OPERA: Alleviating Hallucination in Multi-Modal Large Language Models via Over-Trust Penalty and Retrospection-Allocation

Qidong Huang[1,2,*], Xiaoyi Dong[2,3,†], Pan Zhang[2], Bin Wang[2], Conghui He[2], Jiaqi Wang[2],
Dahua Lin[2], Weiming Zhang[1,†], Nenghai Yu[1]
[1]Anhui Province Key Laboratory of Digital Security, University of Science and Technology of China
[2]Shanghai AI Laboratory [3]The Chinese University of Hong Kong
{hqd0037@mail., zhangwm@, ynh@}ustc.edu.cn {xydong@, dhlin@}ie.cuhk.edu.hk
{zhangpan@, wangbin@, heconghui@}pjlab.org.cn wjqdev@gmail.com

Abstract

*Hallucination, posed as a pervasive challenge of multi-modal large language models (MLLMs), has significantly impeded their real-world usage that demands precise judgment. Existing methods mitigate this issue with either training with specific designed data or inferencing with external knowledge from other sources, incurring inevitable additional costs. In this paper, we present **OPERA**, a novel MLLM decoding method grounded in an **O**ver-trust **P**enalty and a **R**etrospection-**A**llocation strategy, serving as a nearly **free lunch** to alleviate the hallucination issue without additional data, knowledge, or training. Our approach begins with an interesting observation that, most hallucinations are closely tied to the knowledge aggregation patterns manifested in the self-attention matrix,* i.e., *MLLMs tend to generate new tokens by focusing on a few summary tokens, but not all the previous tokens. Such partial over-trust inclination results in the neglecting of image tokens and describes the image content with hallucination. Based on the observation, OPERA introduces a penalty term on the model logits during the beam-search decoding to mitigate the over-trust issue, along with a rollback strategy that retrospects the presence of summary tokens in the previously generated tokens, and re-allocate the token selection if necessary. With extensive experiments, OPERA shows significant hallucination-mitigating performance on different MLLMs and metrics, proving its effectiveness and generality. Our code is at: https://github.com/shikiw/OPERA.*

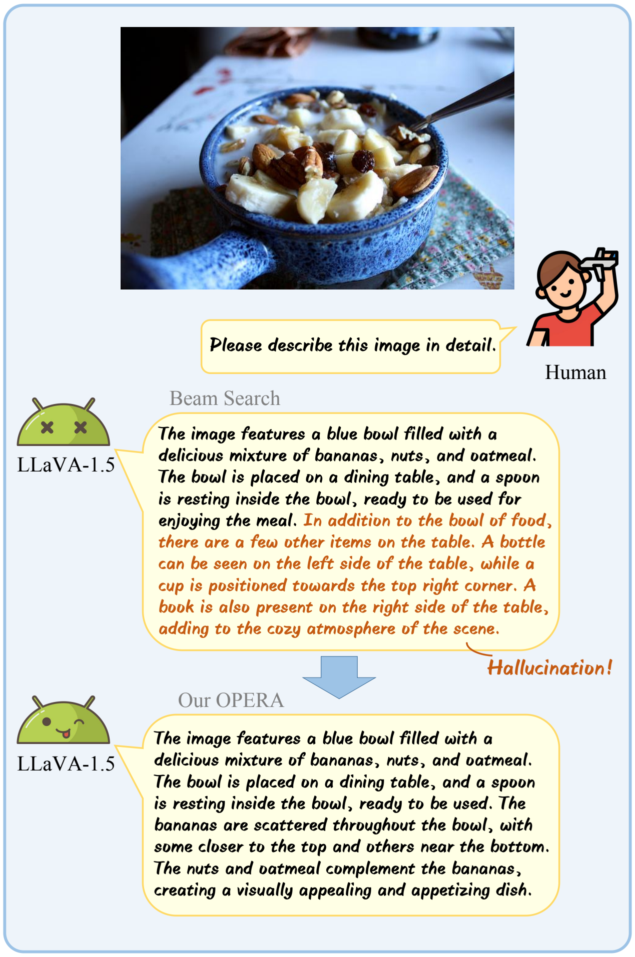

Please describe this image in detail.
Human
Beam Search
LLaVA-1.5
The image features a blue bowl filled with a delicious mixture of bananas, nuts, and oatmeal. The bowl is placed on a dining table, and a spoon is resting inside the bowl, ready to be used for enjoying the meal. In addition to the bowl of food, there are a few other items on the table. A bottle can be seen on the left side of the table, while a cup is positioned towards the top right corner. A book is also present on the right side of the table, adding to the cozy atmosphere of the scene.
Hallucination!
Our OPERA
LLaVA-1.5
The image features a blue bowl filled with a delicious mixture of bananas, nuts, and oatmeal. The bowl is placed on a dining table, and a spoon is resting inside the bowl, ready to be used. The bananas are scattered throughout the bowl, with some closer to the top and others near the bottom. The nuts and oatmeal complement the bananas, creating a visually appealing and appetizing dish.

Figure 1. OPERA's performance on reducing hallucinations.

1. Introduction

Recent advancements in multi-modal large language models (MLLMs) [1, 5, 9, 10, 30, 31, 44, 48] has greatly elevated general-purpose foundation models to unprecedented levels. These models enable users to interact using images as input, facilitating free-flowing communication based on the content of these images. The impressive abilities of

[*] Work done during an internship in Shanghai AI Laboratory.
[†] Corresponding authors.

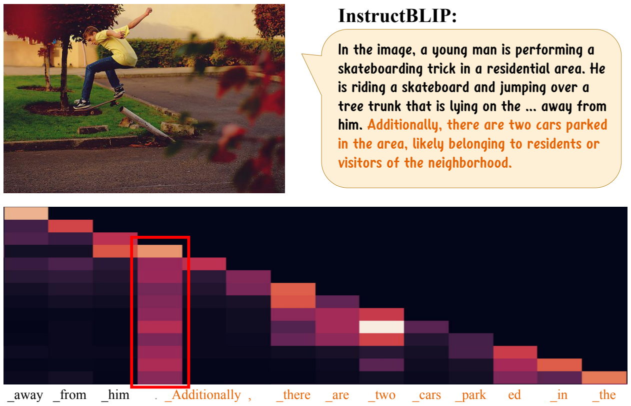

InstructBLIP:
In the image, a young man is performing a skateboarding trick in a residential area. He is riding a skateboard and jumping over a tree trunk that is lying on the ... away from him. Additionally, there are two cars parked in the area, likely belonging to residents or visitors of the neighborhood.
_away _from _him . _Additionally , _there _are _two _cars _park ed _in _the

Figure 2. A case of relationship between hallucinations and knowledge aggregation patterns. Hallucinations are highlighted.

MLLM allows it to be adept at a variety of vision tasks [2, 25, 45], meanwhile easily handling some complex content comprehension [23] or generation [4, 15].

Notwithstanding their remarkable versatility, MLLMs also grapple with a significant challenge known as the "hallucination" problem. Specifically, MLLMs often hallucinate incorrect statements to the user-provided image and prompts, *e.g*., producing irrelevant or nonsensical responses, indentifying inaccurate objects in terms of colors, quantities and locations that do not exist in the image. This flaw poses substantial risks for practical applications of MLLMs to become a trustworthy assistant. For instance, in model-assisted autonomous driving scenarios, such misinterpretations of road scene images may lead to wrong judgments of system and serious traffic accidents.

Various approaches [29, 40, 42, 47] have been proposed to reduce hallucinations in MLLMs. While these methods incur substantial additional costs, including the annotation budget for extra instruction data for training [29], the integration of external knowledge or models, *etc*.

In this paper, we delve into the challenge of mitigating MLLMs' hallucination during inference, without introducing additional data, models, or knowledge. Our investigation commences with a noteworthy '**partial over-trust**' observation found while visualizing self-attention maps for decoded sequences. As illustrated in Figure 2, we discern a recurring pattern where the inception of many hallucinated contents aligns with the subsequent tokens generated after a columnar attention pattern. Notably, these columnar attention patterns often manifest on tokens that lack substantial informativeness, *e.g*., full stop or quotation marks. Intuitively, this peculiarity reveals a weird fact that, a token exhibiting a columnar attention pattern typically possesses limited information, yet exerts a pronounced influence on the prediction of all subsequent tokens. Moreover, as shown in Figure 3, we find that most of the subsequent contents contain reasoning or hallucinations.

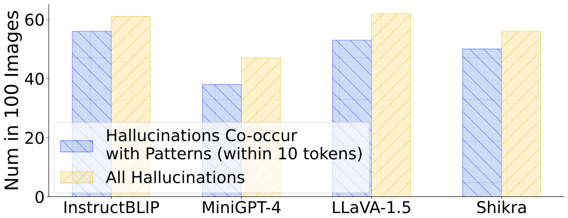
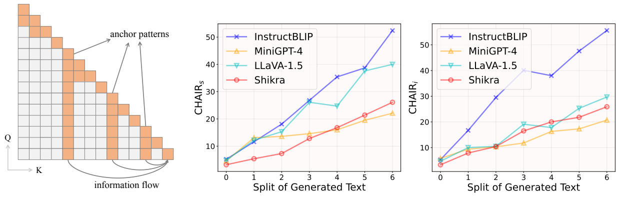


Figure 3. Hallucinations often start within the first 10 tokens after knowledge aggregation patterns.



(a) (b) (c)

Figure 4. (a) The aggregation pattern is consistent with recent 'anchor token' observation. (b), (c) show the increasing CHAIR scores (more hallucinations) on 5,000 randomly selected MSCOCO images when more anchor tokens appear in the context.

'Aggregation pattern' seems to be the nature of LLM. We hypothesize that such tokens serve as summary tokens, which aggregate the crucial knowledge from previous tokens in the sequence and guide the subsequent tokens generation. Our observation is consistent with the recent 'anchor token' [41] observation in the NLP area, which finds the LLM tends to aggregate previous information on a few anchor tokens at shallow layers and predict the next token based on these anchors at the deep layer (Figure 4(a)).

'Aggregation pattern' leads to hallucination of current MLLMs. Current MLLMs usually put the vision tokens at the beginning of the sequence, and they are expected to focus on the vision tokens and provide an precise understanding. However, as the generated text goes longer, it will be easier for vision information to be attenuated during the transmission of information between summary tokens (a single summary token can not remember the dense and rich information given by the whole context). In detail, the subsequent tokens may ignore the forehead image tokens and over-trust the closer summary tokens via their stronger attention attended, leading to hallucinations raised by the model bias, *e.g*., hallucinating "cars" based on the "road" mentioned in the previous sentence. In other words, the more summary tokens appear, the more easily MLLM hallucinations are induced. To prove it, we split the long responses of MLLMs based on the position of summary tokens, and calculate the CHAIR scores for different splits separately. As shown in Figure 4(b)(c), the CHAIR score shows a clear **positive relation** with the split number of the generated text, *i.e*., more hallucinations are generated when more summary tokens appear in the context, manifested as the co-occurrence of them.

To alleviate the partial over-trust issue, we present OPERA, a novel MLLM decoding approach grounded in an **O**ver-trust **Pe**nalty and a **R**etrospection-**A**llocation strategy. The over-trust penalty introduces a weighted score for the candidate selection step in the Beam Search [3, 16, 37], so that the candidate with an over-trust pattern will have lower priority to be selected. Specifically, for each decoding token, we investigate the local window segmented on the self-attention map of the decoded sequence, and devise a column-wise metric to calculate the intensity of knowledge aggregation patterns. This metric produces a value that indicates the over-trust degree between in-window tokens and the summary tokens. It is naturally incorporated with the model logits predicted for the next token in the Beam Search and penalizes the appearance of over-trust patterns. Further, considering the hysteresis of the appearance of the knowledge aggregation pattern, the hallucination may exist in all the candidates when it can be observed. We propose a retrospection-reallocation strategy to help the decoding process roll back to the position of the summary token and reselect better candidates that can avoid such a pattern. Such retrospection is triggered when the location overlap of the maximum of in-window penalty scores reaches a threshold.

With extensive experiments on benchmarks and hallucination metrics, along with GPT-4/GPT-4V assessments, OPERA demonstrates the generalized hallucinations-reducing performance on various MLLM models. Our contributions can be summarized as follows:

- Our OPERA alleviates the MLLMs' hallucination issue during inference, without introducing any external data, knowledge, or additional training.
- We reveals the appearance of hallucinations and over-trust patterns, and propose a penalty-based decoding method equipped with retrospection-reallocation strategy.
- Extensive evaluation including GPT assessments prove the superior performance of OPERA, which serves as a nearly free-lunch to mitigate hallucinations.

2. Related Work

2.1. Multi-Modal Large Foundation Models

Recent progresses of computational resources has greatly facilitated the research into large-scale foundational models incorporated with multi-modal learning. Powered by open-sourcing large language models such as LLaMA [38, 39] and Vicuna [7], MLLMs [1, 6, 9, 18–20, 31, 48] understand and generate diverse content in a more comprehensive way by integrating information from different modalities, such as text, images, and audio. The series of CLIP and BLIP well aligns the text features and image features. LLaVA [31], InstructBLIP [9] and MiniGPT-4 [48] take a step forward in this field, allowing users to interact with these intelligence with images and texts as prompts. All of them share the same two training phases, *i.e.*, pre-trained feature alignment and instruction fine-tuning, to help the model to comprehend the format of instruction input. Shikra [5] incorporates grounding data and teaches the model to understand the grounding knowledge in the given images. All of aforementioned MLLM models suffer from severe hallucination problems. Consequently, we mainly conduct the experiments on these four models in our paper.

2.2. Hallucination in Large Foundation Models

The hallucination [21, 43] refers to the generation of text that is either irrelevant, factually incorrect, or nonsensical in the given context, which is quite severe in current large foundation models. This issue can arise due to overfitting to specific patterns in the training data, lack of understanding of real-world facts, or an inability to effectively contextualize the given input. The primary concern regarding hallucination in LLMs is the factual accuracy of generated content, *i.e.*, conflicting with world knowledge or common sense. In MLLMs, the primary worry centers around faithfulness, *i.e.*, assessing whether the generated answers conflict with user-provided images. Researches on mitigating current LLMs' hallucination issues often focuses on several aspects, including refining the training process, using larger and more diverse datasets [24], or implementing post-training evaluation [11] and correction mechanisms [33, 34]. While for MLLMs, relevant researches are still quite few [29, 42, 47]. However, most of these countermeasures have a large drawback that, they either introduce large quantities of extra data, or resort to more powerful external models or knowledge. Compared with them, our OPERA serves as nearly free lunch for alleviating the hallucination issue, without incurring extra training, data, or knowledge.

2.3. Decoding Strategy in Language Models

Decoding strategies in language models are crucial for determining how these models generate text. They play a pivotal role in shaping the output's quality, relevance, and coherence. Greedy Decoding simply selects the most likely next word at each step. While fast and computationally efficient, greedy decoding often leads to repetitive and less varied text. Beam Search [3, 16, 37] is a more sophisticated approach, beam search keeps track of a predefined number of hypotheses at each step, expanding on them to find a more optimal sequence. Top-k Sampling [12] adds randomness to the generation process by randomly selecting from the top-k likely next words, introducing diversity in the output but can sometimes produce less coherent results. Top-p (Nucleus) Sampling [17] is an evolution of Top-k, Nucleus sampling considers a dynamic number of words that cumulatively reach the probability p. This method provides a balance between randomness and relevance, often leading to more coherent and interesting outputs than Top-k sampling. DoLa [8] decoding is a recently proposed decoding method that aims to mitigate the hallucinations in MLLMs, which contrasts the logits of mature layer and pre-mature layers and rescale the increments as the output. In this paper, we compare our proposed OPERA with these common decoding strategies, focusing on the performance on the hallucination issues of MLLMs.

3. Method

In the following, we first formulate the generation procedure of the MLLMs for the easy understanding of our OPERA, then introduce the calculation of the proposed Over-Trust Logit Penalty and Retrospection-Allocation Strategy respectively.

3.1. Formulation of MLLMs Generation

The generation procedure of LLMs could be parsed into three parts: input formulation, model forward, decoding.

Input Formulation. The input of MLLMs contains both image and text. Putting aside the specific architecture difference, the MLLMs commonly use a vision encoder to extract visual tokens from the raw images, and map them into the LLMs' input space with a cross-modality mapping module. The mapped visual tokens are used as part of the LLM input, along with the text input. We denote the visual tokens as $\mathbf{x}^v = \{x_0, x_1, \dots, x_{N-1}\}$. Here N is the length of the visual tokens and it is a fixed number in most cases. Correspondingly, the input text is tokenized with the tokenizer and we denote it as $\mathbf{x}^p = \{x_N, x_{N+1}, \dots, x_{M+N-1}\}$. The image and text tokens are concatenated as the final input sequence and we denote it as $\{x_i\}_{t=0}^{T-1}$ that $T = N + M$.

Model Forward. The MLLM is trained in an auto-regressive manner with a causal attention mask, each token predicts its next token based on previous tokens, formally:

$$\begin{aligned} \mathbf{h} &= \mathrm{MLLM}(\mathbf{x}_i) \\ \mathbf{h} &= \{h_0, h_1, \dots, h_{T-1}\} \end{aligned} \tag{1}$$

where $\mathbf{h}$ is the output hidden states of MLLM's last layer.

Next, MLLMs use a vocabulary head $\mathcal{H}$ to project the hidden states $\mathbf{h}$ and get the logits (or probabilities) for the next token prediction, formally:

$$p(x_t|x_{<t}) = \mathrm{SoftMax}[\mathcal{H}(h_t)]_{x_t}, \quad x_t \in \mathcal{X}, \tag{2}$$

where we use $x_{<t}$ to simplify the sequence $\{x_i\}_{i=0}^{t-1}$ and $\mathcal{X}$ means the whole vocabulary set.

Decoding. Based on the logits $p(x_t|x_{<t})$, there are several decoding strategy developed, including Greedy Decoding, Beam Search, DoLa, *etc*. The decoded token is concatenated to the last of the original input text for the next-round generation, until the generation is ended.

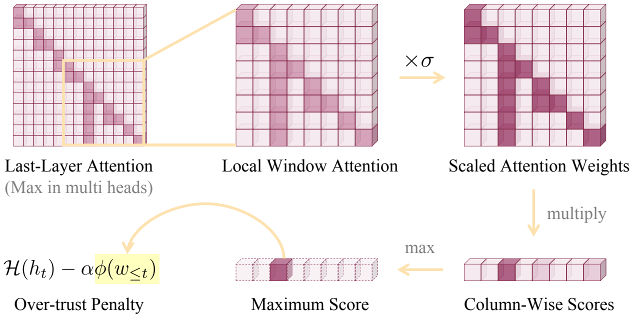


Figure 5. The scheme of calculating the proposed over-trust penalty term. We first cut out a local window on the self-attention map, then we scale up the values and conduct the column-wise multiplication to get a score vector, finally we choose the maximum score as the penalty term.

Our OPERA is based on the Beam Search [3, 16, 37], which is a accumulated-score-based decoding strategy. Briefly, With a given beam size N_{beam}, the Beam Search keeps N_{beam} candidate sequences, where each candidate is a decoded sequence $\mathbf{x}^{N_{beam}}$ with a beam score. When decoding token x_t, each candidate hypothesis will select N_{beam} candidate tokens based on the Top-N_{beam} probabilities in the logits. And finally, the decoding procedure will output the hypothesis wins the best beam score.

3.2. Over-Trust Logit Penalty

As we analyzed in Sec.1, there exists a high-probability co-currence between the hallucination and the knowledge aggregation patterns. However, such pattern has a significant **hysteresis**, *i.e.*, the patterns can not be immediately observed when the corresponding token is decoded, but after several subsequent tokens been decoded, and the hallucination may already occurred. In response to the hysteresis, we propose 'Over-Trust Logit Penalty', an *accumulative penalty weighted in the beam score*, which influences the selection of both the current token and the candidate sequence. A candidate sequence accumulated with a large penalty will have a lower priority to be selected so that the output with hallucinations will be possibly omitted.

In practice, we investigate a local window on the self-attention weights and leverage column-wise product to calculate the metric values. Denote the current generated sequence as $\{x_i\}_{i=0}^{t-1}$ and their casual self-attention weights $\{\omega_{t-1,j}\}_{j=0}^{t-1}$ paid on the next token prediction, in which the weights can be depicted by softmax result as $\omega = \mathrm{SoftMax}(\frac{QK^\top}{\sqrt{D}})$ and Q, K, D denote query feature, key feature, feature dimension respectively. We consider to gather all of previous self-attention weights in a local window for characterizing the knowledge pattern, *i.e.*, the local window attention is defined as

$$\mathbf{W}_{t-1}^k = \{\mathbf{w}^i\}_{i=t-k}^{t-1}, \quad \text{s.t. } \mathbf{w}^i = \{\omega_{i,j}\}_{j=t-k}^{i}, \tag{3}$$

where k denotes the size of local window we cropped on the attention map, $\omega_{i,j}$ means the attention weight assigned by the j^{th} token to the i^{th} token. There are two points should be clarified: 1) our window does not involve the attention weights of image tokens or prompt tokens because we only concentrate on the knowledge aggregation patterns on generated tokens, *i.e.*, $t - k \geq N + M$. 2) we select the maximum weight in attention heads since it usually indicates the strong confidence of models ,and re-normalize the values .

With the local window attention weights $\mathbf{W}_{t-1}^{k}$, we can calculate upon a simple metric to describe the size of the knowledge aggregation pattern. Specifically, we first do some preprocess on $\mathbf{W}_{t-1}^{k}$, including filling the upper triangle of the matrix with zeros and scaling up the attention values as the values are usually too small, *i.e.*,

$$\mathbf{W}_{t-1}^{k} \triangleq \{\mathbf{w}^{i}\}_{i=t-k}^{t-1}, \quad \text{s.t. } \mathbf{w}^{i} = \{\sigma\omega_{i,j}\}_{j=t-k}^{t-1}, \tag{4}$$

where $\{\omega_{i,j}\}_{j=i+1}^{t-1}$ are zeros and σ is a configurable scaling factor.

As illustrated in Figure 5, we then conduct the column-wise multiplication on the lower triangle of the attention matrix and obtain a vector of column-wise scores. Intuitively, the larger score indicates the stronger pattern that exists at the corresponding location. Thus, we select the maximum value of the column-wise score vector as the characteristic of knowledge aggregation patterns. Formally,

$$\phi(\omega_{<t}) = \prod_{i=c}^{t-1} \sigma\omega_{i,c}, \quad \text{s.t. } c = \underset{t-k \leq j \leq t-1}{\arg\max} \prod_{i=j}^{t-1} \sigma\omega_{i,j}. \tag{5}$$

Until now, we have an salient metric to detect the occurring of knowledge aggregation patterns within the local window. With the concern of calculation efficiency and the penalty should not bias the model to unreasonable output, we choose the top-N_{can} in the logit of each beam to consist a candidate set $\mathcal{Y}$, where $|\mathcal{Y}| = N_{can} * N_{beam}$ and N_{beam} is the number of beams. In this way, we limit the prediction within the candidate set and incorporate $\phi(w_{\leq t})$ with the model logits to predict the next token, *i.e.*,

$$p(x_t|x_{<t}) = \text{Softmax}[\mathcal{H}(h_t) - \alpha\phi(w_{\leq t})]_{x_t}, \quad \text{s.t. } x_t \in \mathcal{Y}, \tag{6}$$

where $w_{\leq t}$ simplifies all of attention weights obtained by feeding forward the sequence $\{x_0, x_1, \ldots, x_t\}$.

3.3. Retrospection-Allocation Strategy

With the over-trust logit penalty, we can successfully detect the occurrence of patterns after several subsequent tokens are generated. Normally, the penalty term is able to penalize the candidates which have knowledge aggregation patterns, and encourage other candidates to be predicted. While there still exists a few cases that *all of the candidates get penalized and the hallucination already occurred.*

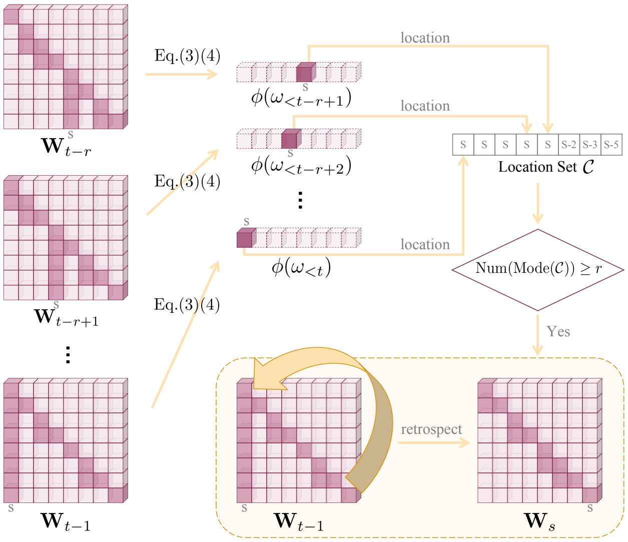


Figure 6. The scheme of the proposed Retrospection strategy. We compute the maximum value coordinates of the past several token's column-wise scores and check if the overlap time is larger than r. If yes, we retrospect the decoding procedure and reselect the next token x_{s+1}.

This case motivates us to rethink the origin of such aggregation patterns: it is caused by the first few subsequent tokens over-trusting the summary token, and the penalty failed to correct them. So an intuitive while aggressive idea is that the pattern will be greatly weakened if we could exclude the tokens that lead to hallucination and re-choose the proper first few tokens after the summary token.

To this end, we propose the **R**etrospection-**A**llocation strategy. Specifically, when the decoding procedure encounters the knowledge aggregation pattern and the hallucination is inevitable, it rolls back to the summary token and selects other candidates for the next token prediction except for the candidates selected before. Empirically, the condition of decoding retrospection is designed as the location overlap of the maximum value in column-wise scores that corresponds to several consecutive tokens, where we manually set the threshold counts as r. Rather than the maximum value that varies between different models, location counting is a much more robust and general metric for the decision.

The whole retrospection process is illustrated in Figure 6. Based on Sec. 3.2, we can easily derive the location coordinate c of the maximum score via Eq. (5). Consequently, we can obtain the location coordinate set of several recently decoded tokens $x_{t-l}, \ldots, x_{t-1}$, *i.e.*,

$$\mathcal{C} = \{c | c = \underset{t-k \leq j \leq z}{\arg\max} \prod_{i=j}^{z} \sigma\omega_{i,j}, z \in [t-l, t-1]\}, \tag{7}$$

where $l > r$ should be specified. We set $l = k$ by default.

Given a sequence $\{x_0, x_1, \ldots, x_{t-1}\}$ and its recent location coordinate set $\mathcal{C}$, we can easily check whether the coordinates are consistent. Formally, the overlap times can be calculated by

$$N_{overlap} = \sum_{c \in \mathcal{C}} \mathbb{1}_{c=s}, \quad \text{s.t. } s = \text{Mode}(\mathcal{C}), \tag{8}$$

where $\mathbb{1}$ is an indicative function that returns 1 for the condition is true and returns 0 for the condition is false, Mode is the function to get the mode of a set of values.

If $N_{overlap} \geq r$, we consider to implement retrospection, regarding $s = \text{Mode}(\mathcal{C})$ as the location of the summary token. Suppose the sequence $\{x_0, x_1, \ldots, x_s, \ldots, x_{t-1}\}$ that has presented knowledge aggregation pattern at the summary token x_s, we intend to roll the decoding procedure back to the sequence $\{x_0, x_1, \ldots, x_s\}$ and select the new next token in the complementary set $\mathcal{Y}/\{x_{s+1}\}$. Since the subsequent rollback will be further forward than previous ones, we manually specify that the rollback location s must be monotonically not decreasing. Additionally, we configure a maximum time β for rollback and consider to roll back to $\{x_0, x_1, \ldots, x_{s-1}\}$ if x_s has already reached the maximum rollback times.

4. Experiment

4.1. Setup

Models. We select four of the most representative MLLM models for evaluation, including InstructBLIP [9], MiniGPT-4 [48], LLaVA-1.5 [30] and Shikra [5]. These MLLM models can be roughly divided into two categories: Both InstructBLIP and MiniGPT-4 adopt Q-former [26] to bridge the features between vision and text modality, using just 32 tokens to efficiently depict image representations. While LLaVA-1.5 and Shikra simply leverage linear projection layers to align the features of two modalities, with 256 or even 576 image tokens as MLLM input. All of these MLLM models apply a well-pretrained model as their vision encoder, such as CLIP [35] and EVA [13], as well as a pretrained language model like LLaMA [38] or Vicuna [7]. Note that all of models used in our paper are 7B models.

Baselines. Since our work targets on the decoding approaches of MLLMs, we choose four decoding methods as the baseline methods, including three common strategies greedy decoding, Nucleus sampling, Beam search decoding and one method DoLa that is designed for mitigating LLMs' hallucination issues. Greedy decoding selects tokens step by step, greedily choosing the one with the highest probability in the language model logits. Improved on greedy decoding, Beam search decoding [3, 16, 37] maintains a set of beams to enlarge the candidate range and select the best on in beams finally. Different from the aforementioned two methods, nucleus sampling [17] concentrates concentrates on the predominant probability mass at each time step, maintaining a small subset of the vocabulary, typically ranging between one and a thousand candidates. DoLa [8], designed for hallucination reduction in LLMs, contrasts the logits of the mature layer with those of pre-mature layers, using the increment as the final output logits. We adopt the default settings of all of these baseline methods, where we unify $N_{beam} = 5$ for both Beam search and our OPERA, and set $p = 0.9$ for nucleus sampling. For DoLa, we use "0,2,4,6,8,10,12,14" as the indexes of candidate pre-mature layers and "32" as the index of the mature layer for DoLa.

Implementation details. Basically, OPERA is established on Beam search where $N_{beam} = 5$ by default. We empirically select $\sigma = 50$ as the scaling factor in Eq. (5), to ensure the attention values on knowledge aggregation patterns could be larger than 1 while the values on weaker attention areas could be smaller than 1. It aims to get the larger multiplication result on knowledge aggregation pattern. For the number N_{can} of candidates, it is a configurable hyperparameter like N_{can} and we set $N_{can} = 5$ by default. Too large N_{can} will consume lots of time during decoding. Besides, we unify $\alpha = 1$, $\beta = 5$ and $r = 15$ for all of MLLMs.

4.2. Quantitative Results

In this section, we evaluate OPERA's performance of mitigating hallucinations on both long descriptions, simplified VQA answers and popular MLLM benchmarks.

CHAIR evaluation on hallucinations. The Caption Hallucination Assessment with Image Relevance (CHAIR) [36] metric is a specifically crafted evaluation tool designed to assess object hallucination issues in image captioning task. More precisely, CHAIR quantifies the degree of object hallucination in a given image description by calculating the ratio of all objects mentioned in the description that are not present in the ground-truth label set. It comprises two distinct assessment dimensions, including CHAIR$_S$ that calculates on sentence-level and CHAIR$_I$ that calculates on image-level. Denoted as C_S and C_I, these two variants can be formulated as the average results of

$$C_S = \frac{\left|\{\text{hallucinated objects}\}\right|}{\left|\{\text{all mentioned objects}\}\right|}, C_I = \frac{\left|\{\text{captions w/ hallucinated objects}\}\right|}{\left|\{\text{all captions}\}\right|},$$

where the integration of CHAIR$_S$ and CHAIR$_I$ enables a thorough and detailed analysis of object hallucination issues in image captioning.

We conduct CHAIR evaluation on MSCOCO dataset [28], which contains more than 300,000 images and 80 objects with annotations. Specifically, we randomly select 500 images in the validate set of COCO 2014 and query different MLLM models with the prompt "`Please describe this image in detail.`" to get their descriptions. Considering the length of sequences can greatly affect the

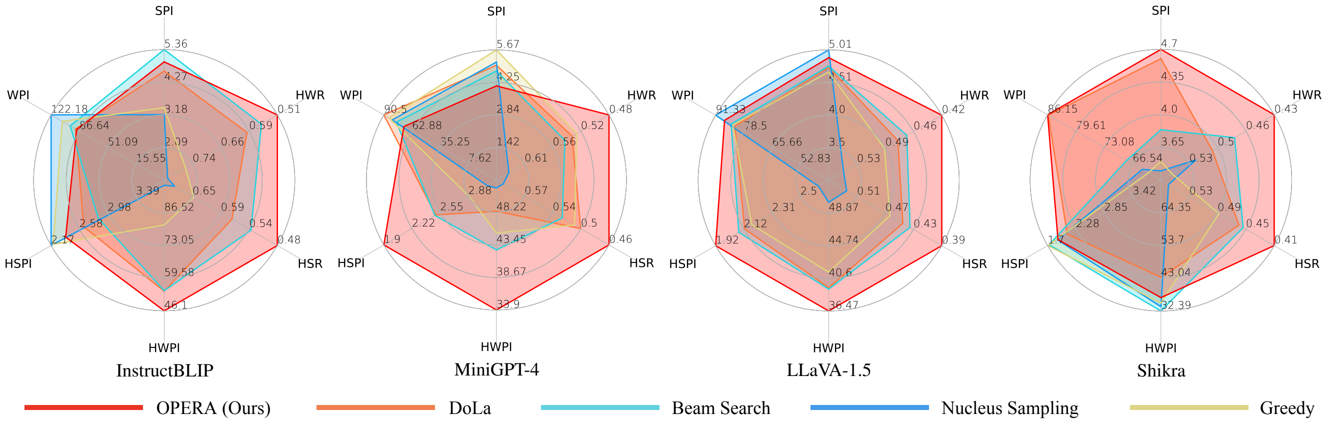


Figure 7. GPT-4 assisted hallucination evaluation [46] results on VG-100K dataset. Six aspects of values are analyzed, including the number of sentences per image (SPI), the number of words per image (WPI), the number of hallucinated sentences per image (HSPI), the number of hallucinated words per image (HWPI), the ratio of hallucinated sentences (HSR), and the ratio of hallucinated words (HWR). Note that larger SPI and WPI, smaller HSPI, HWPI, HSR and HWR are better. Larger radar indicates better performance.

Method	InstructBLIP		MiniGPT-4		LLaVA-1.5		Shikra	
	C_S	C_I	C_S	C_I	C_S	C_I	C_S	C_I
Greedy	58.8	23.7	31.8	9.9	45.0	14.7	55.8	15.4
Nucleus	54.6	24.8	32.6	10.7	48.8	14.2	55.6	15.4
Beam Search	55.6	15.8	30.6	9.5	48.8	13.9	50.4	13.3
DoLa	48.4	15.9	32.2	10.0	47.8	13.8	55.8	15.1
OPERA	**46.4**	**14.2**	**26.2**	**9.5**	**44.6**	**12.8**	**36.2**	**12.1**

Table 1. CHAIR hallucination evaluation results on four MLLM models (*max new tokens* is 512). Denote CHAIR$_S$ as C_S and CHAIR$_I$ as C_I. Smaller values corresponds to less hallucinations.

Method	InstructBLIP		MiniGPT-4		LLaVA-1.5		Shikra	
	C_S	C_I	C_S	C_I	C_S	C_I	C_S	C_I
Greedy	30.0	14.5	24.2	8.2	20.6	6.2	22.0	7.0
Nucleus	30.4	15.7	23.6	8.3	26.2	8.5	22.6	7.6
Beam Search	21.4	7.2	23.6	**7.8**	18.8	5.9	20.2	6.4
DoLa	22.2	7.1	24.2	8.2	20.4	6.3	20.2	6.3
OPERA	**16.6**	**6.8**	**22.6**	8.2	**14.2**	**5.2**	**14.2**	**5.9**

Table 2. CHAIR hallucination evaluation results on four MLLM models (*max new tokens* is 64). Denote CHAIR$_S$ as C_S and CHAIR$_I$ as C_I. Smaller values corresponds to less hallucinations.

values of CHAIR [27], we restrict two types of *max new tokens* to generate descriptions for fair evaluation.

As shown in Table 1 and Table 2, our OPERA obviously surpasses all of baselines decoding methods in both terms of C_S and C_I. Especially on Shikra, our method achieves ∼35% improvement on DoLa. The superior performances of OPERA are consistent between long description generation and short description generation.

GPT-4 assisted evaluation. CHAIR is a strong metric to evaluate the object-existence-level hallucination, while it fails to identify other kinds of hallucination, such as the attribute, location, and relation hallucination of objects. HalluBench [46] is an advanced benchmark, which use the detailed object-level description in the VG dataset [22] as ground-truth, and relay on the advanced GPT-4 to judge the hallucination in the description. In practice, the detailed objects-level description are gathered as a disordered comprehensive description about the image, and the GPT-4 is carefully prompted to judge the hallucination in the MLLM generated descriptions, sentence by sentence. Similar to Section 4.2, the MLLMs are prompted with "`Please describe this image in detail.`" and the *max new tokens* is set to 512.

From Figure 7, we observe that our OPERA generally achieves much less hallucinated sentences or words for describing each image, *e.g.*, ∼30.4% surpassing greedy decoding on the ratio of hallucinated sentences (HSR), and ∼15.4% surpassing DoLa at the ratio of hallucinated words (HWR). It indicates that OPERA does help the model partially overcome the hallucination issue caused by its bias or over-trusting problems. We also notice that OPERA somehow slightly reduce the length of MLLM's output sequence, it is probably attributed by the reducing of those additional hallucinated contents.

GPT-4V assisted evaluation. We further resort to GPT-4Vision, a strong multi-modal assistant that can easily handle the input from vision, language, and voice modality. Typically, we randomly sample 500 images from MSCOCO's validate set and ask different MLLM models to describe these images. For fair comparison, we following [42] and compare the answers obtained from two decoding methods at the same time, *i.e.*, providing the image and both the answers to GPT-4V and prompting it to give a judgement from 0-10 respectively. The prompt emphasizes mitigating the impact of the sequential order fed to GPT-4V and, additionally, paying special attention to the objects mentioned in answers but not appear in the image. It includes instances where the objects are represented in an

Method	InstructBLIP		MiniGPT-4		LLaVA-1.5		Shikra	
	C	D	C	D	C	D	C	D
Beam Search	5.52	5.26	5.29	5.06	5.53	5.15	5.25	5.08
OPERA	**6.26**	**5.27**	**6.87**	**5.08**	**6.32**	**5.16**	**6.29**	**5.26**

Table 3. GPT-4V assisted hallucination evaluation results on MSCOCO. Two aspects are verified, *i.e.*, correctness (C) and detailedness (D). Higher correctness indicates less hallucinations.

Method	InstructBLIP	MiniGPT-4	LLaVA-1.5	Shikra
Greedy	80.0	58.5	82.2	81.1
Nucleus	80.1	57.8	82.5	81.2
Beam Search	84.4	70.3	84.9	82.5
DoLa	83.4	72.8	83.2	82.1
OPERA	**84.8**	**73.3**	**85.4**	**82.7**

Table 4. POPE hallucination evaluation results on four MLLM models. We report the average F1-score computed on *random*, *popular*, and *adversarial* splits of POPE.

incorrect form of colors, positions, or relationships.

As showcased in Table 3, our OPERA achieves up to 27.5% improvements compared with Beam search decoding, while keeping the detailedness of answers. Since GPT-4V's abilities of perception and reasoning are very closed to human beings, the GPT-4V evaluation results somehow reflect the strong performance of reducing hallucinations from the perspective of human's feeling.

POPE evaluation on hallucinations. The Polling-based Object Probing Evaluation (POPE) [27] is a recently introduced method designed to assess hallucination issues in MLLMs. Similar to CHAIR, POPE focuses on evaluating object hallucination, utilizing an essay question format to prompt the model like "`Is There a <object> in the image?`", to determine whether the model can configure out the given image corresponds to a specific object. The complete POPE test comprises three splits: In the"random" split, the evaluation randomly selects objects from the whole dataset. In the "popular" split, the evaluation assesses the presence of objects that most frequently appear in the dataset. In the "adversarial" split, it evaluates the MLLM's ability to identify objects highly relevant to those present in the image.

We verify POPE on four MLLM models and report the average F1 scores in Table 4. Compared with baseline methods, we can observe our OPERA also attains the highest performance among these decoding strategies, albeit with marginal gains. It is essential to clarify that our approach excels specifically in alleviating hallucinations within **lengthy sequences**. In the context of POPE answers, where responses typically start with `Yes` or `No` and conclude as quite brief sequences like "`Yes, there is a <object> in the image.`", the knowledge aggregation patterns, a crucial hypothesis of our method, may not manifest as prominently.

	PPL_1↓	PPL_2↓	Grammar↑	Fluency↑	Natural↑
Greedy	12.72	10.27	**9.58**	**9.01**	8.52
Nucleus	17.17	13.78	8.51	8.53	7.95
Beam Search	**11.11**	**8.89**	9.54	8.95	**8.55**
DoLa	12.89	10.40	9.31	8.89	8.46
OPERA	11.67	9.31	9.54	8.93	8.53

Table 5. The evaluation results for the quality of generated text. We calculate PPL_1 and PPL_2 with *gpt2* and *gpt2-medium* in the *huggingface* model zoo respectively. The ratings of grammer, fluency, and naturalness is given by GPT-4.

	Greedy	Nucleus	Beam	DoLa	**OPERA**
MMBench	64.3	64.0	64.4	63.8	**64.4**
MME	1510.7	1471.9	1504.3	1480.1	**1515.4**

Table 6. OPERA generally improves the MLLM's performance on popular MLLM benchmark.

Text quality evaluation. To assess the overall quality of generated text comprehensively, we adopt PPL (Perplexity, a classical metric in NLP without using reference text), and resort to GPT-4 to assess the grammar, fluency, and naturalness of generated text. We randomly select 1,000 images in MSCOCO and verify on LLaVA-1.5 7B model. The average results are listed above, where PPL_1 and PPL_2 are calculated by pretrained *gpt2* and *gpt2-medium* respectively.

From the results in Table 5, we discover that OPERA can generally keep the quality of generated text from various aspects. Besides, we test OPERA on two popular MLLM benchmark, *i.e.*, MME [14] and MMBench [32], using LLaVA-1.5 7B model. Table 6 shows that OPERA can maintain and even improve MLLM's performance on both MLLM benchmarks.

5. Conclusion

We introduce OPERA, a novel MLLM decoding method that mitigates hallucination without requiring additional data, knowledge, or training costs. It is grounded in an Over-trust Penalty and a Retrospection-Allocation strategy, with the key observation that hallucinations are closely tied to knowledge aggregation patterns in the self-attention matrix, where MLLMs tend to focus on summary tokens, neglecting image tokens and resulting in content hallucination. Experiments show our superiority in reducing hallucination on various MLLMs and metrics.

Acknowledgement. This work is supported in part by the Natural Science Foundation of China under Grant 62121002, U20B2047, U2336206, 62372423, and 62102386. This work is partially supported by the National Key R&D Program of China (2022ZD0160201), and Shanghai Artificial Intelligence Laboratory.

References

[1] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 1, 3

[2] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023. 2

[3] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Audio chord recognition with recurrent neural networks. In *ISMIR*, pages 335–340. Curitiba, 2013. 3, 4, 6

[4] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 2

[5] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023. 1, 3, 6

[6] Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv preprint arXiv:2311.12793*, 2023. 3

[7] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023. 3, 6

[8] Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*, 2023. 4, 6

[9] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. 1, 3, 6

[10] Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Xilin Wei, Songyang Zhang, Haodong Duan, Maosong Cao, et al. Internlm-xcomposer2: Mastering free-form text-image composition and comprehension in vision-language large model. *arXiv preprint arXiv:2401.16420*, 2024. 1

[11] Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023. 3

[12] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2018. 3

[13] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19358–19369, 2023. 6

[14] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023. 8

[15] Zigang Geng, Binxin Yang, Tiankai Hang, Chen Li, Shuyang Gu, Ting Zhang, Jianmin Bao, Zheng Zhang, Han Hu, Dong Chen, et al. Instructdiffusion: A generalist modeling interface for vision tasks. *arXiv preprint arXiv:2309.03895*, 2023. 2

[16] Alex Graves. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012. 3, 4, 6

[17] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2019. 3, 6

[18] Qidong Huang, Jie Zhang, Wenbo Zhou, Nenghai Yu, et al. Initiative defense against facial manipulation. *arXiv preprint arXiv:2112.10098*, 2021. 3

[19] Qidong Huang, Xiaoyi Dong, Dongdong Chen, Hang Zhou, Weiming Zhang, and Nenghai Yu. Shape-invariant 3d adversarial point clouds. *arXiv preprint arXiv:2203.04041*, 2022.

[20] Qidong Huang, Xiaoyi Dong, Dongdong Chen, Weiming Zhang, Feifei Wang, Gang Hua, and Nenghai Yu. Diversity-aware meta visual prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10878–10887, 2023. 3

[21] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023. 3

[22] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017. 7

[23] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*, 2023. 2

[24] Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale N Fung, Mohammad Shoeybi, and Bryan Catanzaro. Factuality enhanced language models for open-ended text generation. *Advances in Neural Information Processing Systems*, 35:34586–34599, 2022. 3

[25] Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *arXiv preprint arXiv:2306.00890*, 2023. 2

[26] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with

frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 6

[27] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023. 7, 8

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

[29] Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Aligning large multi-modal model with robust instruction tuning. *arXiv preprint arXiv:2306.14565*, 2023. 2, 3

[30] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 1, 6

[31] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 1, 3

[32] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023. 8

[33] Potsawee Manakul, Adian Liusie, and Mark JF Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*, 2023. 3

[34] Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*, 2023. 3

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 6

[36] Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4035–4045, 2018. 6

[37] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014. 3, 4, 6

[38] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 3, 6

[39] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 3

[40] Bin Wang, Fan Wu, Xiao Han, Jiahui Peng, Huaping Zhong, Pan Zhang, Xiaoyi Dong, Weijia Li, Wei Li, Jiaqi Wang, et al. Vigc: Visual instruction generation and correction. *arXiv preprint arXiv:2308.12714*, 2023. 2

[41] Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. Label words are anchors: An information flow perspective for understanding in-context learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9840–9855, 2023. 2

[42] Shukang Yin, Chaoyou Fu, Sirui Zhao, Tong Xu, Hao Wang, Dianbo Sui, Yunhang Shen, Ke Li, Xing Sun, and Enhong Chen. Woodpecker: Hallucination correction for multimodal large language models. *arXiv preprint arXiv:2310.16045*, 2023. 2, 3, 7

[43] Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A Smith. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*, 2023. 3

[44] Pan Zhang, Xiaoyi Dong Bin Wang, Yuhang Cao, Chao Xu, Linke Ouyang, Zhiyuan Zhao, Shuangrui Ding, Songyang Zhang, Haodong Duan, Hang Yan, et al. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition. *arXiv preprint arXiv:2309.15112*, 2023. 1

[45] Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Kai Chen, and Ping Luo. Gpt4roi: Instruction tuning large language model on region-of-interest. *arXiv preprint arXiv:2307.03601*, 2023. 2

[46] Zhiyuan Zhao, Bin Wang, Linke Ouyang, Xiaoyi Dong, Jiaqi Wang, and Conghui He. Beyond hallucinations: Enhancing lvlms through hallucination-aware direct preference optimization. *arXiv preprint arXiv:2311.16839*, 2023. 7

[47] Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. Analyzing and mitigating object hallucination in large vision-language models. *arXiv preprint arXiv:2310.00754*, 2023. 2, 3

[48] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 1, 3, 6