Evaluating and Mitigating Number Hallucinations in Large Vision-Language Models: A Consistency Perspective

Anonymous ACL submission

Abstract

Large vision language models have demonstrated remarkable efficacy in addressing challenges related to both textual and visual content. Nevertheless, these models are susceptible to various hallucinations. In this paper, we focus on a new form of hallucination, specifically termed as number hallucination, which denotes instances where models fail to accurately identify the quantity of objects in an image. We establish a dataset and employ evaluation metrics to assess number hallucination, revealing a pronounced prevalence of this issue across mainstream large vision language models (LVLMs). Additionally, we delve into a thorough analysis of number hallucination, examining inner and outer inconsistency problem from two related perspectives. We assert that this inconsistency is one cause of number hallucination and propose a consistency training method as a means to alleviate such hallucination, which achieves an average improvement of 8% compared with direct finetuning method.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities in solving diverse problems by adhering to human instructions. Consequently, researchers have undertaken numerous endeavors to integrate multiple modalities into a single large language model, with the most successful being large vision-language models (LVLMs). LVLMs are designed to process both images and text as input and generate text-only responses. Among all LVLMs, GPT4-V(OpenAI et al., 2023)stands out as the most powerful one, showcasing extraordinary proficiency in addressing various vision-text challenges, including tasks such as vision question answering (VQA) and image captioning.

Nevertheless, despite the success of large vision-language models, they still grapple with several challenges, with hallucination being the most severe among them. In this context, hallucination primarily pertains to the discrepancy between the model's output and the content present in the input image. Among various forms of hallucination, object hallucination, which centers on determining whether an object is present in the image, has been extensively discussed. Numerous methods for evaluation and mitigation have been proposed to tackle this particular challenge(Zhou et al. (2023), Leng et al. (2023), Zhai et al. (2023)). The majority of prior studies employ binary classification questions (answered with 'yes' or 'no') to assess the model's performance(Li et al. (2023b), Wang et al. (2023b)).

While simple object hallucination has been extensively studied, a more nuanced and broadly applicable discussion on object hallucination is still lacking. For a comprehensive understanding of an image, the model must not only discern the presence of a specific object but also accurately determine the quantity of that particular object. Without this capability, the model cannot fully comprehend the image, thereby leaving the door open to potential risks of other forms of hallucination.

In this study, we focus on investigating a new form of object hallucination, specifically termed as number hallucination. Number hallucination occurs when models are unable to accurately identify the correct quantity of a particular type of object in an image. Table 1 illustrates two typical examples of number hallucination. We also conduct detailed experiments and point that using binary classification questions is not enough to fully assess the model's performance. Instead, we choose to use a direct inquiry to the model about the number of a specific object in an image, utilizing prompts to control its output. This approach is more aligned with real-world scenarios and facilitates a more quantitative evaluation of model performance.

Furthermore, in the context of the counting task, there exist multiple related tasks that help reflect overall performance such as binary classification

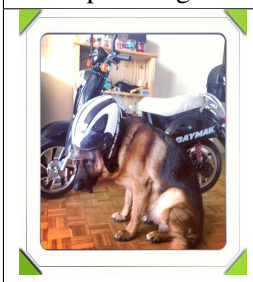
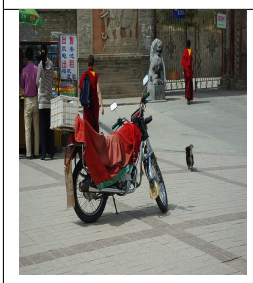
Input Image	Prompt	Output
	How many motorcycles are there in this picture? Answer in a single number.	2 (Gold answer is 1.)
	How many persons are there in this picture? Answer in a single number.	3 (Gold answer is 4.)

Table 1: Two examples of number hallucination.

questions (Section 3.1) and comparative questions (Section 3.2). Naturally we would like to investigate the consistency between these tasks to attain a more comprehensive understanding of number hallucination. We find that inconsistency is a common problem that appears despite task format. Building upon these analyses, we introduce a consistency training method to enhance the model's capability. Our findings demonstrate that finetuning the model from different perspectives yields superior results compared to direct finetuning on the single primary task. This methodological insight is intriguing and holds potential for extension to various tasks that necessitate finetuning of LVLMs.

To summarize, our contributions are listed as follows:

- We introduce a new form of object hallucination, specifically referred to as number hallucination, and present a dataset with 20k pieces of data along with corresponding evaluation results. We point out that all LVLMs we investigated have an average MAE of around 2 on this dataset, which is disasterous. To the best of our knowledge, our study represents the first comprehensive and quantitative exploration of this particular type of hallucination.

- We conduct a detailed analysis of number hallucinations by exploring the inner and outer inconsistency from different perspectives. We point out that all models suffer from both inner and outer inconsistencies despite task or prompt format, which reveals the model's uncertainty and confusion and is a potential cause of number hallucination.

- We propose a consistency training method which is extendable to any large vision-language model (LVLM). We substantiate the effectiveness of this method through our experiments by averagely 8% compared with directly finetuning methods.

2 Evaluating Number Hallucination in LVLMs

In this section, we begin by providing a concise definition of number hallucination in large vision-language models (LVLMs) and outline our evaluation settings for assessing this phenomenon. Subsequently, we present the results of our evaluation and provide brief discussions on the findings.

2.1 Task Definition

Number hallucination occurs when models fail to accurately determine the correct quantity of a specific object category in a given image. Formally, for an image i and given object c, let n_c denote the ground truth number of c visible in i. If a model responds with $n^{'}(\neq n_c)$ c in i (in any format), we classify the model as suffering from number hallucination.

As observed, number hallucination can manifest in various vision-language tasks, including image captioning and vision-question answering (VQA). For the sake of simplicity in quantitative evaluation, we opt for VQA as the foundational format for assessing number hallucinations. For any given image i and object c, the question is formatted as:

Prompt: How many c are there in this picture? Answer in a single number.

This task format offers several advantages. In comparison to caption-based evaluation methods, this format makes it considerably easier to extract meaningful responses (numbers) and provides a clearer indication of the model's performance. Captions may segregate the same object into different sentences or overlook existing objects, even when the model has detected them. These complexities in captions add extra challenges to the evaluation of number hallucinations.

In contrast to binary classification-based methods, firstly, this approach aligns more closely with real-life scenarios and provides a clearer insight into the model's view on this task. Secondly, as

can be seen in Table 3, all models exhibit a notable inclination to answer 'Yes' more frequently than 'No'. This observation underscores the inherent bias of all models toward answering 'Yes', making evaluation methods employing binary classification questions less reliable. Binary classification questions also suffer from severe inconsistency problems, which will be further discussed in Section 3.1. Lastly, binary classification questions fail to capture the severity of hallucinations. For instance, mistaking 4 cats as 3 cats may be relatively acceptable, but mistaking 4 cats as 10 cats is disastrous and entirely unacceptable. It is difficult for binary classification questions to delineate this fine-grained difference.

2.2 Dataset and Evaluation Metrics

Dataset MSCOCO (Lin et al., 2014) is a large-scale object detection, segmentation, and captioning dataset. All tests in this study are conducted on the MSCOCO2014 validation set.[1] Leveraging the provided detection and segmentation labels, we can construct a counting VQA dataset. Symbolically, for any given image i with labeled bounding boxes $\{(b_1, c_1), ..., (b_m, c_m)\}$, where b_i represents a bounding box and c_i represents its corresponding object, the total number n_o of a certain object o in image i can be denoted as:

$$n_o = \sum_{k=1}^{m} I[c_k = o]$$

where

$$I[x] = \begin{cases} 1, & x \; is \; True \\ 0, & otherwise \end{cases}$$

Following this approach, we can construct a dataset with ground truth where each data instance can be denoted as (i, c, n_c), with i denoting the given image and c denoting the specified object. However, based on our observation, the distribution of answers is not uniform. For instance, in the original dataset, there are about 68% questions with answer $n_c = 1$. What's more, the distribution of objects is also non-uniform. To address these biases, we propose double k-uniform sampling. The core of this algorithm is limiting the number of data with object c and answer $n_c = n$ not exceeding k. The details of our algorithm can be found in Algorithm 1. With this method we construct our evaluation dataset with 20k pieces of data. For details of our algorithm and dataset, please refer to Appendix A.

Evaluation Metrics In evaluating the counting task, our focus extends to two perspectives. Firstly, we aim for the model to provide correct answers to as many questions as possible. Secondly, we seek minimal errors. To address the first perspective, we treat this problem as a multi-classification task and employ F-scores(macro-F1 and weighted-F1) to evaluate the model's performance. For the second perspective, we use mean absolute error (MAE) as an evaluation metric. The employment of both metrics provides a more comprehensive and quantitative understanding of the severity of number hallucination. For instance, a model that never accurately counts objects, even with a small MAE, is considered ineffective. Similarly, a model that occasionally counts objects correctly but often makes significant errors should also be deemed problematic. Only when a model that exhibits both high F-scores and low MAE can it be considered free from number hallucinations.

2.3 Evaluation Results

We assess number hallucinations using four of the most advanced large vision-language models (LVLMs)[2], which are LLaVA-v1.5-7B(namely LLaVA), LLaVA-v1.5-13B (namely LLaVA-L)(Liu et al., 2023a), InstructBLIP (Dai et al. (2023), equipped with Vicuna-7B) and MiniGPT-v2 (Chen et al. (2023a), equipped with LLaMA-2-chat-7B). The results are shown in Table 2. We also add a random result (randomly select an answer ranging from from 1 to 10) as a comparison.

Model	Macro-F1	Weighted-F1	MAE
Random	0.086	0.103	3.56
LLaVA	0.186	0.288	1.84
LLaVA-L	0.196	0.299	1.80
InstructBLIP	0.201	0.300	2.35
MiniGPT-v2	0.193	0.293	2.06

Table 2: Number hallucination evaluation results of different LVLMs.

We can see all models exhibit severe number hallucinations, as their performance over Macro-F1, Weighted-F1 and MAE is generally poor. For

[1]Test set is not used because we cannot find annotation results on the official website.

[2]GPT-4V is not used because this large-scale evaluation is very expensive and time-consuming.

instance, they exhibit an average MAE of about 2, while random answers just bear an MAE of 3.56. Even LLaVA-v1.5 models, which demonstrate higher POPE (Liu et al., 2023a)(indicating less severe object hallucination), still experience number hallucinations similar to other models. This outcome underscores the limitations of traditional object hallucination studies and evaluations, and emphasizes the significance of addressing this number hallucination.

Furthermore, the comparison between LLaVA and LLaVA-L yields a surprising result that larger models do not significantly outperform smaller models. This finding suggests that merely scaling up models might not be an effective strategy for mitigating number hallucination, indicating that the issue may not solely stem from a lack of expressive ability. This observation underscores the need for more detailed and comprehensive studies on number hallucination.

3 Consistency Analysis from Multiple Perspectives

In addition to the primary counting task mentioned in Section 2.1 (namely primal task), there are also relevant tasks that provide insights into the number hallucination exhibited by a model from different perspectives. Therefore, we can achieve a more comprehensive understanding of number hallucination by analyzing the consistency among the outputs of the related tasks and the primal task. In this context, we mainly analyze two types of consistency, which are inner consistency and outer consistency. Inner consistency represents the consistency of output within the same task format with different settings, while outer consistency represents the consistency between outputs of the primal task and the related task. We take two related tasks and reveal the severe inner and outer inconsistency that both tasks exhibit. This inconsistency is a common problem shared by both tasks, and is a potential cause of number hallucination.

3.1 Binary Classification Questions

Binary classification questions are widely used for evaluating hallucinations in LVLMs. In this task, for any given object c and number n, the binary classification question can be formatted as:

Prompt: There is(are) n c in this picture, is that correct? Answer yes or no.

By modifying n, we can generate different questions and explore the inner inconsistency between different settings of n. To analyze outer inconsistency, we can specify n as the response to direct asking question (the prompt given in Section 2.1) of the model. We generate questions using the following three ways:

- I: Let n equal to the ground truth.
- II: Let n equal to the model's own generated answer.
- III: With a 50% probability, uniformly select n around the ground truth. Otherwise, let n equal to the ground truth.

Model	Prompt	Accuracy	'Yes' Ratio
LLaVA	I	0.749	0.749
	II	0.437	0.843
	III	0.649	0.604
LLaVA-L	I	0.755	0.755
	II	0.365	0.920
	III	0.604	0.656
InstructBLIP	I	0.992	0.992
	II	0.346	0.948
	III	0.513	0.984
MiniGPT-v2	I	0.751	0.751
	II	0.521	0.742
	III	0.618	0.639

Table 3: Evaluation results on binary classification questions. Accuracy is calculated with the ground truth *yes* or *no* answers. 'yes' ratio is the ratio of *yes* answers generated by the model.

Inner Inconsistency In this context, inner inconsistency refers to model responding 'yes' to even contradictory questions. An example is presented in Table 5 in Appendix B. Despite the evident contradiction between the two questions (different n setting), the model answer 'yes' to both. In fact, by comparing prompt I and prompt II, model shows this inner inconsistency at a ratio of 41%, which is very high because a model that randomly answers 'yes' or 'no' will bear an inner inconsistency of just about 25%. This high ratio reveals a significant challenge that it becomes uncertain what the model's actual understanding is towards the question. Specifically, if the model answers 'yes' to a ground truth, we cannot confidently assert that the model precisely knows the number of objects in the given image. Moreover, even if the model answers 'no' to a hallucinated expression, it remains

unclear whether the model accurately comprehends the ground truth or not. This inner inconsistency suggests that the model struggles to count the number of objects accurately and exhibits confusion, thereby contributing to number hallucination.

Outer Inconsistency In this context, outer inconsistency refers to models responds 'no' to prompt II (because prompt II sets n equaling to its own answer to the direct asking question). If it says this answers is incorrect, it conflicts with itself, which is inconsistent. As can be seen from the 'yes' ratio of prompt II, although model generally tends to agree with its own answers (prompt II), there are still notable instances where the model disagree with its own answers. In fact, models have an average inconsistency rate of 16.5% with its own answers except InstructBLIP. For InstructBLIP, it answers 'yes' to almost every question, and interestingly it still has the lowest 'yes' ratio of prompt II among the three prompts. This outer inconsistency further reveals the model's uncertainty about the counting problem and may contribute to number hallucination.

3.2 Comparison Questions

In addition to directly querying the number of a certain object, assessing the model's ability by comparing the numbers of two objects can also provide insights. Comparison is a task that shares both similarities and differences with direct questioning. To some extent, it is more complex as it involves counting two different objects and comparing the results. However, since the comparison task does not necessitate the model to know the exact number of any object, it cannot fully represent the model's counting ability.

Given an image and two objects c_1 and c_2, the corresponding comparison question can be formatted in the following two ways. By analyzing the output of both prompts and the primal task, we can also observe inner and outer inconsistency.

- I: *Prompt: Which object has a larger number in this picture, c_1 or c_2? If they have the same number, answer same.*

- II: *Prompt: Please select the correct option according to the given picture. A. There are more c_1 than c_2 in this picture. B. There are more c_2 than c_1 in this picture. C. The number of c_1 and c_2 are the same in this picture.*

It can be observed that c_1 and c_2 are symmetric in this format. To mitigate the influence of order and analyze inner consistency, each question is repeated with c_1 and c_2 flipped. The results is shown in Table 4 [3].

Model	Prompt	Accuracy
LLaVA	I	0.456
	II	0.368
LLaVA-L	I	0.478
	II	0.412
InstructBLIP	I	0.377
	II	-
MiniGPT-v2	I	0.448
	II	0.444

Table 4: The results of comparison questions. Accuracy is calculated according to ground truth.

As the results demonstrate, all models perform poorly on this comparison task, aligning with our expectations given the earlier findings that all models suffer from severe number hallucination. As a comparison, if the model only guesses an answer randomly, it will still achieve an accuracy of about 0.33. What's more, a model that always responds with a particular pattern (e.g., always answering 'A') will achieve an accuracy of even more than 0.40, so this result is very disappointing. It is natural that models struggle with comparison problems when they exhibit challenges in accurately counting numbers. However, inner and outer inconsistency in the results can still be observed.

Inner Inconsistency Inner inconsistency is similar to what occurs in binary classification questions. In this context, inner inconsistency refers to the model providing different answers to the same question. By comparing within the same prompt format, it is surprising to find that after c_1 and c_2 are flipped, the model's response sometimes changes (see one example in Appendix B in Table 8).

Our quantitative result shows that models generally output inconsistent answers for prompt I at a ratio of 25%, which is non-negligible. Prompt II also experiences this inconsistency even more severely (an example can be found in Appendix B in Table 9). For prompt II, the model responds with inconsistent answers of around 80%, which is disasterous. This inconsistency reveals two facts. Firstly, the order of input and the format of prompt have

[3] InstructBLIP cannot response meaningful answers to prompt II, so the corresponding result is not shown.

a notable influence on the model's output, making it challenging to evaluate the model's performance with multiple-choice questions. Secondly, this inconsistency also reflects the model's uncertainty about this question, which is a potential cause of number hallucination.

Outer Inconsistency Outer inconsistency in this context refers to such a phenomenon that the model's response to comparison task conflicts with its response to primal task. For instance, if the model responds there being 1 cat and 2 dogs in primal task but it answers there being more cats than dogs in comparison task, we say the model suffers from outer inconsistency.

Based on the definition above, we conduct a quantitative analysis, and it is not surprising to find that models also suffer from outer inconsistency severely on this comparison task. In fact, models only output 32% consistent answers averagely, which means under most circumstances the model does not completely agree with its own answer. In fact, a model that randomly responds to these questions will still achieve an outer consistency of about 33%, which is even higher than current results. Although this comparison task is more complicated and requires more complex reasoning, this low consistency is still absolutely a notable problem and should not be completely attributed to the complexity of this task.

In summary, through the analysis of results from these two perspectives, we observe severe inner and outer inconsistency problems. This inconsistency problem is not limited to any certain task or specific prompt. Instead, inconsistency appears everywhere without any regular pattern. These issues undermine the reliability of evaluation methods based on indirect questions. And, since this inconsistency represents the model's uncertainty and confusion towards this task, we believe inconsistency is a potential contributing factor to number hallucination.

4 Mitigating Number Hallucination

In this section, we propose a consistency training method to mitigate number hallucination.

4.1 Method

The most straightforward method to mitigate number hallucination is directly finetuning of LVLMs using counting questions. However, as analyzed in Section 3, since the model suffers from inner and outer inconsistency from different perspectives of this task, and this inconsistency is a potential cause of number hallucination, we would like to introduce a consistency training method to further mitigate number hallucination.

To elaborate further, rather than directly finetuning with the task defined in Section 2.1, combining the two relevant tasks mentioned in Section 3 may provide the model with a more comprehensive understanding of object numbers, leading to improved performance. Combining different tasks may also enhance the consistency among different perspectives, consequently reducing hallucinations.

Formally, given a training dataset $S = \{(i, c, n_c)\}$, the direct training method constructs a VQA dataset with questions formatted as follows. This method is called *Direct*.

Prompt: How many c *are there in this picture? Answer in a single number.*

Answer: n_c

The consistency training method constructs a VQA dataset with two question formats:

- I: *Prompt: There is(are)* m c *in this picture, is that correct? How many* c*s are there in this picture?*

  Answer: Yes(No). n_c.

- II. *Prompt: Which object is more in this picture,* c_1 *or* c_2*? How many* c_1 *are there in this picture? How many* c_2 *are there in this picture?*

  Answer: $c_1(c_2)$*(The number of* c_1 *and* c_2 *are the same in this picture.)* n_{c1}. n_{c2}.

We construct our consistency training dataset with three settings: using prompt I only(*Consistency(I)*), using prompt II only (*Consistency(II)*) and using both data generated by prompt I and prompt II (*Consistency(I+II)*).

By utilizing related tasks, we aim at improving the inner consistency of both tasks. We add the output of primal task to improve outer consistency. To be more detailed, since the model are trained to output correct answers to two different tasks simultaneously, the outer consistency is explicitly improved. We suppose by considering both inner and outer consistency, the model's performance will be further improved.

4.2 Experiment Setup

Dataset Construction We utilize the MSCOCO2014 (Lin et al., 2014) training

Model	Finetune Method	Marco-F1	Weighted-F1	MAE
LLaVA	None	0.186	0.288	1.84
	Direct	0.190	0.282	1.68
	Consistency(I)	0.198	**0.302**	1.70
	Consistency(II)	0.195	0.287	1.79
	Consistency(I+II)	**0.212**	0.301	**1.65**
InstructBLIP	None	0.201	0.300	2.35
	Direct	0.235	0.324	1.63
	Consistency(I)	0.235	0.346	1.99
	Consistency(II)	0.216	0.322	1.62
	Consistency(I+II)	**0.255**	**0.364**	**1.50**
MiniGPT-v2	None	0.195	0.293	2.05
	Direct	0.219	0.318	**1.61**
	Consistency(I)	0.225	**0.331**	1.69
	Consistency(II)	0.192	0.274	1.98
	Consistency(I+II)	**0.228**	0.320	**1.61**

Table 5: Results of finetuned models.

set as the foundational dataset and construct S in the same manner as described in Section 2.2. We refrain from performing k-uniform sampling on the training set to avoid bias and retain as much data as possible. For details about our dataset construction, please refer to Appendix C.

Model We continue the experiments on LLaVA, InstructBLIP and MiniGPT-v2.[4] For simplicity and efficiency, we freeze the vision encoder and LLM and only finetune the alignment part of each LVLM. Details of our hyper-parameters and models can be found in Appendix C.

4.3 Results and Analysis

The results are presented in Table 5. Some conclusions can be drawn from this result. Firstly, direct finetuning the model is useful for mitigating number hallucinations and provides a fine result.

The *Consistency(I+II)* method not only mitigates number hallucination compared with the unfinetuned models, but also outperforms the *Direct* method by 8% averagely (macro-F1), clearly demonstrating the effectiveness of our approach. This improvement further supports our assumption in Section 3 that inconsistency is a probable cause of number hallucination.

It is also worth mentioning that, even though the *Direct* method bears a more similar format with the testing scenario while our *Consistency(I+II)* method has a relatively different format, our *Consistency(I+II)* method still demonstrates better performance, which further reveals the importance of improving consistency.

Ablation Study We train with different consistency formats to fully reveal the performance of consistency method. *Consistency(I)* outperforms *Direct* on some metrics, which has already shown the effectiveness of consistency method. (For detailed discussions, please refer to Appendix D). However, *Consistency(II)* does not perform as well, which we believe is because comparison only is too difficult and far from our primal task, making it less effective when tested with primal task.

Further, after combining prompt I and II, *Consistency(I+II)* outperforms *Direct* on almost all metrics and also beats *Consistency(I)* and *Consistency(II)* on most metrics, which highlights the effectiveness of our consistency method and the importance of both perspectives. Although either perspective only does not offer good enough results, combining them leads to an excellent result. This observation also meets our assumption that adding more perspectives may further ease the inconsistency problems and resulting in better overall performance. Our method also bears several advantages such as focusing on task and data only, no need for finetuning the LLM part of LVLMs, etc. For details about the advantages of our method and a case study, please refer to Appendix D.

[4] LLaVA and LLaVA-L bear the same architecture, so we only finetune LLaVA for simplicity.

5 Related Works

5.1 Large Vision-Language Models

With the success of large language models (LLMs), there has been a surge in efforts to incorporate additional modalities into LLMs, leveraging their powerful recognition capabilities. BLIP-2 (Li et al., 2023a), for instance, introduces the Q-former to align image input into tokens suitable for utilization by LLMs. LLaVA (Liu et al., 2023b) and LLaVA-v1.5 (Liu et al., 2023a) employ visual instruction tuning and leverage simple linear layers or multi-layer perceptrons to map image output into the text feature space. InstructBLIP (Dai et al., 2023) achieves instruction following abilities by performing instruction tuning on the BLIP-2 backbone. Notably, MiniGPT-4 (Zhu et al., 2023) and MiniGPT-v2 (Chen et al., 2023a) are also powerful LVLMs that exhibit strong performance across various vision-language tasks. Among all open-source and closed-source LVLMs, GPT-4V (OpenAI et al., 2023) retains its status as the most powerful one.

Hallucination is an extensively discussed phenomenon in both traditional vision-language models and large language models and numerous studies have delved into the examination of hallucinations in large vision-language models (LVLMs). The most extensively discussed form of hallucination is object hallucination, centering around the determination of whether a specific object exists in a given image. In the context of caption-based evaluation, CHAIR (Rohrbach et al., 2018) stands out as a classic metric that compares objects mentioned in the caption with those present in the provided image. POPE (Li et al., 2023b), on the other hand, employs a binary classification approach by directly asking the model whether a particular object exists in the provided image. Zhou et al. (2023) conducted an analysis of factors influencing object hallucination and introduced LURE based on their findings. Leng et al. (2023) proposed visual contrastive decoding to mitigate object hallucinations, while Zhai et al. (2023) suggested a switch-based method to address this issue. However, these studies predominantly focus on the mere existence of objects, which may not be sufficient for gauging whether the model truly comprehends the image.

Beyond object hallucination, there are also studies addressing more generalized forms of hallucinations. Wang et al. (2023a) introduced LLMs to aid in the detection and evaluation of hallucinations in captions. Gunjal et al. (Gunjal et al. (2023)) proposed a dataset and reward model to assess and mitigate hallucinations in captions. RAH (Chen et al., 2023b) and FGHE (Wang et al., 2023b) are two datasets similar to POPE that can be employed for evaluating fine-grained hallucinations. However, many caption-based hallucination evaluation methods require a lot of training modules, so the performance and stability of them remain uncertain due to the complexity of captions and training procedure. POPE-like evaluation metrics rely on binary classification questions and might not fully capture the model's ability, making them insufficient for quantitative comparisons.

5.2 Counting Task in VQA

There have been studies specifically addressing the task of counting objects in visual question-answering. Earlier works such as Lempitsky and Zisserman (2010) and Segui et al. (2015) laid some groundwork in this area. Zhang et al. (2018) proposed a counting module that can be integrated into other visual question-answering models. Additionally, Liu et al. (2020) introduced a blockwise counting method to tackle issues like inaccurate generated regression targets and imbalanced samples, which were significant challenges in previous counting methods. However, these methods often require specific designs for the task and modifications to the entire model architecture, which may be insufficient and inefficient in the context of large models nowadays.

6 Conclusion

In this paper, we introduce a new form of object hallucination - number hallucination, and present a 20k dataset along with evaluation metrics and results to assess number hallucination in LVLMs. Our findings reveal that all LVLMs suffer severely from number hallucinations. Through our analysis, we demonstrate that binary classification questions are insufficient evaluation methods. We further explore inner and outer inconsistency from two perspectives related to number hallucination and identify severe inner and outer inconsistency despite task or prompt, highlighting inconsistency as a potential cause of number hallucination. Building on this analysis, we propose a consistency training method to mitigate number hallucination and observe an averagely 8% improvement. This method is extendable to almost any LVLM and applicable to various types of hallucination.

7 Limitations

There are several limitations in our work. First of all, we only use one dataset for discussion instead of using multiple datasets. Secondly, under the constraint of cost we do not perform a quantitative evaluation on GPT-4V, which is a pity. Lastly, we mainly focus on VQA tasks and ignore captioning tasks for its complexity, while we do believe that captioning tasks are still worth discussing.

8 Ethics Statement

We do not perform any specified selection on data or models. All parts of our work are done evenly and automatically. And since we use open-source dataset (MSCOCO) and model weights (LVLMs) just consistently with their intended use (object detection and language generation), we believe there are no ethical problems in our work. ChatGPT is used only to assist writing (grammar checking) in this paper, which we believe is totally acceptable.

References

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023a. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*.

Zhiyang Chen, Yousong Zhu, Yufei Zhan, Zhaowen Li, Chaoyang Zhao, Jinqiao Wang, and Ming Tang. 2023b. Mitigating hallucination in visual language models with visual supervision. *arXiv preprint arXiv:2311.16479*.

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

Anisha Gunjal, Jihan Yin, and Erhan Bas. 2023. Detecting and preventing hallucinations in large vision language models. *arXiv preprint arXiv:2308.06394*.

Victor Lempitsky and Andrew Zisserman. 2010. Learning to count objects in images. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc.

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*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023b. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

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

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning.

Liang Liu, Hao Lu, Haipeng Xiong, Ke Xian, Zhiguo Cao, and Chunhua Shen. 2020. Counting objects by blockwise classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(10):3513–3527.

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish

Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4035–4045, Brussels, Belgium. Association for Computational Linguistics.

Santi Segui, Oriol Pujol, and Jordi Vitria. 2015. Learning to count with deep object features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Junyang Wang, Yiyang Zhou, Guohai Xu, Pengcheng Shi, Chenlin Zhao, Haiyang Xu, Qinghao Ye, Ming Yan, Ji Zhang, Jihua Zhu, et al. 2023a. Evaluation and analysis of hallucination in large vision-language models. *arXiv preprint arXiv:2308.15126*.

Lei Wang, Jiabang He, Shenshen Li, Ning Liu, and Ee-Peng Lim. 2023b. Mitigating fine-grained hallucination by fine-tuning large vision-language models with caption rewrites. *arXiv preprint arXiv:2312.01701*.

Bohan Zhai, Shijia Yang, Chenfeng Xu, Sheng Shen, Kurt Keutzer, and Manling Li. 2023. Halle-switch: Controlling object hallucination in large vision language models. *arXiv e-prints*, pages arXiv–2310.

Yan Zhang, Jonathon Hare, and Adam Prügel-Bennett. 2018. Learning to count objects in natural images for visual question answering. *arXiv preprint arXiv:1802.05766*.

Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. 2023. Analyzing and mitigating object hallucination in large vision-language models. *arXiv preprint arXiv:2310.00754*.

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 Details About Dataset Construction

When constructing dataset, we only retain data (i, c, n_c) with $n_c > 0$. There are two main considerations for this choice. First of all, there are many objects that do not appear in a given image and retaining all these questions is too expensive. If we perform k-uniform sampling similarly, this may result in randoml selection, which is insufficient for reflecting the model's performance according to (Li et al., 2023b). More importantly, we conduct an experiment on LLaVA with randomly selected (i, c, n_c) with $n_c = 0$ and find out that LLaVA correctly answers 95% of these questions, which meets the its high POPE evaluation result (Liu et al., 2023a). Based on these considerations, we only retain data (i, c, n_c) with $n_c > 0$ to focus on the number of an existing object.

We present double k-uniform sampling in algorithm 1. Adjusting the value of k allows us to strike a balance between achieving a more uniform distribution and retaining a larger dataset. In the subsequent discussion, we opt for $k = 50$, resulting in a total of 20,200 data points. The distribution of our dataset is shown in Table 6.

Algorithm 1 Double k-Uniform Sampling

Input dataset $S = \{(i, c, n_c)\}$ and k.
Randomly shuffle the dataset S.
Initialize $m(c, n) \leftarrow 0, S^{'} = \phi$
for $j = 1, 2..., |S|$ **do**
  if $m(c_j, n_{cj}) > k$ **then**
    continue
  end if
  $S^{'} \leftarrow S^{'} \cup \{(i_j, c_j, n_{cj})\}$
  $m(c_j, n_{cj}) \leftarrow m(c_j, n_{cj}) + 1$
end for
Output $S^{'}$

Answer	Percentage
1	19.8%
2	18.7%
3	15.1%
4	11.9%
5	8.0%
6	6.1%
7	4.8%
8	3.7%
9	3.1%
≥ 10	8.8%

Table 6: The distribution of the answers of the questions

As can be seen from the Table 6, though k-uniform sampling is performed, questions with answer $\geq$ 10 is still too less. Therefore to avoid randomness, when calculating F-scores, questions with answers $\geq$ 10 are considered one category.

B Inconsistency Examples

As is mentioned, inconsistency is a common problem. Here are some typical responses containing inconsistency.

Particularly, Table 9 are two responses generated by MiniGPT-v2. As can be seen, they bear the same input image and prompt with similar meanings and the only difference between them is the order of the objects (dog and bicycle). However, model responds with contradictory answers. In the first case, model correctly identifies there being more bicycles than dogs. But in the second case, the model continues to answer 'B', which means it believes there are more dogs than bicycles. This inconsistency is a giant problem because if we only see the first case, we may happily assume that the model correctly understands this picture, while only with both cases can we observe this dilemma.

Prompt: There are 2 motorcycles in this picture, is that correct? Answer yes or no. Answer: Yes	Prompt: There is 1 motorcycle in this picture, is that correct? Answer yes or no. Answer: Yes

Table 7: Example of model answering 'yes' to contradictory questions.

Prompt: Which object has a larger number in this picture, backpack or dog? If they have the same number, answer same. Answer: Dog	Prompt: Which object has a larger number in this picture, dog or backpack? If they have the same number, answer same. Answer: Backpack

Table 8: Example of model's respond changing after the order of objects changes.

Prompt: Please select the correct option according to the given picture. A. There are more dogs than bicycles in this picture. B. There are more bicycles than dogs in this picture. C. The number of dogs and bicycles are the same in this picture. Answer: B	Prompt: Please select the correct option according to the given picture. A. There are more bicycles than dogs in this picture. B. There are more dogs than bicycles in this picture. C. The number of bicycles and dogs are the same in this picture. Answer: B

Table 9: Example of model's response changing after the order of objects changes.

C Dataset, Models and Hyper-Parameters of Finetuning

Dataset We obtain training set S in the same way as mentioned in Section 2.2. This provides us with 233k pieces of data. For *Direct* method, all these data are used by directly asking like Section 2.1.

For *Consistency(I)*, all these data are kept with prompt I, which provides us with 233k pieces of data similarly. As for *Consistency(II)*, 100k pieces of data are randomly selected with prompt II. By combining these two datasets, we get a dataset with 333k pieces of data, corresponding to *Consistency(I+II)*. Note that all these consistency data are automatically constructed from the whole original MSCOCO2014 training set.

Models For LLaVA, the alignment part is a two-layer MLP. For MiniGPT-v2, the alignment part is a linear layer. For InstrcutBLIP, though the whole alignment part is a Q-former, we perform an experiment and observe that only finetuning the linear layer after Q-former instead of the whole Q-former offers a better performance on the direct training method, so we also only finetune the linear layer after Q-former in InstructBLIP.

Hyper-Parameter We finetune each model for three epochs, which costs from 18 to 30 hours depending on different models and methods. The remaining hyper-parameters generally remain unchanged as those provided in the official repository of each model. Batch size is modified to fit the model into our computing resources. Gradient accumulation is used to imitate larger batch size.

Learning Rate	1e-5
Weight Decay	0.05
Batch Size	16
GPUs	4
DeepSpeed	zero1
Seed	42

Table 10: Main hyper-parameters used for tuning LLaVA.

Learning Rate	1e-5
Weight Decay	0.05
Batch Size	8
GPUs	2
Seed	42

Table 11: Main hyper-parameters used for tuning InstructBLIP.

Learning Rate	1e-5
Weight Decay	0.05
Batch Size	24
GPUs	3
Seed	42

Table 12: Main hyper-parameters used for tuning MiniGPT-v2

D Discussion about Our Method

Discussions We observe an interesting phenomenon that *Consistency+I* often offer better weighted-F1 and worse MAE, while *Consistency(I+II)* may bear a littlt bit worse weighted-F1 with much better MAE than *Consitency(I)*. We believe this phenomenon is because of the distribution of our training set. Since we do not perform double k-uniform sampling on our training set as mentioned, there are much more data with small answers rather than large answers, making the model more confident about generating smaller answers after introducing consistency with binary

classification questions. However, when comparison question is introduced, the model is forced to focus on the whole image instead of trying to bypass by generating smaller answers, which result in a small drop in weighted-F1 while an increase in macro-F1 and MAE. This observation also highlights the importance of using these three metrics instead of one particular metric. As can be seen, using three metrics offers a more comprehensive understanding of the model's performance.

Advantages of Our Method Our method offers several advantages. Firstly, it solely focuses on data construction and is independent of specific model architecture. This model-agnostic characteristic makes it extendable to almost any large vision-language model (LVLM). As can be seen from our results, this method improves performance across all tested LVLMs.

Secondly, this method operates without the need to finetune the large language model (LLM) part of the model. Some mitigating methods necessitate finetuning the entire alignment and LLM part of the LVLM, which can be expensive and less efficient. In contrast, our method works with relatively small trainable parts, making it a more cost-effective and efficient solution.

Thirdly, this method may be extended to other hallucination-mitigating tasks. From a methodological perspective, most hallucinations typically involve various perspectives, thus consistency can be analyzed, and similar training methods can be applied.

Prompt: How many bicycles are there in this picture? Answer in a single number.

Answer of *Direct*:	Answer of *Consistency(I+II)*:
3.	5.

Table 13: A successful example of consistency method. The gold answer is 5.

Case Study Table 13 shows a successful example of our consistency method. This is also a case that we use to illustrate inconsistency problem in Appendix B, which further prove the effectiveness of introducing consistency into training. As can be seen, with *Direct* method the model fails to identify the correct number of bicycles, while after introducing *Consistency(I+II)* method the model responds the correct answer. We believe this improvement is led by our consistency method forces the model to think more carefully and comprehensively, and thus reduces uncertainty and confusion.