Owl inherently contains information pertaining to object hallucinations, which can be unearthed by our method. In addition, LogicCheckGPT consistently brings significant improvement for MiniGPT-4 and QWEN-VL-Chat, while maintaining the strong performance of LLaVA-1.5.

**Results on GPT-4v Assisted Evaluation** Apart from “Yes” or “No” questions in POPE and MME,

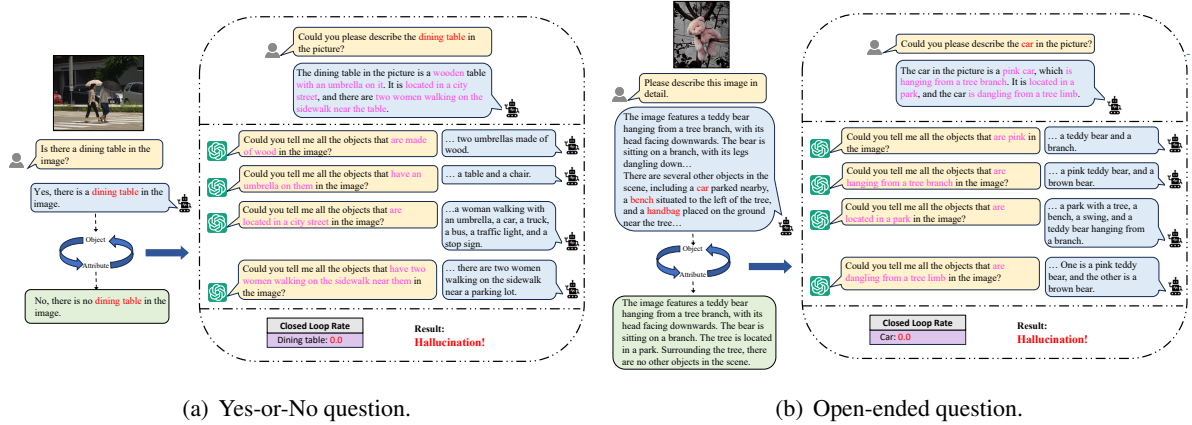


Figure 3: The visualization of two representative examples of our LogicCheckGPT for mPLUG-Owl. The hallucinated objects are highlighted in red and attributes are highlighted in magenta.

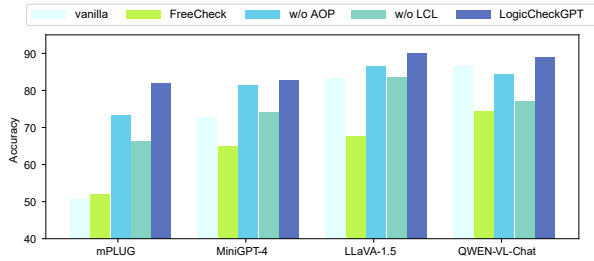


Figure 4: The performance comparison between LogicCheckGPT and several variants, vanilla, FreeCheck, LogicCheckGPT w/o AOP (w/o AOP) and LogicCheckGPT w/o LCL (w/o LCL) across LVLMS on POPE adversarial setting.

we employed GPT-4v to assist in evaluating our LogicCheckGPT for open-text generation, following previous works (Liu et al., 2024; Yin et al., 2023). The results are summarized in Table 3. It can be observed that our method significantly enhances the accuracy of each model, demonstrating the effectiveness of our approach in object hallucination mitigation. In addition, as LogicCheckGPT can remove irrelevant hallucinated information and preserving fluent language structures, it can maintain or even improve relevancy.

### 4.3 Ablation Study

We conduct ablation study for several variants, as illustrated in Fig. 4.

- *vanilla* adopts no hallucination mitigation method.
- *FreeCheck* employs an LLM (GPT-3.5) to autonomously interrogate LVLMS through multi-turn interactions, similar to (Cohen et al.,

2023). This iterative process continues until either the LLM reaches a conclusion or the predefined iteration limit (set at 5 here) is reached. The prompt can be referred in Appendix E.

- *LogicCheckGPT w/o AOP* refers to replacing the Attribute-to-Object prompt in Section 3.4 by inquire only object instead of covering all objects, e.g. “What is/has {attribute} in the image?”.
- *LogicCheckGPT w/o LCL* employs an LLM to determine the logical consistency directly without Logical Closed Loop rate.

We can observe that *FreeCheck* performs worse for most LVLMS, primarily because the LLM fails to raise valuable questions to detect hallucinations. For the existent object, *FreeCheck* often raises questions about other objects that have been mentioned by the LVLMS during the interactions and cannot focus on the examinee object. For the hallucinated object, the questions raised are often simplistic, making it easy for *FreeCheck* to be misled and fail to detect object hallucination. *LogicCheckGPT w/o AOP* brings significant improvements to several models but still falls below our method, indicating that covering a sufficient number of objects is beneficial. Though *LogicCheckGPT w/o LCL* falls between vanilla and w/o AOP, demonstrating the effectiveness of calculating logical closed loop rate. For the powerful QWEN-VL-Chat, which has already achieved impressive performance, only our LogicCheckGPT can enhance its capabilities.



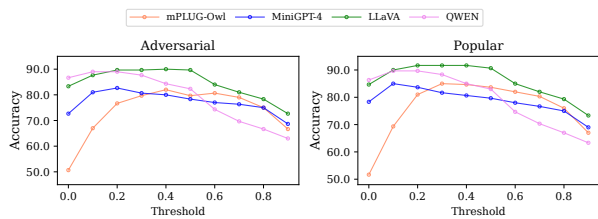


Figure 5: The performance of different threshold  $\lambda$ .

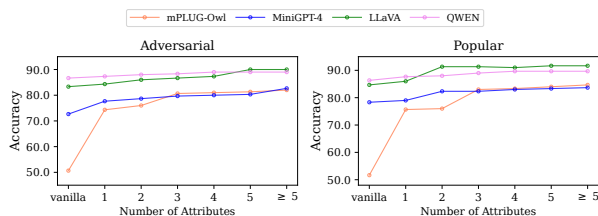


Figure 6: The performance of different number of attributes  $n$ .

## 5 Hyperparameter Analysis

In this section, we conduct experiments on POPE under adversarial and popular setting to analyze the performance fluctuation of LogicCheckGPT with different values of key hyperparameters.

### 5.1 Logical closed loop threshold $\lambda$

We test our method with values of threshold  $\lambda$  ranging from 0.0 to 0.9 as shown in Figure 5 and have the following observations:

There is a significant increase when threshold  $\lambda$  is increasing from 0.0 initially for all models with LogicCheckGPT under both settings. It indicates that our method can significantly distinguish existent objects and hallucinated objects by valid threshold, as models tend to provide logically consistent responses for existent objects.

Subsequently, these these models reach their respective performance peaks. For instance, the performance of mPLUG-Owl achieves the best when  $\lambda$  is 0.4 under adversarial setting, and the best  $\lambda$  for LLaVA is also 0.4. For MiniGPT-4, the  $\lambda$  at which it reached its performance peak varied across different settings, 0.2 and 0.1 respectively. With the continued increase of the threshold, a noticeable decrease in performance can be observed. This is reasonable, as it leads to the misclassification of a large number of existent objects as non-existent.

### 5.2 The number of attributes $n$

The number of attributes  $n$  obtained during the object-to-attribute stage determines the number

of questions that can be generated from these attributes. We report the performance when  $n = 1, 2, 3, 4, 5$  and  $\geq 5$  (See Figure 6) and summarize the observations as follows:

It can be observed that there is an obvious performance improvement when the number of attributes increases across the four LVLMs. This improvement is reasonable as it allows for a greater range of inquiries to verify the presence of the object. As  $n$  exceeds 5, the improvement becomes marginal.

## 5.3 Qualitative Examples

In this section, we have selected two representative examples for mPLUG-Owl covering distinct types of questions, including a binary question, “Is there a dining table in the image?” and an open-ended query, “Please describe this image in detail.”, as illustrated in Fig 3. In the first case 3(a), the model answers “yes” to the binary question, while our method detects that dining table is hallucinated and corrects the output. As for open-ended text generation, the models tends to hallucinate nonexistent objects as the length of the generated sequence increases. However, our method is capable of individually verifying the existence of objects, thereby mitigating hallucinations. More qualitative analysis can be referred in Section F.

## 6 Conclusion

We propose a novel logical closed loop-based framework LogicCheckGPT for object hallucination mitigation in LVLMs. Our motivation stems from the observation that LVLMs often exhibit logically inconsistent responses to hallucinated objects. Therefore, we devise logic consistency probing, which involves asking questions with logical correlations, such as inquiring about attributes from objects and vice versa. Specifically, we break down this process into several steps: object extraction, object-to-attribute inquiring, attribute-to-object inquiring, logical closed-loop checking, and hallucination detection and mitigation. Comprehensive experiments conducted on several benchmarks demonstrate the superiority of our framework.

## Limitations

In this work, we propose a logical closed loop-based framework for object hallucination mitigation. However, our method still has the following two limitations. Firstly, adopting our framework inevitably incurs costs, as we rely on the GPT-3.5

API. Secondly, our work only focuses on addressing object hallucinations. There are also other types of hallucinations, including attribute hallucinations and knowledge hallucinations. Therefore, extending our framework to encompass a broader range of hallucination mitigation represents our future directions.

## Acknowledgements

This work is supported by National Natural Science Foundation of China (62372454,62141608, 62236010).

## References

- Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Xiang Chen, Chenxi Wang, Yida Xue, Ningyu Zhang, Xiaoyan Yang, Qiang Li, Yue Shen, Jinjie Gu, and Huajun Chen. 2024. Unified hallucination detection for multimodal large language models. *arXiv preprint arXiv:2402.03190*.
- Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%\* chatgpt quality. See <https://vicuna.lmsys.org> (accessed 14 April 2023).
- Roi Cohen, May Hamri, Mor Geva, and Amir Globerson. 2023. Lm vs lm: Detecting factual errors via cross examination. *arXiv preprint arXiv:2305.13281*.
- W Dai, J Li, D Li, AMH Tiong, J Zhao, W Wang, B Li, P Fung, and S Hoi. 2023a. Instructblip: Towards general-purpose vision-language models with instruction tuning. *arXiv preprint arXiv:2305.06500*.
- Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023b. Instructblip: Towards general-purpose vision-language models with instruction tuning. *arXiv preprint arXiv:2305.06500*.
- Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. 2023. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*.
- Anisha Gunjal, Jihan Yin, and Erhan Bas. 2023. Detecting and preventing hallucinations in large vision language models. *arXiv preprint arXiv:2308.06394*.
- Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. 2023. Opera: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. *arXiv preprint arXiv:2311.17911*.
- Liqiang Jing, Ruosen Li, Yunmo Chen, Mengzhao Jia, and Xinya Du. 2023. Faithscore: Evaluating hallucinations in large vision-language models. *arXiv preprint arXiv:2311.01477*.
- Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2022. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.
- Seongyun Lee, Sue Hyun Park, Yongrae Jo, and Minjoon Seo. 2023. Volcano: mitigating multimodal hallucination through self-feedback guided revision. *arXiv preprint arXiv:2311.07362*.
- Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. 2023. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. *arXiv preprint arXiv:2311.16922*.
- Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. 2023. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726*.
- Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*.
- Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. 2024. Mitigating hallucination in large multi-modal models via robust instruction tuning. *ICLR*.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *NeurIPS*.
- Holy Lovenia, Wenliang Dai, Samuel Cahyawijaya, Ziwei Ji, and Pascale Fung. 2023. Negative object presence evaluation (nope) to measure object hallucination in vision-language models. *arXiv preprint arXiv:2310.05338*.
- Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Bin Wang, Fan Wu, Xiao Han, Jiahui Peng, Huaping Zhong, Pan Zhang, Xiaoyi Dong, Weijia Li, Wei Li, Jiaqi Wang, et al. 2023. Vigc: Visual instruction generation and correction. *arXiv preprint arXiv:2308.12714*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.
- Peng Xu, Wenqi Shao, Kaipeng Zhang, Peng Gao, Shuo Liu, Meng Lei, Fanqing Meng, Siyuan Huang, Yu Qiao, and Ping Luo. 2023. Lvlm-ehub: A comprehensive evaluation benchmark for large vision-language models. *arXiv preprint arXiv:2306.09265*.
- Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. 2023. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*.
- Li Yifan, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.
- Shukang Yin, Chaoyou Fu, Sirui Zhao, Tong Xu, Hao Wang, Dianbo Sui, Yunhang Shen, Ke Li, Xing Sun, and Enhong Chen. 2023. Woodpecker: Hallucination correction for multimodal large language models. *arXiv preprint arXiv:2310.16045*.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.
- Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. 2024. Analyzing and mitigating object hallucination in large vision-language models. *ICLR*.
- Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

## A Implementation Details

In this work, LogicCheckGPT was constructed utilizing the PyTorch framework (Paszke et al., 2019), incorporating capabilities from HuggingFace’s Transformers library (Wolf et al., 2019). The large language model (LLM) we adopted is GPT-3.5-turbo, to help fulfill each subtask of hallucination mitigation. All base LVLMS and hallucination mitigation methods are re-implemented according to their literature. We maintain the default hyperparameter settings for all backbone LVLMS and baselines. The experiments were conducted using an NVIDIA A100 GPU and an AMD EPYC 7763 CPU. The hallucination threshold  $\lambda$  is searched within  $[0.0, 0.9]$ . The  $\lambda$  is set 0.4, 0.2, 0.4 and 0.2 for mPLUG-Owl, MiniGPT-4, LLaVA-1.5, and QWEN-VL-Chat respectively. The number of attributes  $n$  is set to at least 5. Specifically, in the object-to-attribute inquiring stage, we keep sampling responses from LVLMS until we have obtained at least 5 factored attributes or until we have sampled a maximum of 3 responses.

## B The Choice of LLM for LogicCheckGPT

To investigate the flexibility of our framework, we have conducted experiments by replacing GPT-3.5 with Vicuna-13b-v1.5<sup>3</sup> (Chiang et al., 2023). The results of the Vicuna variant and GPT variant on POPE are summarized in Table 4.

It can be observed that LogicCheckGPT implemented by Vicuna exhibits strong performance across the four LVLMS. Though the Vicuna variant performs slightly worse than the GPT variant, it still demonstrates a strong capability to detect and mitigate object hallucinations within LVLMS.

<sup>3</sup><https://huggingface.co/lmsys/vicuna-13b-v1.5>

Model	Method	Adversarial		Popular		Random	
		Acc	F1	Acc	F1	Acc	F1
mPLUG-Owl	vanilla	50.67	66.81	51.66	67.26	55.33	68.98
	LogicCheckGPT (Vicuna)	80.67	81.98	83.33	84.17	89.00	88.96
	LogicCheckGPT (GPT)	<b>82.00</b>	<b>82.23</b>	<b>84.66</b>	<b>84.45</b>	<b>91.00</b>	<b>90.84</b>
MiniGPT-4	vanilla	72.67	75.88	78.33	79.87	84.33	84.59
	LogicCheckGPT (Vicuna)	81.00	80.27	86.00	84.78	<b>87.67</b>	<b>86.34</b>
	LogicCheckGPT (GPT)	<b>82.67</b>	<b>80.59</b>	<b>83.67</b>	<b>81.51</b>	86.67	85.29
LLaVA-1.5	vanilla	83.33	84.84	84.67	85.89	93.00	<b>93.02</b>
	LogicCheckGPT (Vicuna)	87.33	88.05	87.00	87.77	93.00	92.63
	LogicCheckGPT (GPT)	<b>90.00</b>	<b>89.58</b>	<b>91.67</b>	<b>91.40</b>	<b>93.33</b>	93.00
QWEN-VL-Chat	vanilla	86.67	86.67	86.33	86.37	90.67	90.28
	LogicCheckGPT (Vicuna)	87.67	87.28	88.33	87.88	90.33	89.89
	LogicCheckGPT (GPT)	<b>89.00</b>	<b>88.00</b>	<b>89.67</b>	<b>88.64</b>	<b>91.33</b>	90.71

Table 4: The performance comparison between LogicCheckGPT implemented by Vicuna and GPT on POPE.

Model	# Existent Objects	# Hallucinated Objects
mPLUG-Owl	3.966 ± 1.3427	3.510 ± 1.1783
MiniGPT-4	6.576 ± 3.6195	6.322 ± 4.4117
LLaVA-1.5	5.414 ± 1.5077	5.375 ± 1.2747
QWEN-VL-Chat	4.353 ± 1.6652	3.850 ± 1.1821

Table 5: The average number of attributes extracted in responses when the LVLm is prompted to describe existent or hallucinated objects. The symbol “#” denotes “the number of”.

We also delve into the failure cases of the Vicuna variant and observe that this was due to errors occurring in object extraction and logical closed loop checking. These shortcomings primarily stem from inherent limitations within Vicuna itself and the design of prompts. This underscores the efficacy of our approach and the potential for substituting GPT with alternatives at each stage. As we have unified the framework and divided the whole process into several simple steps, including object extraction, attribute extraction, and rephrasing, we retain the flexibility to substitute GPT with alternative methods tailored to each step, potentially enhancing both efficiency and accuracy.

### C The distribution of the number of extracted attributes

To investigate whether there is a difference in the distribution of the number of extracted attributes between existent and hallucinated objects, we analyzed the number of attributes extracted from each description for both types of objects across 300 POPE adversarial instances, as shown in Table 5.

We have the following observations.

There is a greater abundance of described attributes and a wider variance of existent objects among existent objects compared to hallucinated objects across the four LVLms. This phenomenon can be attributed to the LVLms could provide more detailed descriptions of existent objects based on the image while lacking a solid foundation for hallucinated objects. Notably, the MiniGPT-4 tends to describe more attributes for both existent and hallucinated objects, which aligns with our observation that responses generated by MiniGPT-4 are lengthier. However, as the numbers of attributes for both objects are close, it is still difficult to distinguish them only by the attribute count.

In our work, the number of attributes obtained during the object-to-attribute stage is set to at least 5. Specifically, in the object-to-attribute inquiring stage, we keep sampling responses from LVLms until we have obtained at least 5 factored attributes or until we have sampled a maximum of 3 responses. This approach aims to gather a sufficient number of attributes for the Logical Closed Loop checking process, without being hindered by limitations in attribute quantity. It has been validated in Section 5.2 that as the number of attributes exceeds 5, the improvement becomes marginal.

## D Prompts

### D.1 Object Extraction

The prompt for object extraction is illustrated in Fig. 7.



## **D.2 Attribute-to-Object Inquiring**

### **D.2.1 Attribute Extraction**

The prompt for attribute extraction in attribute-to-object inquiring is illustrated in Fig. 8.

### **D.2.2 Question Formulation**

The prompt for question formulation in attribute-to-object inquiring is illustrated in Fig. 9.

## **D.3 Logical Closed Loop Checking**

The prompt for logical closed loop checking is illustrated in Fig. 10.

## **D.4 Hallucination Detection and Mitigation**

The prompt for hallucination detection and mitigation is illustrated in Fig. 11.

## **D.5 GPT-4V Assisted Evaluation**

The prompt for hallucination detection and mitigation is illustrated in Fig. 12.

## **E FreeCheck Prompts**

The prompts for the ablation variant *FreeCheck* in different stages is illustrated in Table 6.

## **F Qualitative Analysis**

In this section, we visualize several examples of our LogicCheckGPT applied to various LLMs, as shown in Fig. 13, 14, 15. As depicted in 13, for the MPLUG-Owl example, our method effectively identifies the hallucinated objects “people” and “handbag” both of which have logical closed-loop rates of 0.0. In contrast, for MiniGPT-4 (see Fig. 14), our method successfully detects the “building” but fails to identify the hallucinated object “trees”. This failure is attributed to the fact that the described hallucinated attributes are strongly correlated with the object and may inadvertently reveal the object’s identity. For instance, phrases like “have branches that stretch out from their trunks” and “have lush green leaves” suggest that the objects are trees. In addition, for LLaVA-1.5 in Fig. 15, LogicCheckGPT assigns a score of 1.00 to the existent object “clock” and 0.25 to the hallucinated object “person”. Since the score for "person" is below the threshold utilized in our study, we can detect the hallucinated object "person".

Stage	Prompt(s)
(1) System Prompt	You are a language assistant that helps to answer the question according to instructions
(2) Setup	Your goal is to validate a description of an image that declares that an object exists in the image. You have the opportunity to ask multiple questions to verify the authenticity. The claim is: {claim}. Please start with several questions to ask attributes of the object.
(3) Follow-Up Questions	(i) Do you have any follow-up questions? Please only answer with Yes or No. (ii) What are the follow-up questions to the claim: {claim}?
(4) Decision	You have ended the question session due to limited rounds. Based on the interviewee's answers to your questions, what is your conclusion about the question: {question}? Please answer yes/no in the first line and then output explanation.

Table 6: Prompts used for the ablation variant *FreeCheck* in different stages.

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

---

**Prompt**  
You are given a sentence, extract the entities within the sentence for me.

**[Task]**  
Your task is to extract the common objects and summarize them as general categories without repetition, merging essentially similar objects. Avoid extracting abstract or non-specific entities. 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". DO NOT RESPOND WITH ANYTHING ELSE.

Here are some examples:  
[{In-context examples}](#)

Now complete the following:

**[Sentence]**  
{sentence}

**[Response]**

Figure 7: Prompt template of object extraction.

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

---

**Prompt**  
You will receive a piece of text that describes an object, and the given object.

**[Task]**  
Your task is to accurately identify and extract every attribute associated with the given object in the provided text. Each claim should be concise (less than 15 words) and self-contained, corresponding to only one attribute. You MUST only respond in the format as required. Each line should contain the original claim and the modified claim with all the mentions of the given object being replaced with "the object". DO NOT RESPOND WITH ANYTHING ELSE. ADDING ANY OTHER EXTRA NOTES THAT VIOLATE THE RESPONSE FORMAT IS BANNED.

**[Response Format]**  
original claim and modified claim

Here are some examples:  
[{In-context examples}](#)

Now complete the following:

**[Text]**  
{sentence}

**[Entity]**  
{entity}

**[Response]**

Figure 8: Prompt template of attribute-to-object question (1).

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

---

**Prompt**  
You will receive a list of statements of objects in an image.

**[Task]**  
Your task is to rephrase each line of statement into a question following the below question template. In specific, extract attributes of the object to fill in the attribute slot of the template to form questions. DO NOT RESPOND WITH ANYTHING ELSE. DO NOT CHANGE THE QUESTION TEMPLATE.

**[Response Format]**  
Could you tell me all the objects that {ATTRIBUTE SLOT} in the image?

Here are some examples:  
{In-context examples}

Now complete the following:

**[Statements]**  
{statement}

**[Response]**

Figure 9: Prompt template of attribute-to-object question (2).

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

---

**Prompt**  
You are given a statement and a question.

**[Task]**  
Your task is to answer the question based on the statement. The statement is about some objects. The question is to ask whether some specific object exists.  
1. Your response should be limited to one of the following two choices: "Yes"/"No".  
2. Note that instances of a certain category can also belong to its super-categories. For example, a baseball is a subclass of the sports ball.  
3. Note that the table is equivalent to the dining table here.  
4. DO NOT RESPOND WITH ANYTHING ELSE.

**[Response Format]**  
Yes/No

Here are some examples:  
{In-context examples}

Now complete the following:

**[Statement]**  
{statement}

**[Question]**  
Is there a {object} in the statement?

**[Response]**

Figure 10: Prompt template of logic closed loop check.

### System prompt

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

### Prompt

You are given a query, an original answer and supplementary information.

#### [Task]

You are required to correct the original answer to the query in a fluent and natural style, following these rules:

1. Correct the sentences in the origin answer if they are inconsistent with the supplementary information. Remove the objects that are confirmed to not exist in the supplementary information.
2. Do not modify correct sentences. Do not introduce additional information.
3. When giving the refined answer, also pay attention to the given query. The refined answer should be a reasonable answer to the query.
4. Note the dining table is equivalent to the table.

Output only the corrected answer, without introducing extra contents.

Here are some examples:

`{In-context examples}`

Now complete the following:

#### [Query]

`{query}`

#### [Original Answer]

`{origin_ans}`

#### [Supplementary Information]

`{sup_info}`

#### [Response]

Figure 11: Prompt template of refinement.

### System prompt

You are required to score the performance of two AI assistants in describing a given image.

### Prompt

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. 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: Relevancy:

whether the response directly follows the instruction.

Please output the scores 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]

`{answer_1}`

[End of Assistant 1]

#### [Assistant 2]

`{answer_2}`

[End of Assistant 2]

Output format:

Accuracy: `<Scores of the two answers>`

Reason:

Relevancy: `<Scores of the two answers>`

Reason:

Figure 12: Prompt template of GPT-4v Assisted Evaluation.





**Question:** Please describe this image in detail.

**Vanilla:** The image features a black and white motorcycle parked on a street next to a brick wall. The motorcycle is positioned on the left side of the scene, with its front tire slightly raised off the ground. The motorcycle's handlebars are visible, and the front wheel is slightly turned to the right. There are several **people** in the scene, with some standing closer to the motorcycle and others further away. A person is standing near the left side of the motorcycle, while another person is located closer to the right side. A third person is standing further back, closer to the right side of the scene. Additionally, there are two **handbags** placed in the scene, one near the left side of the motorcycle and the other closer to the right side.

**LogicCheckGPT:** The image features a black and white motorcycle parked on a street next to a brick wall. The motorcycle is positioned on the left side of the scene, with its front tire slightly raised off the ground. The motorcycle's handlebars are visible, and the front wheel is slightly turned to the right. There are no **people** in the scene. Additionally, there are no **handbags** placed in the scene.

### Logical Closed Loop Check

<p><b>Q:</b> Could you please describe the <b>people</b> in the image?</p> <p><b>A:</b> The person in the picture is a man, wearing a black shirt and a black hat. He is standing next to a motorcycle, which is parked on the side of the road.</p> <p style="text-align: center;">Object ↔ Attribute</p> <p><b>Q:</b> Could you tell me all the objects that <b>are wearing a black shirt</b> in the image?</p> <p><b>A:</b> In the image, there are two motorcycles wearing black shirts.</p> <p><b>Q:</b> Could you tell me all the objects that <b>are wearing a black shirt</b> in the image?</p> <p><b>A:</b> In the image, there are two motorcycles wearing black hats.</p> <p><b>Q:</b> Could you tell me all the objects that <b>are standing next to a motorcycle</b> in the image?</p> <p><b>A:</b> In the image, there are several objects standing next to a motorcycle. These objects include a chair, a trash can, a bottle, and a cup.</p> <p><b>Q:</b> Could you tell me all the objects that <b>are parked on the side of the road</b> in the image?</p> <p><b>A:</b> In the image, there are several objects parked on the side of the road, including a motorcycle, a car, and a truck.</p>	<p><b>Q:</b> Could you please describe the <b>handbag</b> in the image?</p> <p><b>A:</b> The handbag in the picture is a black and white handbag with a silver handle. It is located on the ground next to a motorcycle, possibly on the sidewalk or in a parking lot. The handbag appears to be in a state of disarray, with some of its contents spilling out.</p> <p style="text-align: center;">Object ↔ Attribute</p> <p><b>Q:</b> Could you tell me all the objects that <b>are black and white with a silver handle</b> in the image?</p> <p><b>A:</b> In the image, there are two black and white objects with a silver handle. One is a motorcycle, and the other is a bicycle.</p> <p><b>Q:</b> Could you tell me all the objects that <b>are located on the ground next to a motorcycle</b> in the image?</p> <p><b>A:</b> In the image, there are several objects located on the ground next to the motorcycle. These objects include a bottle, a cup, a spoon, a fork, a knife, a pair of scissors, a pair of pliers, a wrench, a hammer, a saw, a pair of gloves, a pair of boots, and a pair of shoes.</p> <p><b>Q:</b> Could you tell me all the objects that <b>appear to be in a state of disarray</b> in the image?</p> <p><b>A:</b> In the image, there are several objects that appear to be in a state of disarray. These objects include a motorcycle, a car, a truck, a bicycle, and a suitcase.</p> <p><b>Q:</b> Could you tell me all the objects that <b>have some of their contents spilling out</b> in the image?</p> <p><b>A:</b> In the image, there are several objects with some of their contents spilling out. These objects include a motorcycle, a truck, a car, a motorcycle helmet, a motorcycle gas tank, a motorcycle seat, a motorcycle chain...</p>				
<table border="1"> <tr> <th colspan="2">Closed Loop Rate</th> </tr> <tr> <td>People: 0.00</td> <td>Handbag: 0.00</td> </tr> </table>		Closed Loop Rate		People: 0.00	Handbag: 0.00
Closed Loop Rate					
People: 0.00	Handbag: 0.00				

Figure 13: The example of our LogicCheckGPT for mPLUG-Owl. The hallucinated objects are highlighted in red and attributes are highlighted in magenta.



**Question:** Please describe this image in detail.

**Vanilla:** The image shows a man in green shorts and a white shirt standing on a blue tennis court. He is holding a tennis racket in his right hand and looking towards the camera. The court has white lines for the service boxes and a yellow line for the center line. There are **trees** and a **building** visible in the background.

**LogicCheckGPT:** The image shows a man in green shorts and a white shirt standing on a blue tennis court. He is holding a tennis racket in his right hand and looking towards the camera. The court has white lines for the service boxes and a yellow line for the center line. There are **trees** visible in the background.

### Logical Closed Loop Check

<p><b>Q:</b> Could you please describe the <b>trees</b> in the image?</p> <p><b>A:</b> The tree in the picture appears to be a tall, thin tree with branches that stretch out from its trunk. It has lush green leaves and a thick, rough bark on its trunk. Its location is difficult to determine based on the photo, but it appears to be growing on a hill or incline.</p> <p style="text-align: center;"> </p> <p><b>Q:</b> Could you tell me all the objects that <b>appear to be tall and thin</b> in the image?</p> <p><b>A:</b> The objects that appear to be tall and thin in the image are the tennis player's legs and the tennis racket. The tennis player is standing on a blue tennis court, wearing a green shirt and white shorts. He is holding a tennis racket in his right hand and appears to be ready to hit a shot.</p> <p><b>Q:</b> Could you tell me all the objects that <b>have branches that stretch out from their trunks</b> in the image?</p> <p><b>A:</b> The objects in the image that have branches that stretch out from their trunks are the <b>trees</b> on the left and right sides of the tennis court.</p> <p><b>Q:</b> Could you tell me all the objects that <b>have lush green leaves</b> in the image?</p> <p><b>A:</b> The objects in the image that have lush green leaves are the <b>trees</b> on the left and right sides of the tennis court.</p> <p><b>Q:</b> Could you tell me all the objects that <b>have a thick and rough bark on their trunks</b> in the image?</p> <p><b>A:</b> The objects in the image that have a thick and rough bark on their trunks are the <b>trees</b>.</p> <p><b>Q:</b> Could you tell me all the objects <b>whose location is difficult to determine based on the photo</b> in the image?</p> <p><b>A:</b> The objects whose location is difficult to determine based on the photo in the image are the tennis player's head and the ball he is about to hit.</p>	<p><b>Q:</b> Could you please describe the <b>building</b> in the image?</p> <p><b>A:</b> (1) The building in the picture appears to be a tennis court. It has a blue surface with white lines on it, and there is a man standing on the court holding a tennis racket. The court appears to be outdoors, as there are trees visible in the background. (2) The building in the picture is a tennis court with blue walls and a green roof. It appears to have a net stretched across its center, indicating that it is used for playing tennis.</p> <p style="text-align: center;"> </p> <p><b>Q:</b> Could you tell me all the objects that <b>have a blue surface with white lines on them</b> in the image?</p> <p><b>A:</b> The objects in the image that have a blue surface with white lines on them are the tennis court.</p> <p><b>Q:</b> Could you tell me all the objects that <b>appear to be outdoors</b> in the image?</p> <p><b>A:</b> The image shows a man playing tennis on a blue court. He is wearing a green shirt and white shorts, and he is holding a tennis racket in his right hand. Behind him, there is a white fence and some trees that are visible through the fence. The sky appears to be cloudy, and there is a shadow of the man and the fence on the court.</p> <p><b>Q:</b> Could you tell me all the objects that <b>have blue walls and a green roof</b> in the image?</p> <p><b>A:</b> The image shows a tennis player on a blue and green tennis court. The player is wearing a white shirt, green shorts, and white sneakers. He is holding a tennis racket in his right hand and appears to be about to hit the ball. In the image, There are no objects in the image that have blue walls and a green roof.</p> <p><b>Q:</b> Could you tell me all the objects that <b>have a net stretched across their center</b> in the image?</p> <p><b>A:</b> The objects in the image that have a net stretched across their center are the tennis court and the tennis racket.</p>				
<table border="1"> <tr> <th colspan="2">Closed Loop Rate</th> </tr> <tr> <td>Trees: 0.60</td> <td>Building: 0.00</td> </tr> </table>		Closed Loop Rate		Trees: 0.60	Building: 0.00
Closed Loop Rate					
Trees: 0.60	Building: 0.00				

Figure 14: The example of our LogicCheckGPT for MiniGPT-4. The examinee hallucinated objects are highlighted in red and attributes are highlighted in magenta. We sample two descriptions of “Building” to obtain sufficient attributes.



**Question:** Please describe this image in detail.

**Vanilla:** The image features a group of colorful umbrellas, predominantly yellow and blue, covering a large area in front of a building. The umbrellas are arranged in various positions, creating a visually appealing scene. Some umbrellas are placed closer to the building, while others are spread out across the area. In addition to the umbrellas, there are a few potted plants scattered throughout the scene, adding to the overall aesthetic. A **clock** can be seen on the side of the building, and a **person** is visible in the background, possibly enjoying the view or attending to the umbrellas.

**LogicCheckGPT:** The image features a group of colorful umbrellas, predominantly yellow and blue, covering a large area in front of a building. The umbrellas are arranged in various positions, creating a visually appealing scene. Some umbrellas are placed closer to the building, while others are spread out across the area. In addition to the umbrellas, there are a few potted plants scattered throughout the scene, adding to the overall aesthetic. A **clock** can be seen on the side of the building, and there is no **person** visible in the background.

### Logical Closed Loop Check

<p><b>Q:</b> Could you please describe the <b>clock</b> in the image?</p> <p><b>A:</b> The clock in the picture is located on the side of a building, possibly a restaurant or a cafe. It is positioned above the entrance, making it easily visible to people entering or exiting the establishment. The clock is large and prominently displayed, likely serving as a functional and decorative element for the building. The presence of the clock suggests that the building may be a place where people gather, socialize, or spend time, and it adds a touch of elegance and sophistication to the overall design.</p> <p style="text-align: center;"> </p> <p><b>Q:</b> Could you tell me all the objects that <b>are located on the side of a building</b> in the image?</p> <p><b>A:</b> There are several objects located on the side of a building. These objects include a <b>clock</b>, a row of yellow and blue umbrellas, and a row of blue and yellow awnings.</p> <p><b>Q:</b> Could you tell me all the objects that <b>are positioned above the entrance</b> in the image?</p> <p><b>A:</b> There are two motorcycles wearing black hats. there are several objects positioned above the entrance. These include a <b>clock</b>, a row of yellow and blue umbrellas, and a row of yellow and blue flags.</p> <p><b>Q:</b> Could you tell me all the objects that <b>are large and prominently displayed</b> in the image?</p> <p><b>A:</b> There are several large and prominently displayed objects. These include a large <b>clock</b> on the side of a building, a group of yellow and blue umbrellas, and a row of blue and yellow striped umbrellas.</p> <p><b>Q:</b> Could you tell me all the objects that <b>serve as a functional and decorative element for the building</b> in the image?</p> <p><b>A:</b> The most prominent feature is the large yellow and blue umbrellas, which provide shade and protection from the elements while also ... Additionally, there are potted plants placed around the building ... The presence of a <b>clock</b> on the building's facade serves a functional purpose by allowing passersby to keep track of time.</p>	<p><b>Q:</b> Could you please describe the <b>person</b> in the image?</p> <p><b>A:</b> There is a person standing near buildings under white umbrellas with yellow and blue poles. Although the person's face is not visible, they seem to be part of the scene and might be an attendee of an outdoor event, or simply someone walking by the buildings during a sunny day. The presence of white umbrellas and their location under it suggests they might be seeking shade and protection from the sun.</p> <p style="text-align: center;"> </p> <p><b>Q:</b> Could you tell me all the objects that <b>are standing near buildings under white umbrellas with yellow and blue poles</b> in the image?</p> <p><b>A:</b> In the image, there are several objects standing near buildings under white umbrellas with yellow and blue poles. These objects include a clock, a potted plant, and a couple of umbrellas.</p> <p><b>Q:</b> Could you tell me all the objects that <b>are that have their faces not visible</b> in the image?</p> <p><b>A:</b> In the image, there are several objects that have their faces not visible. These include a clock, a window, and a building.</p> <p><b>Q:</b> Could you tell me all the objects that <b>seem to be part of the scene</b> in the image?</p> <p><b>A:</b> In the image, there are several objects several objects that seem to be part of the scene. These include a building with a clock on its side, and a group of yellow and blue umbrellas.</p> <p><b>Q:</b> Could you tell me all the objects that <b>might be someone walking by the buildings during a sunny day</b> in the image?</p> <p><b>A:</b> In the image, a <b>person</b> is walking by the buildings during a sunny day.</p>				
<table border="1"> <thead> <tr> <th colspan="2">Closed Loop Rate</th> </tr> </thead> <tbody> <tr> <td>Clock: 1.00</td> <td>Person: 0.25</td> </tr> </tbody> </table>		Closed Loop Rate		Clock: 1.00	Person: 0.25
Closed Loop Rate					
Clock: 1.00	Person: 0.25				

Figure 15: The example of our LogicCheckGPT for LLaVA-1.5. The examinee hallucinated objects are highlighted in red, the examinee existent objects are highlighted in blue, and attributes are highlighted in magenta.