

---

# Analyzing and Mitigating Object Hallucination in Large Vision-Language Models

---

Yiyang Zhou<sup>1\*</sup> Chenhang Cui<sup>1\*</sup> Jaehong Yoon<sup>1</sup> Linjun Zhang<sup>2</sup> Zhun Deng<sup>3</sup>  
Chelsea Finn<sup>4</sup> Mohit Bansal<sup>1</sup> Huaxiu Yao<sup>1</sup>

<sup>1</sup>UNC-Chapel Hill, <sup>2</sup>Rutgers University, <sup>3</sup>Columbia University, <sup>4</sup>Stanford University  
zhouyiyangailab@gmail.com, osallymalone@gmail.com, huaxiu@cs.unc.edu email

## 1 Introduction

Large Vision-Language Models (LVLMs) have made significant progress in understanding real-world images [1–6]. Although LVLMs have demonstrated their versatility and linguistic fluency, they often suffer from *object hallucination* in their generated text outputs [7–9]. Object hallucination refers to the phenomenon of generating inaccurate descriptions for a given image, including non-existent objects or omitting essential features. The issue with hallucinatory text generation in LVLMs is that it can mislead and deceive users in downstream applications that depend on these captions or descriptions, ultimately resulting in a negative impact on various fields that employ LVLMs, including robotics [10, 11], medical imaging [12, 13], and human-computer interaction [14, 15].

Early works have attempted to address the problem of object hallucinations in small-scale multimodal pre-trained models by performing either fine-grained alignment across different modalities [16] or reducing object co-occurrence patterns with data augmentation [17, 18]. However, the auto-regressive architecture of LVLMs differs significantly from small-scale multimodal pre-trained models, making their direct utilization impractical. A few recent works [19, 8, 1] have studied to reduce object hallucinations in LVLMs by enhancing the quality of datasets used for fine-tuning. Yet, acquiring a substantial number of high-quality examples for fine-tuning can be time-consuming and labor-intensive, requiring human expertise and effort. Instead, we aim to propose a lightweight method to post-hoc handle object hallucination by introducing **LURE: LVLM hallucination REvisor**.

Concretely, LURE is grounded in a rigorous statistical analysis that elucidates the underlying causalities of object hallucinations in LVLMs. This analysis delves into the relationship between the pre-training data and their corresponding textual responses from LVLMs that exhibit hallucinatory contents [20–22, 1]. Both our empirical and theoretical findings reveal that object hallucinations can be attributed to three key factors: co-occurrence, uncertainty, and object position. First, if the training data contains spurious co-occurring patterns between objects, language models may generate outputs based on these learned spurious associations, thus resulting in hallucinatory descriptions. Second, hallucinations occur more frequently on objects characterized by high uncertainty during generation. Lastly, positional factors also play a role, as more object hallucinations tend to appear in the latter portions of the generated description due to the accumulation of misinterpretation.

Based on our statistical analysis, LURE develops a object hallucination revisor. This revisor takes potentially hallucinatory descriptions as input and converts them into accurate ones. To create the revisor, we first generate a hallucinatory dataset using GPT-3.5 by making two modifications to the original correct captions: (1) Insert additional object texts into the description that are likely to co-occur with the objects contained in the initial description. This modification allows LURE to learn to disentangle such co-occurrence patterns effectively; (2) Replace uncertain objects or those at the end of descriptions with a placeholder tag, encouraging the revisor to re-evaluate these objects. In the end, we train our *hallucination revisor* leveraging the acquired hallucinatory dataset. Once trained, the revisor can seamlessly integrate with any LVLM to correct potential hallucinatory descriptions.

Our primary contribution is LURE, a lightweight and compatible post-hoc approach for rectifying object hallucination in LVLMs. This approach is grounded in our rigorous statistical analyses of object hallucinatory phenomena in LVLMs. Our experiments evaluate LURE on multiple existing

\*Equal contribution. Work was done during Yiyang Zhou and Chenhang Cui’s remote internship at UNC.

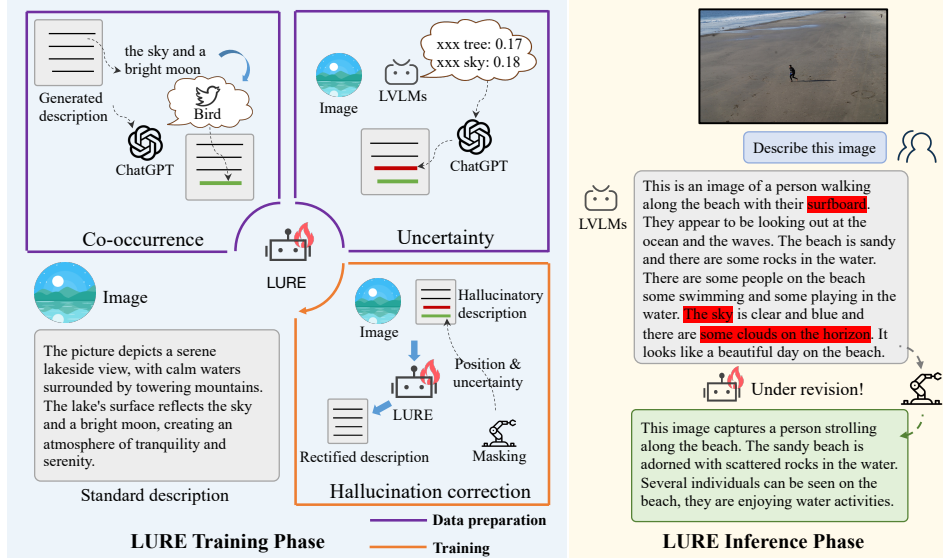


Figure 1: An illustration of LURE Framework: The orange-shaded section shows the training paradigm of LURE, where the black-bordered part represents the hallucinatory data generation phase, including introducing co-occurring objects and replacing either uncertain objects or objects in later positions in the descriptions. The purple-bordered part indicates the revisor training process, with the masking process that can be referenced in Alg. 1, which is located in Appendix B. The orange-shaded section illustrates an example in the inference phase of LURE.

open-source LVLMs. Compared to the best prior method, the results demonstrate that LURE can reduce object hallucination by approximately 23% under general object hallucination evaluation metrics (e.g., CHAIR [17]). Furthermore, LURE ranks first in GPT evaluation and human evaluation.

## 2 LVLm Hallucination Revisor

In this section, we formally introduce our remedy, LURE, which mitigates object hallucinations in large vision-language models. LURE is derived from a statistical analysis of the underlying factors that can cause object hallucination. We will first discuss these factors in this section. Subsequently, we will present LURE, which is designed to reconstruct clean data from corrupted input. To this end, we employ a hallucination revisor in our approach, aiming to transform potentially LVLm-generated hallucinatory descriptions into accurate ones. The framework of LURE is depicted in Figure 1.

**Notations.** Large Vision-Language Models (LVLMs) typically generate sentences in a free-form and auto-regressive manner, predicting the probability distribution of the next token progressively. In this context, we denote the input as  $x$ , the correct answer as  $y$ , and the generated sequence with a length of  $N_s$  as  $s = \{z_1, \dots, z_{N_s}\}$ . For a given LVLm, the probability of generating  $z_i$  as the  $i$ -th token can be described as  $p(z_i | s_{<i}, x)$  (where  $1 \leq i \leq N_s$ ), and  $s_{<i}$  refers to the previously generated tokens  $\{z_1, \dots, z_{i-1}\}$ . Given a description  $s$ , we additionally define the complete object set, which is arranged in the order of appearance, as  $\mathcal{O}_s = \{o_{s,1}, \dots, o_{s,n_h+n_r}\}$ . Here,  $n_h$  and  $n_r$  represent the number of hallucinatory and non-hallucinatory objects, respectively.

**Why Do Large Vision-Language Models Experience Object Hallucination?** We first scrutinize the root causes of object hallucinations in vision-language models via comprehensive statistical analyses from three critical viewpoints: *co-occurrence*, *uncertainty*, and *position*, recognized as the primary factors contributing to object hallucination. Our analysis indicates that object hallucinations are largely associated with co-occurrence patterns of frequently appearing objects in the dataset. Additionally, object hallucinations often exhibit high uncertainty during the decoding process and tend to appear in the latter half of the response text. A more detailed analysis of these three factors can be found in Appendix D. Furthermore, we provide a rigorous theoretical explanation that complements our empirical findings on object hallucinations. After thoroughly investigating the root causes of hallucinations, we then introduce how to train the deploy LURE.

## 2.1 Training Hallucination Revisor

In LURE, to train the hallucination revisor, we first curate a training dataset. Each example in this dataset consists of an image accompanied by a hallucinatory description, with the correct description serving as the output target. A significant challenge encountered during dataset curation lies in the generation of naturally-occurring hallucinatory descriptions. To overcome this challenge, LURE generates hallucinatory descriptions by modifying the accurate descriptions using GPT-3.5. These adjustments are guided by factors related to object hallucination, including co-occurrence, object uncertainty, and object position. In the following, we detail these modifications:

**Introducing Potential Co-Occurrence Objects.** To create a more naturally occurring co-occurrence scenario, rather than relying on counting co-occurrence frequencies from any specific datasets that may contain bias co-occurrence records, LURE leverages GPT-3.5 to deduce and incorporate objects that are most likely to co-occur in the scene into the original description.

**Reconsidering Uncertain Objects & Objects in Later Position in the Descriptions.** Hallucination is more prone to occur in objects with greater uncertainty and objects exist later in the description. In this context, we anticipate that the revisor should place greater emphasis on and reevaluate these objects. To achieve this, we utilize string matching to replace objects with significant uncertainty and those located at the end of the description with the placeholder tag "[IDK]". Here, to quantify object uncertainty in descriptions, we use the uncertainty values of noun tokens as a proxy. Token uncertainty is expressed as the entropy of each token, denoted as  $-\log p(z_i | s_{<i}, x)$ . We classify tokens as uncertain objects if their corresponding uncertainty exceeds a threshold  $\gamma$ , and if they are identified as nouns. Like uncertainty, we determine the later object’s position using the condition  $\text{Index}(z_i) \geq \eta * \text{Length}(s)$  and the threshold  $\eta$ . This approach enables the model to reassess and either replace "[IDK]" with a more appropriate object based on the image or remove it entirely.

Using these modification strategies, for every accurate description, we provide GPT-3.5 with a list of potential co-occurrence objects, and a list of uncertain objects. We then prompt GPT-3.5 to generate the corresponding hallucinatory description using the prompts in Appendix C.3. Finally, we leverage the constructed dataset to fine-tune a LVLM and use it as revisor. Some cases of hallucinatory descriptions are in Appendix F.2. The training pipeline is illustrated in Alg. 1 in Appendix B.

## 2.2 Deploying Hallucination Revisor

In the inference stage, the trained revisor rectifies the generated descriptions. Specifically, similar to the process of constructing hallucinated descriptions during the training phase, in the testing phase, we similarly integrate the placeholder tag "[IDK]" into the generated descriptions. This integration serves the purpose of enforcing the Revisor to reevaluate objects exhibiting high uncertainty or appearing later in the generated text. The inference pipeline is detailed in Alg. 2 in Appendix B.

## 3 Experiments

**Datasets.** MSCOCO [21] is a comprehensive dataset used for image recognition, segmentation, and captioning. It comprises over 300,000 images spanning more than 80 object categories, each with detailed annotations. Following [23, 8], we selected 5,000 unique images from the COCO 2014 training dataset to evaluate performance. To train the hallucination revisor, we randomly selected 5000 image-text pairs from LLaVA-150k [24], ensuring that these images were different from the ones used in testing.

**Baselines.** The comparison methods include: *Original*, which directly use the generated descriptions from LVLMs; *Teacher* [25]; *Chain-of-Thought (CoT)* [26]; *Greedy-Decoding*; *GPT-Ensemble*; *GPT-Teacher*. Detailed descriptions about baselines are in Appendix C.5.

**Evaluated LVLMs.** We performed experiments utilizing six of the most recent LVLMs, with their corresponding language models specified in parentheses: MiniGPT-4 (Vicuna 13B) [2], LLaVa (LLaMA 13B) [1], MMGPT (LLaMA 7B) [6], LLaMA-Adapter (LLaMA 7B) [27], mPLUG-Owl (LLaMA 7B) [3], and InstructBLIP (Vicuna 7B) [28].

**Hyperparameter Settings.** Unless specified, all experiments in the paper are using MiniGPT-4 as the backbone of the revisor, along with the training parameter settings provided in Appendix C.2.

Table 1: Automated hallucination evaluation is performed under six LVLMs using CHAIR<sub>S</sub> ( $C_S$ ) and CHAIR<sub>I</sub> ( $C_I$ ), where smaller values indicate less object hallucination. For additional metrics, please refer to Appx. E.1.

	MiniGPT-4		LLaVa		MMGPT		LLaMA-Adapter		mPLUG-Owl		InstructBLIP	
	$C_S \downarrow$	$C_I \downarrow$	$C_S \downarrow$	$C_I \downarrow$	$C_S \downarrow$	$C_I \downarrow$	$C_S \downarrow$	$C_I \downarrow$	$C_S \downarrow$	$C_I \downarrow$	$C_S \downarrow$	$C_I \downarrow$
Original	26.8	7.3	54.0	11.3	56.6	11.0	58.8	13.7	71.2	16.5	40.0	8.2
Teacher	24.0	5.7	49.9	9.3	53.4	7.5	40.8	9.4	62.4	13.0	36.4	7.5
CoT	31.6	9.4	47.6	9.0	48.8	17.5	43.3	9.4	56.9	13.4	35.7	7.8
Greedy-Decoding	25.1	6.6	50.9	10.0	50.6	8.4	55.9	13.7	55.1	12.8	35.5	7.8
GPT-Ensemble	41.0	10.6	43.0	10.7	51.0	11.1	47.1	13.0	52.0	15.2	51.0	13.0
GPT-Teacher	25.3	7.6	38.0	7.8	26.7	9.3	49.0	12.4	22.0	9.0	32.0	7.8
<b>LURE (ours)</b>	<b>19.7</b>	<b>4.9</b>	<b>27.1</b>	<b>6.4</b>	<b>22.2</b>	<b>5.6</b>	<b>35.3</b>	<b>9.1</b>	<b>18.8</b>	<b>5.4</b>	<b>21.0</b>	<b>5.1</b>

Table 2: We conducted evaluations for description ranking, comparing the four strongest baselines in both human ('H') and GPT ('G') evaluations. Metrics represent the average rankings within the top 1-5 positions, with lower rankings indicating less hallucination.

	MiniGPT-4		LLaVa		MMGPT		LLaMA-Adapter		mPLUG-Owl		InstructBLIP	
	G ↓	H ↓	G ↓	H ↓	G ↓	H ↓	G ↓	H ↓	G ↓	H ↓	G ↓	H ↓
Original	3.97	3.10	4.55	4.62	3.66	3.25	4.79	4.45	4.25	3.98	4.29	4.77
Teacher	3.36	3.83	3.30	3.07	3.09	3.20	3.00	3.13	3.25	3.66	3.34	3.53
CoT	2.44	2.83	3.05	2.52	4.38	4.07	2.63	2.10	3.75	3.13	2.78	2.21
GPT-Teacher	3.56	3.28	2.45	2.96	2.16	2.90	2.68	3.24	2.50	2.44	3.12	2.56
<b>LURE (ours)</b>	<b>1.67</b>	<b>1.96</b>	<b>1.65</b>	<b>1.83</b>	<b>1.61</b>	<b>1.58</b>	<b>1.90</b>	<b>2.08</b>	<b>1.25</b>	<b>1.79</b>	<b>1.47</b>	<b>1.93</b>

### 3.1 Evaluation Strategies and Results

**Evaluation Metrics.** In our evaluation, we employ both automated and human assessment strategies to evaluate the quality of object hallucination in images. The automated evaluation is based on the CHAIR metrics presented in [17]. In addition, we extend our evaluation to include other widely used metrics. Detailed descriptions and results for these additional metrics can be found in Appendix E.1. Details about evaluation metrics can be found in Appendix C.4.

**Results.** In Table 1 and Table 2, we report the results of automated evaluations and human and GPT evaluations under different LVLMs, respectively. Here, taking cost into account, we only compare LURE with the four strongest methods in human and GPT evaluations. Although Teacher, CoT, and GPT-Teacher can improve the performance compared to the original descriptions in most cases, LURE significantly enhances performance over these strong baselines, which effectively reduces object hallucination in generated descriptions. One potential reason for this is that all of these baselines experience error propagation to some extent. For instance, CoT’s linear guidance can lead to errors if the object listing step is incorrect. In contrast, LURE directly corrects hallucinatory descriptions using guidance from potential factors that can trigger hallucinations.

### 3.2 Analysis of LURE

We also evaluate the performance of LURE and present the following conclusions: (1) The performance gains of LURE are mainly derived from the revisor’s power in reducing hallucination rather than the use of constructed hallucination datasets. (2) The key factors we’ve identified do help reduce hallucinations in LVLMs and benefit the training process of the revisor. (3) Different LVLMs can be used as the backbone of the revisor, and all of them indicate strong performance. For these three conclusions, we provide a detailed analysis below:

**Are the Performance Gains of LURE from Using Constructed Hallucination Datasets?** To verify that the performance gains of our method are not from using additional data to train the revisor, we fine-tuned the original LVLMs with the additional dataset. The results on MiniGPT-4 are shown in Table 3, where "Original" represents the descriptions of MiniGPT-4. According to Table 3, LURE outperforms the fine-tuned LVLMs, which indicates that our method indeed reduces object hallucination by post-hoc rectifying potential hallucinatory descriptions rather than using additional data.

Table 3: Compared LURE to fine-tuning method using the training data of revisor.

Model	CHAIR <sub>S</sub> ↓	CHAIR <sub>I</sub> ↓
Original	26.8	7.3
FT (add'l data)	31.0	7.2
<b>LURE (Ours)</b>	<b>19.7</b>	<b>4.9</b>

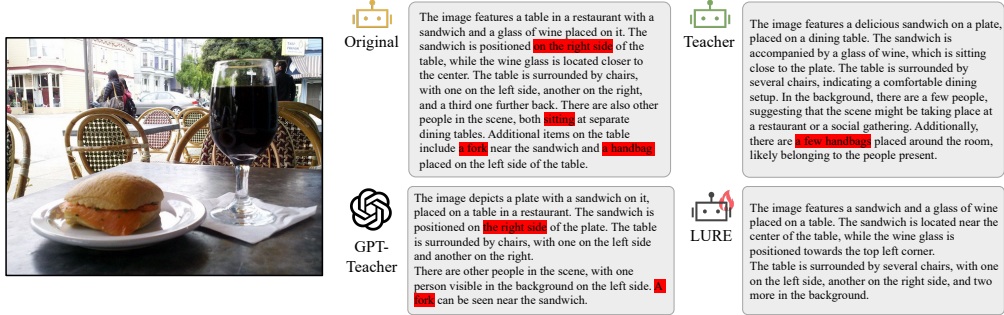


Figure 2: A case study comparing the levels of hallucination among various baselines.

**Ablation Study – Do the Hallucination Factors Contribute Performance Gains?** To demonstrate the impact of considering co-occurrence, uncertainty, and object position in reducing hallucination, we conducted ablation experiments and report the results in Table 4, where “Original” represents the descriptions of MiniGPT-4. In the ablation experiments, we trained and deployed the revisor without each of the three factors, one at a time. The results show that all three factors contribute to training a strong hallucination revisor to reduce object hallucination. Furthermore, we have also conducted an analysis of the changes in these three factors before and after applying the revisor, as presented in Appendix E.2. This analysis demonstrates that LURE can effectively reduce instances of hallucination caused by these factors.

Table 4: Ablation studies on three hallucination factors.

Model	CHAIR <sub>S</sub> ↓	CHAIR <sub>T</sub> ↓
Original	26.8	7.3
w/o Co-occurrence	22.6	4.9
w/o Uncertainty	21.2	5.4
w/o Position	22.3	5.8
<b>LURE (Ours)</b>	<b>19.7</b>	<b>4.9</b>

**Robustness Analysis of the Hallucination Revisor.** We further analyze the robustness of the revisor with respect to different backbones. Specifically, we trained the revisor on the same dataset using different backbones: MiniGPT-4, LLaMA-adapter, and mPLUG-Owl. The results are reported in Table 5, where “Original” represents the descriptions of MiniGPT-4. We can observe that despite the varying performance of each backbone, LURE consistently improve the performance compared to the original description, which further indicate the effectiveness of LURE. Additionally, we analyze the results of LURE with respect to various uncertainty thresholds in Appendix E.3. The findings demonstrate that LURE exhibits strong performance across a wide range of uncertainty thresholds.

Table 5: Performance under different hallucination revisor backbones.

Backbone	CHAIR <sub>S</sub> ↓	CHAIR <sub>T</sub> ↓
Original	26.8	7.3
MiniGPT-4	19.7	4.9
LLaMA-adapter	21.3	5.2
mPLUG-Owl	22.1	5.4

**Case Analysis.** We select several strong baselines and presented a case with rectified descriptions in Figure 2. Compared with other approaches, LURE excels in providing a more accurate image description. In the case, LURE accurately depicts the primary elements (e.g., sandwich, chair, plate) while avoiding hallucinatory objects like the fork and handbag. Although other baselines partially reduce hallucination, they still exhibit object hallucinations in their descriptions. Additionally, we also mitigate logical errors to some extent, including object orientation and actions. Further case analyses can be found in Appendices F.3 and F.4.

## 4 Conclusion

In this paper, our objective is to address the challenge of object hallucination in LVLMs. We introduce a lightweight post-hoc method, named LVM Hallucination Revisor (LURE), designed to rectify object hallucination in the generated descriptions produced by LVLMs. LURE is grounded in three key factors known to contribute to object hallucination: co-occurrence, uncertainty, and object position. These factors have been demonstrated to induce hallucination both empirically and theoretically. Our experiments, conducted on six open-source LVLMs, demonstrate the effectiveness of LURE in mitigating object hallucination in LVM-generated descriptions.

## References

- [1] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.
- [2] 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.
- [3] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023.
- [4] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726*, 2023.
- [5] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023.
- [6] Tao Gong, Chengqi Lyu, Shilong Zhang, Yudong Wang, Miao Zheng, Qian Zhao, Kuikun Liu, Wenwei Zhang, Ping Luo, and Kai Chen. Multimodal-gpt: A vision and language model for dialogue with humans, 2023.
- [7] Junyang Wang, Yiyang Zhou, Guohai Xu, Pengcheng Shi, Chenlin Zhao, Haiyang Xu, Qinghao Ye, Ming Yan, Ji Zhang, Jihua Zhu, et al. Evaluation and analysis of hallucination in large vision-language models. *arXiv preprint arXiv:2308.15126*, 2023.
- [8] 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.
- [9] Anisha Gunjal, Jihan Yin, and Erhan Bas. Detecting and preventing hallucinations in large vision language models. *arXiv preprint arXiv:2308.06394*, 2023.
- [10] Jinjie Mai, Jun Chen, Bing Li, Guocheng Qian, Mohamed Elhoseiny, and Bernard Ghanem. Llm as a robotic brain: Unifying egocentric memory and control. *arXiv preprint arXiv:2304.09349*, 2023.
- [11] Haokun Liu, Yaonan Zhu, Kenji Kato, Izumi Kondo, Tadayoshi Aoyama, and Yasuhisa Hasegawa. Llm-based human-robot collaboration framework for manipulation tasks. *arXiv preprint arXiv:2308.14972*, 2023.
- [12] Sheng Wang, Zihao Zhao, Xi Ouyang, Qian Wang, and Dinggang Shen. Chatcad: Interactive computer-aided diagnosis on medical image using large language models. *arXiv preprint arXiv:2302.07257*, 2023.
- [13] Mingzhe Hu, Shaoyan Pan, Yuheng Li, and Xiaofeng Yang. Advancing medical imaging with language models: A journey from n-grams to chatgpt. *arXiv preprint arXiv:2304.04920*, 2023.
- [14] Gary M Olson, James D Herbsleb, and Henry H Reuter. Characterizing the sequential structure of interactive behaviors through statistical and grammatical techniques. *Human-Computer Interaction*, 9(3-4):427–472, 1994.
- [15] Paul Brie, Nicolas Burny, Arthur Sluÿters, and Jean Vanderdonckt. Evaluating a large language model on searching for gui layouts. *Proceedings of the ACM on Human-Computer Interaction*, 7(EICS):1–37, 2023.
- [16] Ali Furkan Biten, Lluís Gómez, and Dimosthenis Karatzas. Let there be a clock on the beach: Reducing object hallucination in image captioning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1381–1390, 2022.
- [17] 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.

- [18] Jae Myung Kim, A Koepke, Cordelia Schmid, and Zeynep Akata. Exposing and mitigating spurious correlations for cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2584–2594, 2023.
- [19] Lei Li, Yuwei Yin, Shicheng Li, Liang Chen, Peiyi Wang, Shuhuai Ren, Mukai Li, Yazheng Yang, Jingjing Xu, Xu Sun, et al. M<sup>3</sup>it: A large-scale dataset towards multi-modal multilingual instruction tuning. *arXiv preprint arXiv:2306.04387*, 2023.
- [20] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems*, 24, 2011.
- [21] 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.
- [22] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3558–3568, 2021.
- [23] 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.
- [24] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.
- [25] Swarnadeep Saha, Peter Hase, and Mohit Bansal. Can language models teach weaker agents? teacher explanations improve students via theory of mind. *arXiv preprint arXiv:2306.09299*, 2023.
- [26] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.
- [27] Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023.
- [28] 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.
- [29] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021.
- [30] Yan Zeng, Xinsong Zhang, and Hang Li. Multi-grained vision language pre-training: Aligning texts with visual concepts. *arXiv preprint arXiv:2111.08276*, 2021.
- [31] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [32] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.
- [33] 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.
- [34] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

- [35] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%\* chatgpt quality, March 2023. URL <https://lmsys.org/blog/2023-03-30-vicuna/>.
- [36] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. [https://github.com/tatsu-lab/stanford\\_alpaca](https://github.com/tatsu-lab/stanford_alpaca), 2023.
- [37] Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A Smith. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*, 2023.
- [38] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.
- [39] 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.
- [40] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.
- [41] Markus Freitag and Yaser Al-Onaizan. Beam search strategies for neural machine translation. *arXiv preprint arXiv:1702.01806*, 2017.
- [42] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. *Advances in neural information processing systems*, 32, 2019.
- [43] Linjun Zhang, Zhun Deng, Kenji Kawaguchi, and James Zou. When and how mixup improves calibration. In *International Conference on Machine Learning*, pages 26135–26160. PMLR, 2022.
- [44] Edward James Hannan, AJ McDougall, and Don Stephen Poskitt. Recursive estimation of autoregressions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 51(2): 217–233, 1989.
- [45] Ching-Kang Ing. Accumulated prediction errors, information criteria and optimal forecasting for autoregressive time series. *The Annals of Statistics*, 35(3):1238–1277, 2007.
- [46] Jie Ding, Vahid Tarokh, and Yuhong Yang. Bridging aic and bic: a new criterion for autoregression. *IEEE Transactions on Information Theory*, 64(6):4024–4043, 2017.
- [47] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.
- [48] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.
- [49] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.
- [50] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.



## A Related Work

**Vision-Language Models.** Vision-language pre-trained models, as exemplified by [29, 30], demonstrate substantial capabilities in modeling interactions between visual and textual information, especially when fine-tuned for specific tasks. Recently, autoregressive large-scale language models (LLMs) [31–36] have ushered in a new era of vision-language models. These models, known as LVLMs, integrate LLMs with visual modality and showcase impressive visual understanding through end-to-end training techniques that directly decode visual and text tokens in a unified manner [1–4]. However, similar to VLMs, LVLMs also face the challenge of object hallucination [7, 17]. This form of object hallucination is more pronounced and widespread in the long-form descriptions produced by LVLMs compared to the shorter descriptions generated by VLMs [37].

**Hallucination in VLMs and LVLMs.** In VLMs, hallucination typically refers to scenarios where the generated descriptions contain information that does not exist in the visual modality [17, 16, 7]. Addressing object hallucination in VLMs is primarily achieved through techniques such as fine-grained contrastive learning [30], ROI feature fusion [16], and eliminating co-occurrence patterns through data augmentation [18]. However, the training paradigms between traditional VLMs and recent LVLMs differ, and the new autoregressive training paradigm in LVLMs makes it challenging to directly apply hallucination mitigation methods used in VLMs to LVLMs. Recent research has begun to address the issue of object hallucination in LVLMs, including hallucination evaluation and detection [7, 8, 23], as well as the construction of higher-quality datasets for fine-tuning [9, 19, 8, 1]. Nevertheless, acquiring a substantial number of high-quality examples can be time-consuming and labor-intensive. Instead, grounded in statistical analysis of hallucination, we propose a conceptually different approach, LURE, to post-hoc rectify object hallucination. We have already demonstrated its effectiveness in reducing hallucination and its compatibility with various LVLMs.

## B Training and Inference Algorithms

---

### Algorithm 1 Training LVLM Hallucination Revisor in LURE

---

**Require:** training image set  $\mathcal{X}$ ; groundtruth descriptions  $\mathcal{Y}$ ; LVLM  $\mathcal{M}(\cdot)$ ; uncertainty threshold  $\gamma$ ; hallucination revisor  $\mathcal{R}_\theta(\cdot)$  with parameters  $\theta$ ; position threshold  $\eta$

- 1: Use GPT-3.5 to construct hallucinatory description set  $\mathcal{H}_{old}$  (see Appendix C.3 for more details)
- 2: Initialize the revisor’s parameter  $\theta$  and an empty set  $\mathcal{H}_{new} \leftarrow \{\}$
- 3: **while** not converged **do**
- 4:   **for** each image  $x \in \mathcal{X}$  and the corresponding hallucinatory description  $h \in \mathcal{H}_{old}$  **do**
- 5:     Generate description  $s = \mathcal{M}(x)$  with object set  $\mathcal{O}_s$
- 6:     **for** object  $o_{s,i} \in \mathcal{O}_s$  **do**
- 7:       **if**  $o_{s,i}$  in  $h$  and  $-\log p(o_{s,i}|\mathcal{M}, x) \geq \gamma$  **then**
- 8:         Add placeholder tag “[IDK]” to  $h$ , i.e.,  $h \leftarrow \text{Mask}(h, o_{s,i})$
- 9:       **if**  $o_{s,i}$  in  $h$  and  $\text{Index}(o_{s,i}) \geq \eta * \text{Length}(h)$  **then**
- 10:         Add placeholder tag “[IDK]” to  $h$ , i.e.,  $h \leftarrow \text{Mask}(h, o_{s,i})$
- Put  $h$  into  $\mathcal{H}_{new}$
- 11:   Update parameter  $\theta$  with autoregressive loss  $\mathcal{L}(\mathcal{R}_\theta(\mathcal{H}_{new}), \mathcal{Y})$

---

---

### Algorithm 2 Inference Pipeline of LURE

---

**Require:** test image  $x_t$ ; LVLM  $\mathcal{M}(\cdot)$ ; trained hallucination revisor  $\mathcal{R}_\theta^*(\cdot)$ ; uncertainty threshold  $\gamma$ , position threshold  $\eta$

- 1: Generate description  $s_t = \mathcal{M}(x_t)$  with object set  $\mathcal{O}_{s_t}$
- 2: **for** object  $o_{s_t,i} \in \mathcal{O}_{s_t}$  **do**
- 3:   **if**  $-\log p(\text{object}|\mathcal{M}, x) \geq \gamma$  **then**
- 4:     Add placeholder tag “[IDK]” to  $s_t$ , i.e.,  $s_t \leftarrow \text{Mask}(s_t, o_{s_t,i})$
- 5:   **if**  $\text{Index}(o_{s_t,i}) \geq \eta * \text{Length}(s_t)$  **then**
- 6:     Add placeholder tag “[IDK]” to  $s_t$ , i.e.,  $s_t \leftarrow \text{Mask}(s_t, o_{s_t,i})$
- 7: **return**  $\mathcal{R}_\theta^*(s_t)$

---

## C Experimental Details

### C.1 Experimental Setting for the Hallucination Analysis

**Experimental Setting for D.1.** The objects in this experiment are based on the 80 object labels annotated in [17] from the COCO dataset, and the image descriptions are generated by MiniGPT-4 based on inference results from 5000 images in the COCO 2014 train dataset.

**Experimental Setting for D.2.** Because uncertainty and position analysis are relatively independent from co-occurrence, in order to avoid conducting statistical analysis on the training set distribution, the statistical data for uncertainty analysis is derived from MiniGPT-4’s descriptions of 200 images from the COCO 2014 test dataset. The computation of uncertainty is performed using  $-\log p(z_i|s_{<i}, x)$ .

**Experimental Setting for the Analysis of D.3.** Similar to the uncertainty analysis, we used the manually annotated descriptions of MiniGPT-4 for 200 images from the COCO 2014 test dataset, due to the need for precise positioning.

### C.2 Training Settings For Revisor

The overall revisor training setting is similar to MiniGPT-4. Here, we only need one A100 80G GPU for training, which takes approximately 10 minutes. We present hyperparameter settings of the LURE during the training phase, as shown in Table 6.

Table 6: Training hyperparameters.

Hyperparameters	
Training steps	410
Warmup steps	50
Max length	512
Batch size of multi-modal instruction data	12
Optimizer	AdamW
Learning rate	3e-5
Learning rate decay	Cosine
AdamW $\epsilon$	1e-6
AdamW $\beta$	(0.9, 0.999)
Weight decay	0.05

### C.3 Prompts for Training Dataset

We leverage the in-context few-shot learning capability of GPT-3.5 to generate hallucinatory data automatically for revising. Initially, we prompt GPT-3.5 to provide a list of objects that are highly likely to co-occur with the objects mentioned in the given description. Next, we use LVLMS (such as MiniGPT-4) to generate descriptions for the training set of 5000 images. During this process, we will save nouns with  $-\log p(z_i|s_{<i}, x)$  greater than the uncertain threshold  $\gamma$  in the decoding process to the list of uncertain objects corresponding to each image. Subsequently, we direct the model to take the original description and incorporate a randomly chosen word from the “co-occur objects” list, as well as another randomly chosen word from the “uncertain objects” list, into it. Detailed prompts are listed in Table 7 and a few examples are presented in Table 12.

### C.4 Details about Evaluation

In this section, we will provide the details about our evaluation metrics.

**Automated Object Hallucination Evaluation.** We follow the guidelines presented in [17] to perform an automated calculation of CHAIR metrics for the MSCOCO dataset, where 80 objects are involved in this automated evaluation process.

**Human and GPT Evaluations.** Although automated evaluation strategies are efficient, they cannot encompass all objects present in the evaluated images. To overcome this limitation, we conducted a

Table 7: The prompt for the GPT-3.5 API to generate the required hallucination dataset. “Instruction 1” is used to ask ChatGPT to provide a list of co-occurring objects based on the description, while “Instruction 2” is used to integrate the objects obtained from the co-occurring object list and the objects from the list of uncertain objects into the given description.

---

**Instruction 1:**

List three other objects that you think are most likely to appear with the objects in the scene described below:

{description}

Output in strict accordance with the following format:

Object one

Object two

Object three

---

**Instruction 2:**

Input caption: {description}

co\_objects list: {co\_objects list}

uncertain\_objects list: {uncertain\_objects list}

Select one object from “co\_objects list” and “uncertain\_objects list” respectively and add it to “Input caption” to get “Output caption”. (Try not to change the format)

Output caption:

---

comprehensive human evaluation involving several native speakers. Please refer to Appendix C.6 for the evaluation interface. In this human evaluation, participants are assigned the task of annotating hallucinatory objects and we rank different methods based on human feedback. In addition to human evaluation, inspired from [38], we also prompt GPT-3.5 to compare different descriptions. In this GPT evaluation, we provide the annotated information, including detection boxes and captions, and anticipate that GPT-3.5 can provide an ranking for the descriptions from various methods. For GPT evaluation, we use the prompts referenced in Table 9 in the Appendix.

## C.5 Details about Baseline

In this section, we will provide a detailed explanation of the settings used for the baseline in Table 1, including some parameter settings and prompt configurations. The detailed prompt for baselines can be seen in Table 8.

- **Teacher:** The “Teacher” approach involves generating short descriptions for the images via blip2 [39] and using them as context to guide the model in generating descriptions. By providing these descriptions as additional information, the model can benefit from the guidance and produce more accurate or relevant descriptions.
- **CoT:** The “CoT” method asks the model to first list the objects it identifies in the image and then describe the image based on those objects. It draws inspiration from the concept of chain of thought [26] and aims to guide the model in generating accurate descriptions by focusing on object recognition.
- **Greedy-Decoding:** The difference between the “Greedy-Decoding” strategy and the “Original” strategy is that in the “Greedy-Decoding” strategy, the model uses greedy decoding instead of sampling during the generation of image descriptions to produce the most deterministic output. This approach is used to explore the potential connection between the generation of illusions and the use of sampling.
- **GPT-Ensemble:** In “GPT-Ensemble,” we utilize GPT-3.5 to summarize the common elements in the descriptions generated by multiple LVLMS, excluding the one being evaluated. Subsequently, we employ GPT-3.5 to rewrite the description of the evaluated LVLMS, using the identified common elements from the descriptions of the other models to correct any dissimilar parts in the evaluated model’s description.

Table 8: Prompts for baselines.

---

<p><b>Teacher:</b>  Reference caption:  {blip2 caption}  Please refer to reference caption and describe this picture:</p>
---

---

<p><b>CoT:</b>  Human:  Please list the main objects in the picture and strictly follow the following format:  {object1, object2, object3.....}  AI:  {objects list}  Human:  Describe this image  AI:  {description}</p>
---

---

<p><b>GPT-Ensemble:</b>  Reference captions 1:{description of model 1}  Reference captions 2:{description of model 2}  Reference captions 3:{description of model 3}  Reference captions 4:{description of model 4}  Reference captions 5:{description of model 5}  Original Description:{description}  Synthesizing the commonalities of Reference captions 1-5, and then removing the parts in the Original Description that do not align with the commonalities, while preserving the original format. Answer:</p>
---

---

<p><b>GPT-Teacher:</b>  Reference caption:  {blip2 caption}  Original description:  {description}  Rewrite the original description to align it with the reference caption, delete some objects that you think are hallucinations, and keep the original format. Answer:</p>
--

---

- **GPT-Teacher:** "GPT-Teacher" represents the process of providing the GPT-3.5 API with contextual references and descriptions from the model's output, allowing it to revise the inaccurate description generated by the model into a more accurate version based on the contextual information.

## C.6 Details about Manual annotation evaluations


The manual evaluation annotation interface provides a user-friendly interface for performing manual annotations and capturing evaluation feedback. The interface is hosted on the Amazon Web Services (AWS) platform, which offers scalability, reliability, and security for handling annotation tasks. As shown in Figure 3, we annotated all objects and hallucinated objects in the descriptions based on the images. We then provided a binary label (0/1) to indicate whether each description contained hallucinations. Based on the fine-grained annotation results, similar to GPT evaluation, we sorted the results from different baselines.

**Instructions:**

- (1) the object list mentioned in the description
- (2) the list of hallucinatory objects mentioned in the description (limited to objects only)
- (3) whether hallucination exists in this description (1 if yes, 0 otherwise)

**Format Requirements:**

- (1) Please fill in the "object list" in the following format ("obj" refers to all the objects mentioned in the description.):  
obj1, obj2, obj3
- (2) Please fill in the "hallucinatory objects" in the following format ("hobj" refers to the illusory objects that exist in the description but are not present in the image):  
hobj1, hobj2, hobj3



**Model description:** There are four children on a bench wearing baseball uniforms. Two of them are boys and the other two are girls. All of them are smiling and looking at the camera. The bench is located in a park or a playground, as there are trees visible in the background. There is a building on the right side of the image. The children are wearing helmets, and one of them is holding a baseball bat.

object list

---

hallucinatory objects

---

1/0

**Submit**

Figure 3: Human evaluation annotation interface.

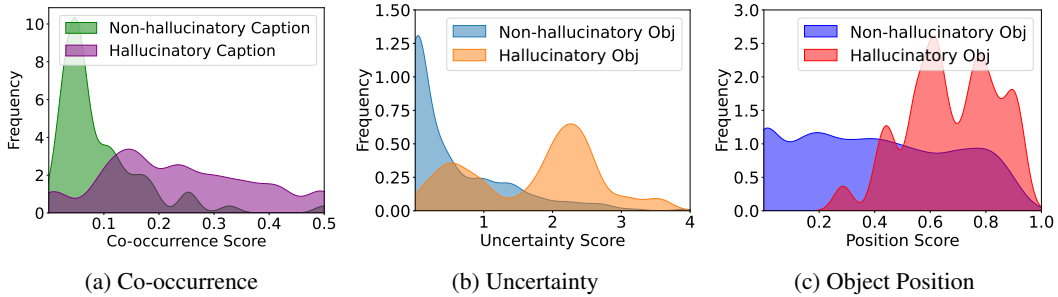


Figure 4: Comparison between hallucinatory and non-hallucinatory captions under different factors.

## D Detailed Analysis and Proof

### D.1 Co-Occurrence and Spurious Correlation Among Objects

In the realm of multi-modal models, “co-occurrence” denotes the frequent appearance of specific objects. When the training data includes spurious co-occurring patterns among objects, language models can generate outputs based on these learned associations. However, these associations may not hold true for test examples, resulting in hallucinatory outputs. For example, “grass” and “sky” frequently co-occur in the training data. The model falsely associates them and tends to generate “grass” and “sky” together even when only “grass” is present in the context.

In order to assess the influence of co-occurrence on object hallucination, we draw inspiration from [16] and introduce a *Co-occurrence Score* denoted as *CoScore*. For each image description  $s$ , the corresponding co-occurrence score  $\text{CoScore}_s$  is computed as the summation of co-occurrence degrees across all hallucinatory objects  $\{o_{s,1}, \dots, o_{s,n_h}\}$ , which is defined as:

$$\text{CoScore}_s = \sum_{i=1}^{n_h} \sum_{j=1, o_{s,j} \neq o_{s,i}}^{n_r+n_h} \frac{|\mathcal{S}(o_{s,i}) \cap \mathcal{S}(o_{s,j})|}{|\mathcal{S}(o_{s,i})| + |\mathcal{S}(o_{s,j})|}. \quad (1)$$

Here,  $\mathcal{S}(\cdot)$  denotes the set of all descriptions that mention a specific object, and  $|\mathcal{S}(\cdot)|$  represents the cardinality of this set.

Based on the definition of *CoScore*, we compare the distribution of co-occurrence scores between hallucinatory and non-hallucinatory captions (please refer to Appendix C.1 for our experimental

setting), As shown in Figure 4a, hallucinatory captions tend to exhibit higher co-occurrence scores, which suggests a stronger association between object hallucination and co-occurrence.

## D.2 Object Uncertainty

In language modeling, beam search [40, 41] is employed to predict words iteratively, introducing inherent uncertainty into the search process (Please refer to illustrative examples in Appendix F.1). This uncertainty is used as a measure of the model’s confidence in generating the next token, and can be related to the hallucination problem, as objects with higher uncertainty are more likely to be inaccurate. Here, we aim to quantitatively investigate the potential relationship between the uncertainty associated with objects at each prediction step and the hallucinations.

Concretely, we represent the probability of autoregressive decoding for each object token as  $p(o_{s,i}|s_{<k}, x)$ , where  $k$  denotes the positional index of object  $o_{s,i}$ . For each object  $o_{s,i}$ , the corresponding *Uncertainty Score* is defined as:

$$\text{UnScore}_{s,i} = -\log p(o_{s,i}|s_{<i}, x), \quad (2)$$

where a higher value of the uncertainty score indicates greater uncertainty. In Figure 4b, we perform a statistical analysis examining the connection between hallucination and object uncertainty (refer to Appendix C.1 for experimental details). Similar to the analysis of co-occurrence, hallucinatory objects are predominantly observed in the high-uncertainty range, while non-hallucinatory objects are more frequently generated in the certain range.

## D.3 Object Position in Generated Descriptions

Interestingly, we also find a significant correlation between the object position in the generated descriptions and hallucination, where dominant hallucinations occur in the latter part of the descriptions. To validate it, we introduce the *Positioning Score* denoted as PoScore for each object  $o_{s,i}$  as follows:

$$\text{PoScore}_{s,i} = \frac{\text{Index}(o_{s,i})}{N_s}, \quad (3)$$

where  $\text{Index}(o_{s,i})$  signifies the position index of object  $o_{s,i}$  within the entire description. Based on the definition of PoScore, we conduct a analysis of the positions of hallucination in the descriptions, illustrated in Figure 4c (refer to Appendix C.1 for experimental details). These findings indicate that high-density areas of hallucinatory objects predominantly appear towards the end of the sequence. This pattern corroborates our observation that object hallucination frequently occurs in the latter segments of generated text. One plausible explanation for this observed trend is rooted in the autoregressive text generation process. In the initial stages, the model closely adheres to the semantic information of its input image, resulting in coherent beginnings. However, as the generation progresses, the accumulation of past hallucinatory information and emerging uncertainties may steer the model off-course, ultimately leading to a more pronounced emergence of object hallucination.

## D.4 Theoretical Explanation

After examining these empirical correlations, we proceed to offer theoretical insights to explain them (all proofs can be found in Appendix D). Specifically, we focus on predicting the  $i$ -th token, denoted as  $z_i$ , and introduce a predictive function denoted as  $f$ . For each object  $k$  within a set of objects represented as  $[K]$ , the function  $f_k(s_{<i}, x)$  signifies the predicted score associated with the  $k$ -th object. Here,  $K$  is defined as the total number of objects under consideration, and we use  $y_k = 1$  to denote the presence of the  $k$ -th object in an image and  $y_k = -1$  otherwise. Furthermore, we make an assumption that  $f_k(s_{<i}, x)$  can be expressed as  $\langle \phi_k(s_{<i}, x), \beta_k \rangle$ ,  $\phi_k(s_{<i}, x) | y_k \sim N(y_k \cdot \mu_k^*, I_d)$  and  $\Pr(y_k = 1) = \Pr(y_k = -1) = 1/2$ . For a training set  $\mathcal{D}$ , the optimizer for the  $k$ -th class parameter  $\beta_k$  trained on  $\mathcal{D}$  is defined as:  $\hat{\beta}_k = \frac{1}{|\mathcal{D}|} \sum_{(s_{<i}, x, y_{i,k}) \in \mathcal{D}} y_{i,k} \cdot \phi_k(s_{<i}, x)$ , where  $y_{i,k} \in \{-1, 1\}$  represents whether object  $k$  will occur at position  $i$ . Such a model and optimizer are commonly used in the theoretical analysis of deep learning models [42, 43].

**Co-occurrence.** Based on this definition, we first consider co-occurrence. Without loss of generality, we assume that  $K = 2$ , and the first and second classes are frequently observed together, i.e., we observe  $(\phi_1(s_{<i}, x), \phi_2(s_{<i}, x))$  among a fraction  $\rho_0 \in (0, 1)$  of samples when both  $y_1$  and  $y_2$  are

equal to 1. Here, to simplify the autoregressive process while maintaining sequential prediction manner, we consider using  $\hat{f}_1 = \langle \phi_1(s_{<i}, x), \hat{\beta}_1 \rangle$  for the prediction of the first object, and in the second prediction, we model the information passed from the first information by  $\langle \phi_1(s_{<i}, x), \hat{\beta}_1 \rangle$ , and consider  $\hat{f}_2 = \langle \phi_1(s_{<i}, x), \hat{\beta}_1 \rangle + \langle \phi_2(s_{<i}, x), \hat{\beta}_2 \rangle$ . The model outputs the second object if  $\hat{f}_2(s_{<i}, x) > 0$ .

Under this setting, we consider two sampling schemes: (1) Each class is sampled according to the original training distribution; (2) Each class is sampled by setting  $\rho < \rho_0$ . These two sampling schemes result in two subset of samples  $\mathcal{D}^{(1)}$ ,  $\mathcal{D}^{(2)}$  with the same size. Denote the classifiers trained on  $\mathcal{D}^{(1)}$  and  $\mathcal{D}^{(2)}$  by  $\{\hat{f}_k^{(1)}\}_{k \in \{1,2\}}$  and  $\{\hat{f}_k^{(2)}\}_{k \in \{1,2\}}$  respectively. Theorem D.1 reflect reducing co-occurrence issue can lead to smaller test misclassification error  $Err(\cdot)$ .

**Theorem D.1** *Suppose  $\|\mu_k^*\|^2 \ll d$ ,  $d/|\mathcal{D}^{(k)}| \rightarrow \kappa$  for  $k \in \{1, 2\}$  and universal constants  $\kappa > 0$ . We have*

$$Err(\hat{f}_2^{(2)}) \leq Err(\hat{f}_2^{(1)}).$$

**Uncertainty.** We then turn our attention to object uncertainty. Here, we consider the two following sampling schemes: (1) Each class is sampled with equal probability  $1/K$ ; (2) Each class is sampled if the uncertainty score, defined as  $-\log(\hat{p}_k)$ , is above a certain threshold  $\gamma > 0$ . Here,  $\hat{p}_k$  is calculated as follows:  $\hat{p}_k = \frac{1}{|\mathcal{D}^{tr}|} \sum_{(s_{<i}, x, 1)} \sigma(\langle \phi_k(s_{<i}, x), \hat{\beta}_k \rangle)$ , where  $\mathcal{D}^{tr}$  represents the training set. These two schemes result in two subsets of samples  $\mathcal{D}^{(1)}$  and  $\mathcal{D}^{(2)}$  with the same size. Given  $x$  and  $s_{<i}$ , we make a prediction about whether the  $k$ -th object is present in the image using  $\hat{f}_k$ . Theorem D.2 illustrates that sampling more certain objects can lead to a reduction in test error.

**Theorem D.2** *Suppose  $\|\mu_k^*\|^2 \ll p$ ,  $d/|\mathcal{D}^{(k)}| \rightarrow \kappa$  for  $\kappa > 0$  and  $k \in [K]$ . We will have with probability at least  $1 - o(1)$ ,*

$$\frac{1}{K} \sum_{k=1}^K Err(\hat{f}_k^{(2)}) \leq \frac{1}{K} \sum_{k=1}^K Err(\hat{f}_k^{(1)}).$$

**Object Position.** The effect of object position on object hallucination is closely tied to error or prediction uncertainty accumulation in autoregressive models. This topic has been extensively studied in time series analysis, and several theoretical models have been established to investigate it [44–46].

## D.5 Proof of Theorem D.1

Let us denote  $N = |\mathcal{D}^{(1)}| = |\mathcal{D}^{(2)}|$ . For the detection rule of the first object, we have

$$\hat{\beta}_k^{(1)} = \frac{1}{|\mathcal{D}^{(1)}|} \sum_{(s_{<i}, x, y_{i,k}) \in \mathcal{D}^{(1)}} y_{i,k} \cdot \phi_k(s_{<i}, x).$$

As  $\phi_k(s_{<i}, x) \mid y_{i,k} \sim N(y_{i,k} \cdot \mu_k^*, I)$ , we write

$$y_{i,k} \cdot \phi_k(s_{<i}, x) = \mu_k^* + \epsilon_{i,k}.$$

Now, suppose among all samples, a fraction  $\rho_0 \in (0, 1)$  of samples have both  $y_1$  and  $y_2$  are equal to 1. We can then write

$$(\hat{\beta}_1^{(1)}, \hat{\beta}_2^{(1)}) = (\rho_0 \mu_1^* + \frac{1}{N} \sum_{i=1}^{\rho_0 \cdot N} \epsilon_{i,1}, \rho_0 \mu_2^* + \frac{1}{N} \sum_{i=1}^{\rho_0 \cdot N} \epsilon_{i,2}).$$

Use  $\Phi(\cdot)$  to denote the cumulative distribution function of a standard normal distribution. Then for the prediction function  $\hat{f}_2 = \langle \phi_1(s_{<i}, x), \hat{\beta}_1^{(1)} \rangle + \langle \phi_2(s_{<i}, x), \hat{\beta}_2^{(1)} \rangle$ , we have

$$\begin{aligned} \text{Err}(\hat{f}_2^{(1)}) &= \frac{1}{2} \mathbb{P}(\langle \phi_1(s_{<i}, x), \hat{\beta}_1^{(1)} \rangle + \langle \phi_2(s_{<i}, x), \hat{\beta}_2^{(1)} \rangle < 0 \mid y = 1) \\ &\quad + \frac{1}{2} \mathbb{P}(\langle \phi_1(s_{<i}, x), \hat{\beta}_1^{(1)} \rangle + \langle \phi_2(s_{<i}, x), \hat{\beta}_2^{(1)} \rangle > 0 \mid y = -1) \\ &= \Phi\left(-\frac{\langle \mu_1^*, \hat{\beta}_1 \rangle + \langle \beta_2, \hat{\mu}_2^* \rangle}{\sqrt{\|\hat{\beta}_1\|^2 + \|\hat{\beta}_2\|^2}}\right) \\ &= \Phi\left(-\frac{\rho_0 \|\mu_1^*\|^2 + \rho_0 \|\mu_2^*\|^2}{\sqrt{\rho_0^2 \|\mu_1^*\|^2 + \rho_0^2 \|\mu_2^*\|^2 + \frac{\rho_0 \cdot d}{N} + \frac{\rho_0 \cdot d}{N}}}\right) + o(1). \end{aligned}$$

Similarly, we have

$$\text{Err}(\hat{f}_2^{(2)}) = \Phi\left(-\frac{\rho \|\mu_1^*\|^2 + \rho \|\mu_2^*\|^2}{\sqrt{\rho^2 \|\mu_1^*\|^2 + \rho^2 \|\mu_2^*\|^2 + \frac{\rho \cdot d}{N} + \frac{\rho \cdot d}{N}}}\right) + o(1).$$

As  $\Phi\left(-\frac{\rho \|\mu_1^*\|^2 + \rho \|\mu_2^*\|^2}{\sqrt{\rho^2 \|\mu_1^*\|^2 + \rho^2 \|\mu_2^*\|^2 + \frac{\rho \cdot d}{N} + \frac{\rho \cdot d}{N}}}\right)$  is monotonically increasing with  $\rho$ , we complete the proof.

## D.6 Proof of Theorem D.2

We first analyze the uncertainty score. In fact, we have

$$\begin{aligned} \hat{p}_k &= \frac{1}{|\mathcal{D}^{tr}|} \sum_{(s_{<i}, x, 1)} \sigma(\langle \phi_k(s_{<i}, x), \hat{\beta}_k \rangle) \\ &= [\sigma(\langle \phi_k(s_{<i}, x), \hat{\beta}_k \rangle)] + o_P(1) \\ &= \left[ \frac{1}{1 + \exp(\|\mu_k^*\|^2 + \|\mu_k^*\| \cdot Z)} \right] + o_P(1), \end{aligned}$$

where  $Z \sim N(0, 1)$  is the standard normal random variable.

Therefore,  $\hat{p}_k$  decreases when  $\|\beta_k\|$  increases. Choosing samples with small  $\hat{p}_k$  (i.e., large  $-\log(\hat{p}_k)$ ) correspond to larger sample sizes for the classes with larger  $\|\mu_k^*\|$ .

Then we analyze the misclassification error. For  $\hat{f}_k = \text{sgn}(\langle \phi(s_{<i}, x), \hat{\beta}_k \rangle)$ , we have

$$\begin{aligned} \text{Err}(\hat{f}_k) &= \mathbb{P}(\text{sgn}(\langle \phi(s_{<i}, x), \hat{\beta}_k \rangle) \neq y) = \frac{1}{2} \mathbb{P}(\langle \phi(s_{<i}, x), \hat{\beta}_k \rangle < 0 \mid y = 1) \\ &\quad + \frac{1}{2} \mathbb{P}(\langle \phi(s_{<i}, x), \hat{\beta}_k \rangle > 0 \mid y = -1) \end{aligned}$$

As  $\phi_k(s_{<i}, x) \mid y \sim N(y_k \cdot \mu_k^*, I_d)$ , we have

$$\mathbb{P}(\langle \phi_k(s_{<i}, x), \hat{\beta}_k \rangle < 0 \mid y = 1) = \mathbb{P}(\langle \phi(s_{<i}, x), \hat{\beta}_k \rangle > 0 \mid y = -1) = \Phi\left(-\frac{\langle \mu_k^*, \hat{\beta}_k \rangle}{\|\hat{\beta}_k\|}\right).$$

As  $\hat{\beta}_k = \mu_k^* + \frac{1}{n_k} \sum_{i=1}^{n_k} \epsilon_i := \mu_k^* + \frac{1}{\sqrt{n_k}} Z$ , we have

$$\frac{\langle \mu_k^*, \hat{\beta}_k \rangle}{\|\hat{\beta}_k\|} = \frac{\|\beta_k\|^2 + \frac{1}{\sqrt{n_k}} \langle \mu_k^*, Z \rangle}{\sqrt{\|\mu_k^*\|^2 + \frac{2}{\sqrt{n_k}} \langle \mu_k^*, Z \rangle + \frac{1}{n_k} \|Z\|^2}}.$$

As we assume  $\|\mu_k^*\|^2 \ll d$ , we have

$$\frac{\langle \mu_k^*, \hat{\beta}_k \rangle}{\|\hat{\beta}_k\|} = \frac{\|\mu_k^*\|^2}{\sqrt{\|\mu_k^*\|^2 + \frac{d}{n_k}}} + o(1).$$

As a result, if the total sample size is fixed, choosing large  $n_k$  for small  $\|\mu_k^*\|$  will make the average misclassification error small.



## E Additional Analysis of LURE

### E.1 Model Performance Analysis with Additional Metrics

In this section, we conduct additional analysis using commonly used metrics from vision-language models on the same dataset, and discuss the applicability of these methods to hallucination evaluation.

#### E.1.1 Descriptions of Additional Metrics

**BLEU** BLEU (Bilingual Evaluation Understudy [47]) is a metric used to evaluate the quality of machine-generated translations by comparing them to one or more reference translations. The BLEU score is based on the idea of precision in  $n$ -grams, which are contiguous sequences of  $n$  words. It measures how well the generated translation matches the reference translations in terms of  $n$ -gram overlap.

**BertScore** BERTScore [48] is a method for evaluating the quality of natural language generation or summarization systems. BERTScore measures the similarity between a reference text and a generated text by computing contextualized embeddings using BERT.

**ROUGE-L** ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation - Longest Common Subsequence [49]) is an evaluation metric commonly used in natural language processing and text summarization tasks. It is designed to measure the quality of a machine-generated summary by comparing it to one or more reference summaries.

**CLIP** CLIP (Contrastive Language-Image Pretraining [50]) score is a metric used to evaluate the performance of the vision-language model, which measures how well the model can correctly associate images with their corresponding captions or textual descriptions.

#### E.1.2 Results

In Table 10, we present the performance of different models and baselines on these metrics. Based on the experimental results, it is evident that LURE outperforms the other baselines in both text translation metrics and image-text matching metrics, with a notable improvement in the CLIP Score metric. This could be attributed to the higher sensitivity of the CLIP Score, as compared to text translation metrics like BLEU, in capturing object-level differences. These findings are consistent with the overall experimental results presented in Table 1, further confirming the effectiveness of LURE. However, we have also identified certain issues related to the BLEU metric for text translation. The differences between baselines were not very pronounced, possibly because such metrics tend to emphasize the evaluation of text style rather than object-level distinctions. These metrics may not be well-suited for assessing hallucinations and long-form descriptions when compared to CHAIR.

Table 9: The prompt for ChatGPT3.5 evaluation.

---

<b>Instruction:</b>
Suppose you are a hallucination annotator who judges the degree of hallucination based on objects, and you have the following image information. Reference captions:{five captions from COCO}
Bounding box:{bounding boxes}
Please just provide the ranks for the below descriptions without any explanation, where the caption ranks first with the most hallucinations. The output format: [caption_x,...]
Descriptions:
caption_1: {description_1}
caption_2: {description_2}
caption_3: {description_3}
caption_4: {description_4}
caption_5: {description_5}
Output:

---

Table 10: Performance of different models and baselines on general metrics.

Models		BLEU-1	BLEU-2	BLEU-3	BLEU-4	BERTS	ROUGE-L	CLIPS
mPLUG-Owl	Original	30.37	14.59	5.618	2.505	<b>86.87</b>	30.21	0.168
	CoT	25.04	11.48	4.229	1.954	86.61	29.86	0.189
	Teacher	29.91	14.22	5.519	2.431	86.76	31.15	0.192
	Greedy-Decoding	30.29	14.30	5.509	2.502	86.59	30.35	0.208
	GPT-Ensemble	29.74	13.91	5.121	2.367	85.94	28.90	0.159
	GPT-Teacher	28.19	14.13	6.181	3.128	86.65	<b>30.87</b>	0.215
	<b>LURE (ours)</b>	<b>30.44</b>	<b>15.47</b>	<b>6.640</b>	<b>3.576</b>	86.65	30.31	<b>0.267</b>
LLaVa	Original	30.88	15.46	6.984	3.586	86.96	31.53	0.242
	CoT	29.94	15.01	7.042	3.718	86.99	31.82	0.211
	Teacher	30.52	15.54	7.334	3.906	87.11	31.76	<b>0.256</b>
	Greedy-Decoding	31.76	17.21	8.491	4.223	87.01	32.50	0.249
	GPT-Ensemble	25.68	16.24	7.047	2.893	84.10	30.84	0.201
	GPT-Teacher	22.06	19.54	3.393	1.493	85.94	27.62	0.251
	<b>LURE (ours)</b>	<b>35.94</b>	<b>21.81</b>	<b>11.33</b>	<b>6.804</b>	<b>87.39</b>	<b>32.59</b>	0.238
LLaMA-Adapter	Original	29.95	15.36	7.324	3.875	86.83	<b>31.77</b>	0.179
	CoT	25.45	11.41	4.233	1.687	86.48	39.98	0.201
	Teacher	26.71	12.88	5.388	2.636	86.65	30.50	0.142
	Greedy-Decoding	30.66	14.63	6.920	2.309	86.90	31.69	0.211
	GPT-Ensemble	24.92	11.21	4.678	1.890	84.92	27.12	0.140
	GPT-Teacher	25.13	10.25	3.929	1.684	85.85	28.68	0.186
	<b>LURE (ours)</b>	<b>30.94</b>	<b>15.81</b>	<b>7.334</b>	<b>3.804</b>	<b>86.96</b>	31.60	<b>0.223</b>
MiniGPT-4	Original	31.22	16.57	9.270	5.190	86.96	31.75	0.157
	CoT	33.68	20.57	10.72	6.430	86.09	32.39	0.177
	Teacher	32.69	19.87	9.870	5.350	86.06	30.72	0.142
	Greedy-Decoding	35.12	22.89	12.38	6.770	87.22	33.93	0.198
	GPT-Ensemble	29.65	19.22	9.878	5.330	85.77	29.83	0.140
	GPT-Teacher	33.37	20.28	11.52	5.770	87.01	31.89	0.182
	<b>LURE (ours)</b>	<b>41.20</b>	<b>23.17</b>	<b>13.18</b>	<b>7.580</b>	<b>87.88</b>	<b>35.34</b>	<b>0.210</b>
MMGPT	Original	27.27	12.66	5.680	2.290	79.79	29.03	0.177
	CoT	26.11	12.30	5.580	2.250	76.90	28.77	0.192
	Teacher	26.56	12.38	5.600	2.260	80.16	22.09	0.162
	Greedy-Decoding	30.15	15.11	6.320	3.573	86.62	31.77	0.188
	GPT-Ensemble	24.59	13.77	5.673	2.882	84.22	25.78	0.156
	GPT-Teacher	23.60	10.92	4.610	2.010	83.11	23.43	0.178
	<b>LURE (ours)</b>	<b>32.71</b>	<b>16.24</b>	<b>7.407</b>	<b>3.830</b>	<b>87.01</b>	<b>32.31</b>	<b>0.201</b>
InstructBLIP	Original	29.46	14.52	5.670	2.421	86.71	31.64	0.218
	CoT	24.04	12.61	4.086	1.837	85.50	28.07	0.229
	Teacher	25.61	12.22	4.321	1.963	85.93	29.89	0.294
	Greedy-Decoding	29.22	13.98	5.605	2.344	86.11	32.57	0.276
	GPT-Ensemble	26.32	13.11	5.101	2.396	85.04	30.77	0.198
	GPT-Teacher	24.91	11.92	4.652	2.097	85.81	29.49	0.205
	<b>LURE (ours)</b>	<b>29.77</b>	<b>15.23</b>	<b>5.708</b>	<b>2.634</b>	<b>87.94</b>	<b>32.95</b>	<b>0.307</b>

## E.2 Additional Analysis about the Hallucination Factors

### E.2.1 Comparative Studies on Changes in Hallucination Factors

To validate that our method reduces co-occurrence, uncertainty, and object positional bias that affect object hallucination, we further verify by evaluating the proportion of hallucinatory objects in high uncertainty, high co-occurrence, and sentence-ending positions. We compared the changes in various proportions of descriptions using MiniGPT-4 and LURE on the COCO 2014 test dataset. Here, we first describe how we calculate the object ratio under different factors:

**Ratio of Co-occurrence-Based Hallucinatory Objects.** Similar to uncertainty hallucination ratio, we obtain the  $C_{ratio}$  by calculating ratio of the number of hallucination objects with high co-occurrence

Table 11: Uncertainty-based hallucination object ratio, co-occurrence-based hallucination object ratio, and sentence-ending hallucination object ratio analysis on several models.

Models		Co-occurrence $C_{Ratio}$	Uncertainty $U_{Ratio}$	Position $S_{Ratio}$
MiniGPT-4	Original	0.106	0.221	0.227
	<b>LURE (ours)</b>	<b>0.071</b>	<b>0.145</b>	<b>0.139</b>
LLaVa	Original	0.243	0.103	0.331
	<b>LURE (ours)</b>	<b>0.142</b>	<b>0.086</b>	<b>0.139</b>
LLaMA-Adapter	Original	0.295	0.178	0.442
	<b>LURE (ours)</b>	<b>0.176</b>	<b>0.102</b>	<b>0.272</b>
mPLUG-Owl	Original	0.128	0.229	0.259
	<b>LURE (ours)</b>	<b>0.106</b>	<b>0.127</b>	<b>0.151</b>
MMGPT	Original	0.110	0.157	0.418
	<b>LURE (ours)</b>	<b>0.089</b>	<b>0.114</b>	<b>0.154</b>
InstructBLIP	Original	0.213	0.147	0.389
	<b>LURE (ours)</b>	<b>0.123</b>	<b>0.090</b>	<b>0.156</b>

score and the total number of objects with high co-occurrence score:

$$C_{ratio} = \frac{\sum_{s=1}^{M_h} \mathbb{1}[\text{CoScore}_s \geq \text{CoScore}_{mean}]}{\sum_{m=1}^M \mathbb{1}[\text{CoScore}_m \geq \text{CoScore}_{mean}]}, \quad (4)$$

where  $M_h$  is the number of hallucinatory descriptions,  $M$  represents the number of total descriptions, and  $\text{CoScore}_{mean} = \frac{1}{M} \sum_{m=1}^M \text{CoScore}_m$ .

**Ratio of Uncertainty-Based Hallucinatory Objects.** We obtain the  $U_{ratio}$  by calculating ratio of the number of hallucination objects with high uncertainty and the total number of objects with high uncertainty:

$$U_{ratio} = \frac{\sum_{s=1}^M \sum_{i=1}^{n_h} \mathbb{1}[\text{UnScore}_{s,i} \geq \text{UnScore}_{mean}]}{\sum_{m=1}^M \sum_{j=1}^{n_h+n_r} \mathbb{1}[\text{UnScore}_{m,j} \geq \text{UnScore}_{mean}]}, \quad (5)$$

where  $\text{UnScore}_{mean} = \frac{1}{M(n_h+n_r)} \sum_{m=1}^M \sum_{j=1}^{n_h+n_r} \text{UnScore}_{m,j}$ .

**Ratio of Hallucinatory Objects in Later Part of the Sentence.** For the ratio of hallucinatory objects in later part of the sentence., we calculate the  $S_{ratio}$  by calculating ratio of the number of hallucination objects in later part of the sentence and the total number of objects in later part of the sentence:

$$S_{ratio} = \frac{\sum_{s=1}^M \sum_{i=1}^{n_h} \mathbb{1}[\text{PoScore}_{s,i} \geq \eta]}{\sum_{m=1}^M \sum_{i=1}^{n_h+n_r} \mathbb{1}[\text{PoScore}_{m,i} \geq \eta]}, \quad (6)$$

where  $\eta$  is the position threshold.

**Results.** Based on the data presented in Table 11, it is evident that all three categories of ratios in the descriptions of LURE reduce when compared to the ratios of the original descriptions. This observation indicates that the elements of uncertainty, co-occurrence, and object position have contributed less to hallucinations in LURE.

### E.3 Parameter Sensitivity Analysis on Uncertainty

To further illustrate the robustness of our model, we conducted a parameter sensitivity analysis on the threshold of uncertainty. The uncertainty threshold  $\theta$  determines the proportion of replacing ‘‘object’’ with [IDK]. From the Figure 5, we can observe that our model is robust within a certain range of uncertainty threshold.

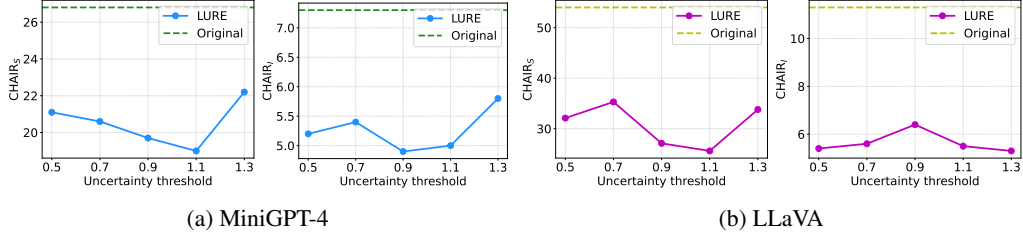


Figure 5: Sensitivity analysis of uncertainty threshold using MiniGPT-4 and LLaVA as revisor backbones.



This image depicts a group of people sitting around a table. The people are wearing different clothes. There is a **window** in the background, and the room appears to be well-lit. The walls of the room are painted white and there are two doorways that lead to other rooms.

The probability of the vocabulary within the red box

book: 0.17, window: 0.15, child: 0.13, wooden: 0.09, small: 0.07 .....

Figure 6: Case of uncertainty in the MiniGPT-4.

## F Additional Case Studies

### F.1 Cases of Uncertainty

We provide an example using MiniGPT-4 to illustrate the uncertainty present in LVLMs during the decoding process. In the example, we display the word probabilities in the vocabulary at the location of hallucinatory words (sorted in descending order of probability). As shown in Figure 6, we have displayed the decoded tokens and their probabilities at the point where the hallucinatory word “window” occurs. We can observe that the probability of the hallucinatory word “window” is comparable to that of “book”. The uncertainty in the model’s decoding path is highly influenced by the text generated earlier, leading to the incorrect selection of the word “window” when generating this token.

### F.2 Cases of Our Training Dataset

Here, we present some cases of training data constructed using GPT-3.5, as shown in Table 12. “Original caption” represents the original standard description, while the “Hallucination caption” column represents the hallucinated description constructed by GPT-3.5. The red portions in the hallucination captions indicate the hallucinations added by GPT-3.5 based on co-occurring object lists and uncertain object lists.

### F.3 Cases of Rewriting Captions

In this section, we present several examples of rectified descriptions to demonstrate the capabilities of LURE in reducing hallucination. From 8 we can find that our model demonstrates a high level of proficiency in removing or substituting hallucinatory objects.

### F.4 Additional Case Comparison between LURE and Baselines

We carefully selected several baselines that demonstrated promising performance based on our experimental results and conducted a thorough comparison with our proposed method. The detailed results of this comparison can be found in Figure 9. Upon comparing the descriptions generated by Revisor with those from the other methods, it becomes evident that Revisor surpasses the others in terms of accuracy and level of detail in describing the image.

The description produced by Revisor effectively captures the key elements of the image, such as the presence of a man wearing a white shirt walking on the tennis court while holding a tennis racket, as well as the presence of other individuals in the scene. On the contrary, the other methods fall short in various aspects. The "Original" method's description includes numerous hallucinated objects like the "net" and "cap." Although the "CoT" method's description has fewer hallucinated objects, it is observed that errors in the step-by-step reasoning process, such as incorrectly stating the presence of two tennis players, lead to corresponding errors in subsequent descriptions.

While the "Teacher" method's description is somewhat accurate, it still struggles to eliminate hallucinated objects effectively. Although GPT demonstrates strong textual comprehension abilities, it can still make mistakes when rewriting descriptions due to the absence of visual patterns, resulting in the omission of hallucinated objects and introducing errors.



**Original Caption:** The image shows a man walking down a rainy sidewalk while holding a bright red umbrella to stay dry. The man walks next to a building as rain pours down, making the umbrella a necessary accessory. In addition to the man with the red umbrella, there are several other people in the scene, some of which are carrying handbags despite the wet conditions. Towards the edge of the image, a person holding a small umbrella can be seen, indicating that others are also trying to shield themselves from the rain. The busy street with multiple passersby creates an atmosphere of a bustling city adapting to the rainy weather.

**Hallucination Caption:** The image shows a man walking down a rainy sidewalk while holding a bright red umbrella to stay dry. **Cars driving by with their headlights on contribute to the bustling city atmosphere.** The man walks next to a building as rain pours down, making the umbrella a necessary accessory. In addition to the man with the red umbrella, there are several other people in the scene, some of which are carrying handbags despite the wet conditions. Towards the edge of the image, a person holding a small umbrella can be seen, indicating that others are also trying to shield themselves from the rain. The busy street with multiple passersby creates an atmosphere of a bustling city adapting to the rainy weather.

(a) Case 1.



**Original Caption:** The image depicts a woman in a green shirt sitting at a dining table eating a meal. She is using a fork to pick at the food on her plate, which is positioned right in front of her. The table is set with a variety of items like a sandwich, a bowl, and multiple utensils such as knives and spoons. There are also several cups placed on the table. Although there are other chairs around the table, the woman appears to be dining alone, adding a sense of solitude to the scene. Other empty chairs can be seen in various positions around the dining table. Additionally, there are a few other individuals in the background, but they don't seem to be engaging with the woman or her meal.

**Hallucination Caption:** The image depicts a woman in a green shirt sitting at a dining table eating a meal. She is using a fork to pick at the food on her plate, which is positioned right in front of her. The table is set with a variety of items like a sandwich, **fries**, a bowl, and multiple utensils such as knives and spoons. There are also several cups placed on the table. A napkin and water glass are neatly placed beside her plate. Although there are other chairs around the table, the woman appears to be dining alone, adding a sense of solitude to the scene. Other empty chairs can be seen in various positions around the dining table. Additionally, there are a few other individuals in the background, but they don't seem to be engaging with the woman or her meal. **The salt and pepper shakers are placed at the center of the table**, within easy reach of the woman.

(b) Case 2.

Table 12: Cases of generating hallucinatory descriptions.

<Input image>

Describe this image



LURE

The image features a woman playing tennis on a clay court. She is in the middle of a swing, holding a tennis racket and preparing to hit the ball. The woman is wearing a pink dress and white shoes, which complement her athletic attire. There are several other **people** in the scene, likely **spectators** or **fellow players**, positioned around the court. **Some of them** are standing closer to the woman, while **others** are further away, observing the game. The main focus of the image remains the woman tennis player in action.



Under revision!!!

The image features a female tennis player standing on a clay tennis court. The player is wearing a red dress and is holding a tennis racket in her right hand, ready to play. Her left hand is raised above her head, possibly to catch the ball during the game. The player appears to be focused and determined, as she is getting ready to make a serve.

<Input image>

Describe this image



LURE

The image captures a man skillfully riding a wave on a surfboard in the ocean. He is crouching down on his board, maintaining balance and control as he navigates the wave. The surfer is positioned towards the center of the image, with the wave surrounding him. In the background, there are **several other people** in the water, possibly observing or waiting for their turn to surf. Some of them are closer to the **shore**, while **others** are further out in the ocean. The scene conveys the excitement and challenge of surfing, as well as the camaraderie among those who share this passion.



Under revision!!!

The image shows a person standing on a surfboard in the ocean, riding a large wave. The person is wearing a black shirt and black pants and is positioned on top of the surfboard as he rides the wave. The wave is large and has a lot of water coming off of it. The person appears to be enjoying the ride.

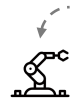


<Input image>

Describe this image



The image shows a man laying on his stomach in a grassy field, with water and a frisbee scattered around him. The man is wearing a pair of sunglasses and is smiling, as if he's having fun playing with the frisbee. The image is sunny, with green grass and trees in the background. There are also a few people visible in the distance, watching the man play.



Under revision!!!

The image shows a man laying on his stomach in a park, with a frisbee scattered around him. The man is smiling, as if he's having fun the frisbee. The image is sunny, with green grass and trees in the background. There are also a few people in the park, watching the scene.

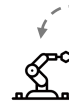


<Input image>

Describe this image



This image is of a man doing a trick on a snowboard in mid-air with his back facing the camera. The man is wearing a red jacket and black pants, and he is holding onto the snowboard with both hands. The background is a city street with buildings and cars in the distance. There are also banners hanging from the buildings.



Under revision!!!

The image features a man in mid-air, likely doing a trick on a snowboard. He is wearing a red jacket and black pants, and he is holding onto the snowboard with both hands. The man is the main focus of the scene, as he is the only person visible in the air.

Figure 8: Additional cases of rectified descriptions



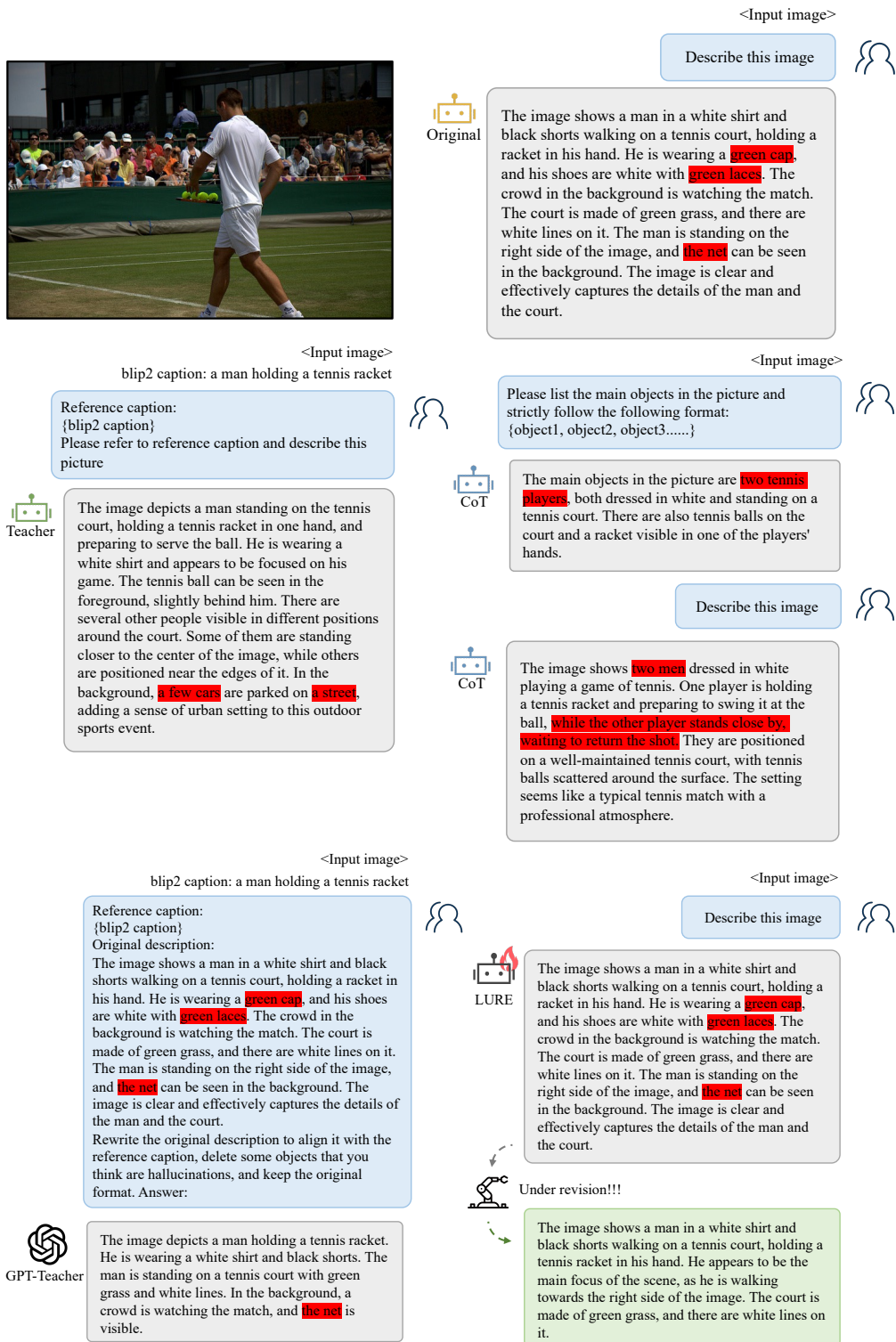


Figure 9: Case study of several strong baselines, including detailed dialogue flow of the real inquiry process for each baseline.