Detecting and Preventing Hallucinations in Large Vision Language Models

Anisha Gunjal*, Jihan Yin*, Erhan Bas†
Scale AI
anishagunjal@utexas.edu, jihan.yin@berkeley.edu, erhan.bas@gehealthcare.com

Abstract

Instruction tuned Large Vision Language Models (LVLMs) have significantly advanced in generalizing across a diverse set of multi-modal tasks, especially for Visual Question Answering (VQA). However, generating detailed responses that are visually grounded is still a challenging task for these models. We find that even the current state-of-the-art LVLMs (InstructBLIP) still contain a staggering 30 percent of the hallucinatory text in the form of non-existent objects, unfaithful descriptions, and inaccurate relationships. To address this, we introduce M-HalDetect, a Multimodal Hallucination Detection Dataset that can be used to train and benchmark models for hallucination detection and prevention. M-HalDetect consists of 16k fine-grained annotations on VQA examples, making it the first comprehensive multi-modal hallucination detection dataset for detailed image descriptions. Unlike previous work that only consider object hallucination, we additionally annotate both entity descriptions and relationships that are unfaithful. To demonstrate the potential of this dataset for hallucination prevention, we optimize InstructBLIP through our novel Fine-grained Direct Preference Optimization (FDPO). We also train fine-grained multi-modal reward models from InstructBLIP and evaluate their effectiveness with best-of-n rejection sampling (RS). We perform human evaluation on both FDPO and rejection sampling, and find that they reduce hallucination rates in InstructBLIP by 41% and 55% respectively. We also find that our reward model generalizes to other multi-modal models, reducing hallucinations in LLaVA and mPLUG-OWL by 15% and 57% respectively, and has strong correlation with human evaluated accuracy scores. The dataset is available at https://github.com/hendryx-scale/mhal-detect.

Introduction

Large language models (LLMs) have transformed the AI landscape in recent years, scaling their training data to trillions of tokens and their parameter count to hundreds of billions (Brown et al. 2020; Achiam et al. 2023; Touvron et al. 2023). This has unlocked powerful emergent behaviors, and seen widespread adoption through the use of chat agents such as ChatGPT. Recently, advances in multi-modal models have seen adoption around grafting visual backbones onto pre-trained large language models, resulting in LVLMs (Liu et al. 2023b; Dai et al. 2023; Ye et al. 2023). While this has led to strides in overall VQA performance, it brings along the same challenges that plague these LLMs - a significant one being the propensity to generate hallucinations.

In language models, hallucinations occur when the model produces inaccurate or misleading factual information that cannot be supported by existing knowledge stores (Ji et al. 2023; Bang et al. 2023). In the context of VQA for LVLMs, hallucinations can manifest as responses containing references or descriptions of the input image that are incorrect (Li et al. 2023). It is essential to address and mitigate these hallucinations to enhance the reliability and accuracy of multi-modal models in real-life usecases. However, these multi-modal hallucinations are hard to programatically detect and often requires human supervision, which can be costly.

To facilitate automatic hallucination detection, we build a diverse human-labeled dataset using VQA responses from InstructBLIP, as seen in Figure 1. We train multiple reward models of various densities (sentence and sub-sentence level) on this dataset for hallucination detection. An effective way to use these reward models to reduce hallucinations is to use them to generate rewards in a reinforcement learning setup (Ziegler et al. 2019; Stiennon et al. 2020; Nakano et al. 2021), although the resulting final model can only be as effective as the original reward model used (Bai et al. 2022). Therefore, in this paper, we focus on measuring the quality of these reward models, exploring classification metrics, and using best-of-n rejection sampling as an approximation of the system’s performance. Similar to (Rafailov et al. 2023), we also directly optimize InstructBLIP with fine-grained Direct Preference Optimization (FDPO), a novel variation of DPO in which we leverage fine-grained annotation information from individual examples, rather than collecting relative preference signals from pairs of texts. Both methods show significant success in reducing hallucination rates from InstructBLIP, and furthermore, rejection sampling with our reward models reduces hallucination rates in other multi-modal models as well - LLaVA (Liu et al. 2023b) and mPLUG-OWL (Ye et al. 2023).

Our main contributions are as follows:

1. We create and release M-HalDetect, a new hallucination detection dataset focused on fine-grained annotations at...
a sub-sentence level over detailed image descriptions.

2. We show that InstructBLIP can be optimized using Fine-grained DPO (FDPO) using the M-HalDetect dataset to reduce hallucination rates by 41%.

3. We show that reward models trained on this dataset can reduce hallucination rates by 55% in InstructBLIP with best-of-64 rejection sampling. The reward model generalizes to other LVLMs, reducing hallucination rates in LLaVA and mPLUG-OWL by 15% and 57% respectively with best-of-16 sampling.

4. We show that our reward model is an effective evaluator of hallucination rates, giving scores aligned with human ratings.

**Related Work**

Large Vision Language Models (LVLMs) have seen performative advancements in tasks such as generating text from images (Li 2023) and multi-modal in-context learning (Alayrac et al. 2022). Recent work has focused on utilizing instruction tuning techniques to enhance the zero-shot performance of instruction-aware LVLMs across different vision-language tasks (Liu et al. 2023b; Dai et al. 2023). These approaches utilize GPT-4 to generate multi-modal instruction tuning datasets (Liu et al. 2023b) where the image context is provided to GPT-4 through symbolic representations of the image such as captions and object bounding boxes. Others combine datasets across various multi-modal tasks (Dai et al. 2023) with hand-crafted instructions, a method that has found success in training traditional LLMs (Wei et al. 2021). This achieves state-of-the-art performance in a variety of multi-modal tasks such as visual and video QA, image captioning and classification.

Nevertheless, a significant challenge associated with LVLMs has emerged: preventing hallucinations when generating textual output. It is essential to address and mitigate these hallucinations to enhance the reliability and accuracy of LVLMs in production use cases.

**Hallucination Analysis in LVLMs** In (Li et al. 2023), the evaluation metric "POPE" is proposed to evaluate hallucinations in LVLMs by polling questions about generated text. They observed that current state-of-the-art LVLM (InstructBLIP) has the lowest object hallucination rates among recent LVLMs. Another relevant contribution by Liu et al. (Liu et al. 2023a) is the introduction of the LRV dataset. This dataset contains positive and negative instructions specifically designed to enhance the robustness of LVLMs against hallucination and inconsistent text generation. Furthermore, they proposed a method called GÀVIE, which leverages GPT-4 to assist in evaluating preferred answer generations.

These studies collectively contribute to the understanding and mitigation of hallucination-related challenges in LVLMs, by providing evaluation metrics, datasets, and evaluation methods that enhance the reliability and consistency of text generation in multi-modal models. Our work extends the scope of the previous works by not only considering hallucinations on the presence of objects, but also on descriptions of objects such as relative positioning or attributes. We also consider hallucinations on complex object reasoning.

**Aligning to Human Preferences** Despite having strong zero-shot performance on classical language benchmark datasets, pre-trained LLMs still struggle to produce detailed generations on par with those written by real humans. Supervised fine-tuning on demonstration data written by humans is not enough, where recent works have focused on using Reinforcement Learning with Human Feedback (RLHF) to address this problem (Stiennon et al. 2020; Touvron et al. 2023; Ouyang et al. 2022; Achiam et al. 2023).

RLHF typically uses Proximal Policy Optimization (Schulman et al. 2017), to optimize a policy model with rewards from a reward model. This reward model is typically trained on preference pairs of same-prompt generations, often sourced from the base policy model. This preference is usually given by humans, though attempts have been made to use more traditional metrics such as BLEU (Papineni et al. 2002) and ROUGE (Ganesan 2018) as proxies. Using human preferences is more effective in aligning LLMs to human preferences (Stiennon et al. 2020), though seen mixed results in hallucination prevention. Ouyang et al. (Ouyang et al. 2022) found that RLHF helps smaller (6B) language models reduce their hallucination rate, while having the opposite effect on larger models (175B). In this paper, we will focus on relatively smaller multi-modal models (7B) that can be more accessible to end users.

DPO has emerged recently as a viable alternative to RLHF for preference alignment, optimizing the policy model di-
rectly without needing to train a reward model and sample rewards through reinforcement learning (Rafailov et al. 2023). It has shown comparable performances with RLHF in summarization and chatbot usecases on language models, and maintains strong performance in higher temperature sampling. At the same time, it avoids the unstable and brittle process of training models with RL (Engstrom et al. 2020).

**Fine-grained Preferences** A limitation of both RLHF and DPO is their lack of fine-grained interpretability regarding what makes one generation more preferred than the other. Recent research has made significant progress in leveraging fine-grained user preferences to improve the performance and interpretability of reward models. For example, Wu et al. (Wu et al. 2023) utilize fine-grained human feedback to train multiple reward models at different density levels. These reward models covered passage level preferences as in the traditional RLHF setting, but also sentence level and sub-sentence level preferences in the form of error identification. (Lightman et al. 2023) employs process supervision, providing human feedback on individual steps for more robust rewards.

To extend this fine-grained feedback mechanism into the multi-modal domain, we introduce a new dataset for multi-modal hallucination detection. Our dataset comprises of 4,000 images with 4 detailed descriptions each, for a total of 16,000 image description pairs, annotated at the sub-sentence level to indicate the accuracy of the generated descriptions. Similarly to (Wu et al. 2023), we train sub-sentence and sentence level reward models on this dataset. We also modify the DPO loss to utilize fine-grained annotations.

**M-HalDetect : Multi-Modal Hallucination Detection Dataset**

**Dataset Description** In this section, we introduce the M-HalDetect dataset that incorporates fine-grained annotations for identifying hallucinations in detailed image descriptions generated by LVLMs. The dataset comprises of image-description pairs sampled from 4,000 images taken from the val2014 split of the Common Objects in Context (COCO) dataset (Lin et al. 2014). The dataset is divided into a training set with 3,200 images and a development set with 800 images.

We choose to utilize the validation set of COCO to avoid potential training data regurgitation from LVLMs trained on the COCO training set. This is roughly 10% of the original COCO validation set, leaving enough data untouched to not impact further validation too heavily.

To generate responses, we prompt InstructBLIP (Dai et al. 2023) with each image and a randomly selected question from a pool of instructions for describing an image. We initially reuse instructions from ones used in InstructBLIP’s detailed image description training data, which were sourced from the LLaVA-150k (Liu et al. 2023b) dataset. During initial analysis, we observed that doing so led to less diverse responses, potentially due to the influence of this dataset during training. To address this, we added in our own prompts to improve generation diversity. Refer to the appendix?? for details on dataset and diverse prompt generation, training, and inference analysis.

We sample four responses using nucleus sampling from InstructBLIP with a temperature value set to 1.0. This creates 16k image-prompt-response triplets, split between 12800 samples in the train split and 3200 samples in the val split.

**Dataset Categories** The annotation process involves categorizing different segments of each response into three categories: (i) Accurate, (ii) Inaccurate, and (iii) Analysis. We also include an Unsure category for ambiguous cases. We define the classes as follows:

- **Accurate** Objects exist in the image, their descriptions are accurate according the image, and any described relationships can be accurately inferred from the image.
- **Inaccurate** Objects do not exist in the image or their descriptions are inaccurate. Furthermore, if the analysis about the image is not plausible, it is also marked as Inaccurate.
- **Analysis** Scene or object analysis including complex reasoning or interpretations about the image. These are portions of the data that are more subjective and not grounded visually within the image.
- **Unsure** This category is reserved as a last resort if annotators cannot make a judgment about the sentence segment into one of the above three categories.

We provide fine-grained annotations for these 3 categories on the detailed descriptions of images generated by the LVLM. The annotations are provided at sub-sentence level - i.e. one sentence can comprise of multiple segments from different classes, as seen in Figure 1.

To make the annotation process user-friendly, we allow a leeway to the annotators to miss a few words in the annotations if there are too many segments in a sentence to be annotated. The unmarked words in a sentence are by default considered as ”Accurate”. In our analysis, we noticed that sometime annotators skip annotating punctuation, connector words, or introductory sub-sentences such as ”The image features” (illustrated in Figure 1).

**Dataset Collection** To collect the annotations, we employed Scale AI’s RAPID(ScaleAI 2023) labeling tool and involved 10 randomly selected human annotators. These annotators had to qualify by passing a training course with a minimum accuracy of 85% on the example tasks to be selected for the final tagging task. The annotators are presented with an image and four responses about the image generated by InstructBLIP. Their task is to annotate segments of the sentence into one of the categories. An example annotation task is illustrated in Figure 1.

**Method**

**Multi-Modal Reward Model**

We implement a multi-modal reward model for detecting the presence of hallucinations generated by LVLMs. Specifically, we reuse the InstructBLIP weights and architecture, swapping the final embedding layer with a classification
head. We do this as initializing the reward model from the generative model weights improves training robustness and reward generalization in later RL (Zheng et al. 2023). InstructBLIP consists of an image encoder that extracts image features and a linear mapping layer that projects these features. These image feature are passed to an instruction-aware attention layer, the QFormer, that attends instructions over the projected image features. The QFormer outputs are passed to a frozen pretrained decoder as soft prompts, prefixed to the instruction. For this paper, we choose to use Vicuna (Chiang et al. 2023) as the frozen decoder following the original InstructBLIP.

We train reward models at sentence level and sub-sentence level densities. For each image-text pair, we run one forward pass similar to (Lightman et al. 2023), and set target class labels at the token concluding each segment, masking out all other indices in the segment. We optimize with cross-entropy loss. We fine-tune the entire decoder and reward model head, while freezing the rest of the model. Ablations on model freezing, hyperparameters as well as details on training can be found in the extended version.

Sentence-level Reward Prediction

We condense the labeled sub-sentence segments in MHalDetect into sentence-level segments for a more structured reward format - this makes it more straightforward to run rejection sampling and train with RL, without worrying about localizing proper segments. We identify these sentences using the Natural Language Toolkit (Bird, Klein, and Loper 2009). For each sentence, if there is any segment that is inaccurate, we label the entire sentence as inaccurate. While this may introduce some noise when converting partially inaccurate sentences, we see in Figure 2 that the frequency of such sentences is low. Furthermore, if a sentence has a segment with the “unsure” category, we merge that sentence into the inaccurate class. We experiment with two levels of label granularity with this dataset:

- **Binary Classification**: Condense Analysis and Accurate classes into the Accurate class. In this setting we have two classes: Accurate and Inaccurate
- **Ternary Classification**: In this setting, we have three classes: Accurate, Inaccurate and Analysis.

Segment-level Reward Prediction

We also train a finer-grained reward model that make hallucination judgments on segments of sentences as opposed to entire sentences. This can provide less noisy signal when training on annotations, especially with longer compound sentences and hallucinations isolated to small portions of a sentence. We train on this data in a similar fashion to the sentence level rewards, by labeling the end token index of each span or segment of annotated text into its corresponding label. We then mask out every other index in the sequence. As a baseline, we assume perfect localization of the annotation segments as an upper bound for the performance of this method. Future works can consider training a segment localization model in parallel with the reward model, to detect when hallucinations start and end. Since we do not do this, we cannot use this reward model for rejection sampling, and evaluate purely on classification metrics over the test set. Similar to sentence-level reward prediction baselines, we also experiment with the binary and ternary variants of the segment-level reward prediction models.

Rejection Sampling (RS)

We use the trained reward models to perform rejection sampling on the generations of InstructBLIP to promote selection of less hallucinatory responses. We do this on the passage level, computing reward scores for the whole generation at once. We calculate the reward score by averaging the non-hallucination negative log probabilities of each sentence. This represents the normalized negative log probability of the entire passage containing no hallucinations. We compute rejection sampling in a best-of-n and worst-of-n setting, for $n = 16, 64$, to study the ability of the reward model in selecting the best generations from InstructBLIP, and the variance in quality between generations.

As we train two types of sentence level reward models (binary and ternary, including the analysis class), we experiment with using both models for reward scoring. We found in our initial experiments that although the binary reward model is able to penalize hallucinations with low scores, it tends to give very high scores towards the analysis class. We
The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)

Figure 4: Rejection sampling examples with ternary reward model labels per sentence. Score for each response is computed using the average negative log-probability per sentence of a hallucination.

Since we don’t have preferences over pairs of generations, but spans of fine-grained preferences throughout each generation, our FDPO loss can be modeled as

$$L_{FDPO} (\pi_\theta; \pi_{ref}) = -E_{(x,y,c) \sim \mathcal{D}} \left[ \log \sigma (\beta k) \right]$$

$$k = \begin{cases} 
- \infty & c = 0 \\
- r & c = 1 \\
- \infty & c > 1
\end{cases}$$

with sample segments $x, y, c$ being drawn from the dataset. Here, $x$ is the entire input up until the start of the current segment, $y$ is the generated segment, and $c$ is the class of the current segment, with $c = 1$ being the preferred class, $c = 0$ being the dispreferred class, and all other classes being ignored. Since segments are non-overlapping, we can run a single forward pass for each sample to calculate the loss of all segments within the sample all at once.

This formulation allows us to categorize each class into positive, negative, or neutral signal, the latter of which will be ignored during training. We run ablations on including the analysis class as either a negative or neutral class when optimizing InstructBLIP with FDPO. We fine-tune only the QFormer and language head, keeping the rest of the model frozen. We use $\beta = 0.5$ for all our FDPO experiments, and train for a maximum of 5 epochs with $lr = 10^{-6}$, warmup ratio of .03, and a cosine scheduler.

**Evaluation**

Recent works in multi-modal LLMs(Liu et al. 2023b,a) sometimes use GPT-4 as a human proxy to qualitatively evaluate LM outputs. Specifically, GPT-4 is prompted to give a preference score to a LM generation, either as a stand-alone or compared against GPT-4’s own generation. This metric enables automatic evaluation without depending on human evaluators.

However, this is plagued with systematic bias such as sensitivity to the ordering of responses (Wang et al. 2023). Furthermore, GPT-4’s public API does not yet support image inputs. Recent multi-modal works instead pass image context in the form of captions and object bounding boxes. In several cases, this symbolic input cannot represent the image.
Table 1: Results of reward model and human evaluation scores. The RM Score is the average negative log probability of the passage not containing hallucinations, while the human evaluation score is the percentage of content that was truthful. A perfect RM score would be 0, and a perfect human evaluation score would be 1.

Table 2: Baseline Reward Model Results
We introduce M-HalDetect, a novel multi-modal fine-grained hallucination detection dataset for benchmarking and training LVLMs to produce more truthful generations. We train fine-grained multi-modal reward models to perform rejection sampling against InstructBLIP. We innovate FDPO to optimize InstructBLIP directly on M-HalDetect, avoiding the need for preference pairs. Both methods significantly reduce InstructBLIP’s hallucination rate, extending their effectiveness to the multi-modal domain, and demonstrating the usefulness of M-HalDetect in catching and reducing hallucinations. We show this dataset is generalizable across multiple LVLMs, successfully reducing the hallucination rates of LLaVA and mPLUG-OWL.

While we show strong performance with rejection sampling, it is prohibitively slow for inference in real-world use-cases. The next step would be to optimize a generative model, perhaps InstructBLIP, using reinforcement learning with our trained reward models to create a higher quality LVLM for instruction aware VQA.

A limitation of modern day applications towards training large models with fine-grained feedback is that training typically takes place over multiple iterations of model training and feedback collection. This ensures the final model is more robustly aligned with the high level training objective. In this paper, we only perform one cycle of collecting response feedback and training. Indeed, when analyzing some of the responses, we can see hints of overfitting to our training objective - image descriptions are slightly more generic than before, and the preciseness of descriptions may have gone down. Future work can extend our dataset and methods to also account for descriptiveness and informativeness, training multiple reward models for optimizing a more robust final model.

**Conclusion**

We evaluate two variations of FDPO across the three classes - one that ignores analysis (IA), and one that disfavors analysis (DA), merging it with the inaccurate class. We see in Table 1 that marking analysis as a negative class does not impact hallucination rates in a significant way when training with FDPO, and may actually worsen rates at higher temperatures. We suspect that this may be because InstructBLIP’s generations often have the last sentence being subjective analysis of the image, followed by an end of sequence token. Pushing down the likelihoods of generating this sentence increases the likelihood of the generation being lengthened, potentially inducing additional hallucinations as the model runs out of accurate content to describe.

On the other hand, we see that ignoring analysis in FDPO training almost cuts hallucination rates in half. Even sampling at high temperature, generations still on average contain less hallucinations than the baseline InstructBLIP model sampled at 0 temperature, where it would have the least propensity to hallucinate. This is slightly better than best-of-16 rejection sampling, and almost as good as best-of-64 rejection sampling. This performance gap is to be expected as rejection sampling can generalize over the entire set of possible model generations, whereas FDPO is more limited in optimizing only over the data that it sees in the training data. Though, there is a trade-off in this performance, as best-of-n rejection sampling is slower in inference by a factor of n.

![Figure 5: Human evaluation scores against reward scores for all human evaluated results.](image)

![Figure 6: Reward model score means and variances as n increases in best-of-n rejection sampling. We see diminishing returns as we increase n.](image)

from InstructBLIP, they can still be used successfully in evaluating and improving on other LVLMs. It is interesting to see LLaVA’s baseline model performing so strongly - we suspect this is because LLaVA is trained specifically for generating detailed descriptions, whereas InstructBLIP and mPLUG-OWL are more general models with a wide range of task applicability.

Additionally, we study the correlation between reward model and human evaluation scores. In Figure 5, we see that across all human evaluated results, there is a clear and strong correlation between our reward model scores and human accuracy scores. Although this is by no means a robust replacement for human annotations, this shows the potential of training models as specific evaluators for hallucinations. Despite the noisiness, such a model could be used for early hyper-parameter selection, being much more cost-effective than humans evaluation.

**Fine-Grained DPO** We evaluate two variations of FDPO across the three classes - one that ignores analysis (IA), and one that disfavors analysis (DA), merging it with the inaccurate class. We see in Table 1 that marking analysis as a negative class does not impact hallucination rates in a significant way when training with FDPO, and may actually worsen rates at higher temperatures. We suspect that this may be because InstructBLIP’s generations often have the last sentence being subjective analysis of the image, followed by an end of sequence token. Pushing down the likelihoods of generating this sentence increases the likelihood of the generation being lengthened, potentially inducing additional hallucinations as the model runs out of accurate content to describe.

On the other hand, we see that ignoring analysis in FDPO training almost cuts hallucination rates in half. Even sampling at high temperature, generations still on average contain less hallucinations than the baseline InstructBLIP model sampled at 0 temperature, where it would have the least propensity to hallucinate. This is slightly better than best-of-16 rejection sampling, and almost as good as best-of-64 rejection sampling. This performance gap is to be expected as rejection sampling can generalize over the entire set of possible model generations, whereas FDPO is more limited in optimizing only over the data that it sees in the training data. Though, there is a trade-off in this performance, as best-of-n rejection sampling is slower in inference by a factor of n.
Acknowledgements
We thank Sean Hendryx and Utsav Garg for their feedback and support through internal development of the paper.

References


