THRONE: An Object-based Hallucination Benchmark for the Free-form Generations of Large Vision-Language Models

(Supplementary Material)

Prannay Kaul[1]* Zhizhong Li[2]† Hao Yang [2] Yonatan Dukler [2]
Ashwin Swaminathan [2] C. J. Taylor [2] Stefano Soatto [2]
VGG, University of Oxford[1] AWS AI Labs[2]
prannay@robots.ox.ac.uk {lzhizhon,haoyng,dukler,swashwin,taylorcj,soattos}@amazon.com

Contents

---

*Work conducted during an internship at Amazon

†Corresponding author

1 Voting Mechanism Ablation

Model	k	P_{ALL}	R_{ALL}	F^{1}_{ALL}	$F^{0.5}_{\text{ALL}}$	P_{CLS}	R_{CLS}	F^{1}_{CLS}	$F^{0.5}_{\text{CLS}}$	Ignore %
Adapter-v2	9	63.6	73.3	68.1	65.3	68.2	70.6	69.4	68.7	2.4
	5	61.8	75.0	67.7	64.0	65.7	72.0	68.7	66.9	0.0
	8	63.4	73.8	68.2	65.2	68.0	70.8	69.4	68.5	1.4
Adapter-v2.1	9	63.8	73.7	68.4	65.5	67.4	71.2	69.3	68.1	2.4
	5	61.7	75.3	67.8	64.0	64.7	72.5	68.4	66.1	0.0
	8	63.6	74.1	68.5	65.5	67.2	71.5	69.3	68.0	1.5
InstructBLIP	9	70.8	74.3	72.5	71.5	77.2	71.9	74.5	76.1	2.5
	5	68.2	77.2	72.4	69.8	73.2	74.3	73.7	73.4	0.0
	8	70.6	75.1	72.8	71.4	76.8	72.3	74.5	75.9	1.5
Otter-Image	9	33.0	31.2	32.1	32.7	25.2	16.9	20.2	22.9	8.5
	5	25.6	34.7	29.5	27.1	16.4	20.1	18.0	17.0	0.0
	8	32.4	31.8	32.1	32.3	23.9	17.2	20.0	22.2	4.8
MiniGPT-4	9	81.7	59.8	69.0	76.1	79.9	61.8	69.7	75.5	2.9
	5	74.8	64.9	69.5	72.6	73.0	65.4	69.0	71.3	0.0
	8	80.8	61.1	69.6	75.9	79.1	62.4	69.8	75.1	1.8
MiniGPT-v2	9	79.0	66.6	72.3	76.2	77.6	67.0	71.9	75.2	2.8
	5	73.6	71.4	72.5	73.1	72.1	70.5	71.3	71.7	0.0
	8	78.4	67.8	72.7	76.0	76.9	67.7	72.0	74.8	1.8
LLaVA-Mistral	9	86.8	71.8	78.3	83.6	84.4	64.2	70.8	77.5	2.7
	5	82.8	75.9	78.3	81.2	78.5	68.3	71.2	77.4	0.0
	8	86.3	73.1	78.5	83.3	83.7	65.0	71.2	77.4	1.6
mPLUG-Owl	9	55.5	71.9	62.6	58.1	66.3	68.3	67.3	66.7	2.4
	5	54.3	73.9	62.6	57.3	63.7	69.9	66.6	64.8	0.0
	8	55.5	72.6	62.9	58.2	66.2	68.6	67.4	66.7	1.4
LRV-Instruction-v2	9	82.0	56.7	67.0	75.3	78.4	58.8	67.2	73.5	3.6
	5	77.5	60.8	68.1	73.5	74.6	61.9	67.7	71.7	0.0
	8	81.7	57.9	67.8	75.5	78.0	59.4	67.4	73.4	2.0
LLaVA-v1.3	9	80.5	65.2	72.1	76.9	79.9	65.3	71.9	76.5	2.4
	5	76.4	68.7	72.4	74.7	75.6	68.0	71.6	73.9	0.0
	8	80.0	66.3	72.5	76.9	79.4	65.8	72.0	76.3	1.4

Table 1. **Comparison of Voting Mechanisms in THRONE.** We compare three different voting mechanisms: *unanimous*, $k = 9$; *simple majority*, $k = 5$; and *jury majority*, $k = 8$. Moreover in each case we report the number of ignore labels as a result of each voting mechanism. The number of labels is $400,000$ for a given LVLM. We find that the number of ignore labels is low is almost all cases and metrics are strongly correlated (> 0.99). In THRONE, we use a *unanimous* voting mechanism ($k = 9$) to minimize the likelihood of hallucination judgement errors.

2 THRONE vs. POPE Sampling

Model	Sampling	P_{ALL}	R_{ALL}	F^{1}_{ALL}	$F^{0.5}_{\text{ALL}}$	P_{CLS}	R_{CLS}	F^{1}_{CLS}	$F^{0.5}_{\text{CLS}}$
Adapter-v2	THRONE	63.6	73.3	68.1	65.3	68.2	70.6	68.3	68.0
	POPE	77.3	73.2	75.2	76.5	84.6	70.0	76.9	81.2
	Δ	(-13.7)	0.2	(-7.1)	(-11.1)	(-16.3)	0.5	(-8.6)	(-13.2)
Adapter-v2.1	THRONE	63.8	73.7	68.4	65.5	67.4	71.2	68.1	67.5
	POPE	79.2	73.2	76.1	77.9	85.4	71.8	76.3	80.6
	Δ	(-15.4)	0.5	(-7.7)	(-12.4)	(-18.0)	(-0.6)	(-8.2)	(-13.2)
InstructBLIP	THRONE	70.8	74.3	72.5	71.5	77.2	71.9	73.1	75.2
	POPE	82.2	74.4	78.1	80.5	85.0	70.0	77.9	82.6
	Δ	(-11.4)	0.0	(-5.5)	(-9.0)	(-7.8)	1.9	(-4.8)	(-7.4)
Otter-Image	THRONE	33.0	31.2	32.1	32.7	25.2	16.9	18.7	21.5
	POPE	66.5	34.9	45.7	56.3	70.7	17.3	35.6	49.2
	Δ	(-33.5)	(-3.7)	(-13.7)	(-23.6)	(-45.5)	(-0.4)	(-17.0)	(-27.7)
MiniGPT-4	THRONE	81.7	59.8	69.0	76.1	79.9	61.8	67.6	73.6
	POPE	89.1	58.7	70.8	80.7	88.2	60.0	70.5	78.9
	Δ	(-7.5)	1.1	(-1.7)	(-4.7)	(-8.2)	1.8	(-2.9)	(-5.4)
MiniGPT-v2	THRONE	79.0	66.6	72.3	76.2	77.6	67.0	70.4	74.0
	POPE	88.3	65.8	75.4	82.7	89.8	68.5	76.3	82.6
	Δ	(-9.3)	0.8	(-3.1)	(-6.5)	(-12.2)	(-1.5)	(-5.9)	(-8.6)
LLaVA-Mistral	THRONE	74.7	77.2	75.9	75.2	78.0	76.1	76.3	77.1
	POPE	83.8	76.6	80.0	82.2	87.8	74.6	80.0	84.4
	Δ	(-9.1)	0.7	(-4.1)	(-7.1)	(-9.8)	1.5	(-3.7)	(-7.3)
mPLUG-Owl	THRONE	55.5	71.9	62.6	58.1	66.3	68.3	65.2	65.3
	POPE	72.9	71.2	72.0	72.6	82.0	64.5	74.1	78.6
	Δ	(-17.4)	0.7	(-9.4)	(-14.4)	(-15.6)	3.7	(-8.9)	(-13.3)
LRV-Instruction-v2	THRONE	82.0	56.7	67.0	75.3	78.4	58.8	65.0	71.5
	POPE	88.6	54.8	67.7	78.9	85.0	56.2	68.8	77.4
	Δ	(-6.6)	1.9	(-0.7)	(-3.6)	(-6.6)	2.6	(-3.7)	(-5.9)
LLaVA-v1.3	THRONE	80.5	65.2	72.1	76.9	79.9	65.3	70.4	75.2
	POPE	85.5	61.6	71.6	79.3	87.9	61.7	72.6	80.8
	Δ	(-4.9)	3.6	0.5	(-2.4)	(-8.0)	3.6	(-2.2)	(-5.5)
LLaVA-v1.5	THRONE	68.1	61.0	64.4	66.6	69.9	56.4	62.2	66.8
	POPE	81.5	64.4	72.0	77.4	86.2	59.2	70.4	78.8
	Δ	(-13.4)	(-3.4)	(-7.6)	(-10.9)	(-16.2)	(-2.7)	(-8.2)	(-12.0)

Table 2. **Balanced Sampling (POPE) vs. Exhaustive Sampling (THRONE):** Applying POPE sampling to THRONE leads to an underestimation of the prevalence of Type I hallucinations regardless of LVLM.

In the main paper, we demonstrated how the sampling method used in POPE [1] leads to an underestimation of Type II hallucinations and outline a complete version of POPE (POPE-C), which shows the true extent of Type II hallucinations in LVLMs. In this section, we perform the opposite—we apply POPE type sampling to THRONE and compare the results to the complete sampling method used in THRONE as outlined in the main paper. Tab. 2 shows the results of applying POPE style sampling to THRONE, once again, applying POPE style sampling leads to a large underestimation of Type I hallucinations. POPE style sampling applied to THRONE leads to a mean underestimation of $F^{0.5}_{\text{CLS}}$ by 10.9 points compared to complete sampling, which is the default in THRONE.

3 CHAIR Overview

We present a detailed description of the CHAIR evaluation presented in [2] below. The method of CHAIR, like THRONE, attempts to capture the extent of hallucinations in free-form generated text pertaining an image, however, focuses on more traditional image captioners. Similarly to THRONE, CHAIR does not use concept-focused prompts (*i.e.* instead is focused on "Type-I") and is intended to be used only in captioning tasks. CHAIR defines a manual pipeline on-top of the annotated MSCOCO image dataset to produce a list of ground truth objects present in the scene.

Given a set of ground truth objects in an image and a model-generated image caption, CHAIR extracts the objects present in the scene via a traditional hard-rule extraction method and then attempts to map each of the predicted objects into one of the 80 class categories of MSCOCO. The mapping of the extracted objects from the caption into the class set of MSCOCO uses a pre-defined synonym dictionary for each object category. Once the objects of the predicted caption are extracted and mapped to one of the MSCOCO categories, CHAIR evaluates for "false-positive" predictions (*i.e.* hallucinations). In particular the authors introduce two variants of CHAIR, the first variant CHAIR_i quantifies the extent of hallucinated instances as,

$$\text{CHAIR}_i = \frac{|\{\text{hall. object}\}|}{|\{\text{pred. object}\}|}.$$

In this setting, hallucinations would be objects extracted from the model-generated captions that after being mapped, are still not present in the ground-truth object list for the corresponding instance. We note that CHAIR_i can be viewed as measuring the "false discovery rate" (FDR) that is $1 - P$ where P is the precision. Using Precision as the main metric, however is limiting as it does not take into account the "False Negative Rate" (FNR) which is $1 - R$ where R is the recall. This implies that by lacking recall measurements, CHAIR_i may assign high scores to short and incomplete captions which are not comprehensive in detailing the image. This is in stark contrast with the new generation of LVLM models powered by LLMs which are designed to be more exhaustive and detailed and makes the use of CHAIR_i problematic when evaluating with LVLMs. We note that CHAIR_i is the main metric used when people report "CHAIR scores", and typically reported numbers correspond to the mean CHAIR_i score across the MSCOCO validation set of images.

The second variation of CHAIR is the CHAIR_s which simply measure the number of sentences (predictions) that include at least 1 hallucination as compared with all sentences considered,

$$\text{CHAIR}_s = \frac{|\{\text{sentences w/ hall. object}\}|}{|\{\text{all sentences}\}|}. \tag{1}$$

Note that CHAIR_s does not measure the extent of hallucination within a sentence, just the existence of at least one hallucination. This is problematic as it does not capture the extent of hallucination in the sentence especially for long-form text and does not elucidate if a caption contains many or a single hallucination.

Producing ground truth in CHAIR To create a list of ground truth objects from MSCOCO annotations, the authors of CHAIR harness two annotation types to produce the most exhaustive list of ground truth objects. First the authors directly use all of the instance segmentation labels for each image, which they aggregate into a unique list of objects existing in the image. Next the authors use the 5 human-labelled captions of each image in MSCOCO and use the same extraction and mapping pipeline applied to the predictions to produce an additional set of objects that are present in the captions. Both objects lists are combined and the authors note that captioning ground truth and instance segmentation ground truth objects are often complementary as they follow different styles. Therefore combining objects from both types of object lists is beneficial for the most exhaustive final ground truth list.

Method	CHAIR == THRONE?	#Responses (%)	#Responses Evaluated	#Judgements	#Judgement Errors	Error Rate
CHAIR/THRONE	✓	38350 (69.7%)	55	157	4	2.5%
THRONE	✗	16650 (30.3%)	110	376	30	8.0%
CHAIR	✗	16650 (30.3%)	110	477	111	23.3%

Table 3. **Summary of Qualitative Evaluation:** Our qualitative results show that for responses in which THRONE and CHAIR differ, there is a large difference in the error rate. When THRONE and CHAIR agree, the error rate is small.

#Responses Analyzed	#False Positives Identified	#Hallucinations	#Misclassifications
90	71	69	2

Table 4. **Hallucinations Dominate False Positives:** Human evaluation establishes the vast majority ($\frac{69}{71} \approx 97\%$) of false positive object classes in LVLM responses are true hallucinations rather than plausible misclassifications of objects.

4 Qualitative Evaluation of THRONE

4.1 Evaluation Method

We include a self-contained file (`THRONE_qual_eval_results.html`) in the Supplementary Material, which shows the qualitative evaluation and comparison of THRONE and CHAIR. For each LVLM evaluated, we sample 10 COCO images at random in which THRONE and CHAIR *disagree* and 5 COCO images in which THRONE and CHAIR *agree*. Therefore, we qualitatively evaluate 165 responses. These results are summarized in Tab. 3

By calculating error rates for each of these cases and noting the proportion of responses in which THRONE and CHAIR *disagree* we can estimate the overall error rate of each method using a weighted sum.

$$\begin{aligned}
\textbf{Method Error Rate} &= (\text{Agreement Proportion}) \times (\text{Error Rate in Agreement Case}) \\
&\quad + (\text{Disagreement Proportion}) \times (\text{Error Rate in Disagreement Case}) \\
\textbf{CHAIR Error Rate} &= 0.697 \times 0.025 + 0.303 \times 0.233 = 0.088 = 8.8\% \\
\textbf{THRONE Error Rate} &= 0.697 \times 0.025 + 0.303 \times 0.080 = 0.043 = 4.3\%
\end{aligned}$$

4.2 Discussion

We find that the plurality of errors made in THRONE relate to a mismatch between the LM definition of a certain class and the definiton in COCO. The most clear example is in the `tv` COCO class. In COCO, this class includes computer monitors, whereas for an LM, the implication of the existence computer monitors in an LVLM response does not lead to a “`yes`” response when asked `Is there a tv in this image?` or similar. When doing an manual evaluation our human oracle *is* aware of the particular COCO class definitions and answers accordingly. Using a handcrafted rule for `tv` and other similar COCO classes, we would expect the error rate of THRONE to reduce significantly, but in THRONE we deliberately avoid the use of handcrafted rules.

As mentioned in the main paper, the errors in CHAIR are more fundamental and result due to simple text matching of synonyms not being able to discriminate between abstract concepts alluded to in a response and direct objects implied to exist in the image based on the response.

Tab. 4 shows results for human analysis of false positives. We analyze 90 responses, the 15 samples for mPLUG-Owl, MiniGPT-v2, MiniGPT-4, LLaVA-7b-v1.5, LLaVA-7b-v1.3 and InstructBLIP—the final 90 responses in the self-contained file: `THRONE_qual_eval_results.html` in the Supplementary Material.

5 Improved Baseline Implementation Details

In the main paper, we introduced a simple method to augment the LLaVA visual instruction tuning data with an object enumeration task to reduce Type I and Type II hallucinations when used to train LLaVA models. The format used for object enumeration is:

```
Instruction: <image> Give a list of objects and locations
             in the image.
Response:    {class_name_1} [{location_1}/absent]
             ...
             {class_name_N} [{location_N}/absent]
```

where `location_i` represents the location of the bounding box center point on a 3×3 grid.

We give additional details on the construction of the object enumeration task here.

5.1 Object Enumeration Implementation Details

The LLaVA visual instruction tuning data contains 157712 samples applied to 81479 images from the COCO training set (some images correspond to multiple samples). We ensure the absolute character length of our object enumerate task for a single sample is not exceedingly long—we do not want the visual instruction tuning data to be pushed outside the context length of the LLaVA model. This is done by limiting the number of instances *per class* in a sample to 3.

For each *sample* we construct an object enumeration task using bounding box data as follows: first, sort bounding box annotations for a given image by box area in descending order; second, loop over the sorted annotations adding the instance (`class_name_i, location_i`) to the object enumeration task if there are less than 3 instances of `class_name` in the task; third, sample 6 negative classes and append them to the object enumeration task using `absent` as the location string.

The sampling of negative classes is detailed next.

5.2 Negative Sampling Implementation Details

To sample negatives in the object enumeration task we first build a co-occurence matrix from the bounding box annotations. The pseudocode for building this matrix is as follows:

```
from pycocotools.coco import COCO
import numpy as np
train_dset = COCO(instances_path)
num_cats = len(train_dset.getCatIds())
co_occur = np.array((num_cats, num_cats))
cat_id2cont_id = {x: i for i, x in sorted(enumerate(train_dset.getCatIds()))}
for iid in train_dset.getImgIds():
    anns = train_dset.loadAnns(train_dset.getAnnIds(imgIds=iid))
    pres_cats = [coco_cid2cont_cid[x['category_id']] for x in anns]
    pres_cats = np.unique(pres_cats)
    for r in pres_cats:
        for c in pres_cats:
            if r != c:
                co_occur[r, c] += 1
```

After building this co-occurence matrix negative classes are sampled in a manner which is aware of the classes present in a given image. The pseudocode is as follows (using some variables from the above pseudocode):

```
present_cat_ids: List[int]  # list of category ids present in the image
present_cont_ids = [cat_id2cont_id[x] for x in present_cat_ids

# combine co-occurence across present categories
# ensuring present categories can not be sampled
present_co_occur = co_occur[present_cont_ids].copy()
present_co_occur[:, present_cont_ids] = 0
present_co_occur = present_co_occur / present_co_occur.sum(axis=1, keepdims=True)
present_co_occur = present_co_occur.sum(axis=0)
present_co_occur = present_co_occur / present_co_occur.sum()
# sharpen distribution
present_co_occur = present_co_occur ** 10
present_co_occur = present_co_occur / present_co_occur.sum()


rng = np.random.RandomState(iid)
neg_ids = rng.choice(
    sorted(train_dset.getCatIds()),
    size=6,
    p=present_co_occur,
    replace=False
)
```

This method of sampling yields negative classes which commonly co-occur with positive classes in a given image. Therefore, the object enumeration task trains the LVLM to distinguish individual objects and classes rather than relying on global context.

5.3 Object Enumeration Data Details

In Table 3 of the main paper, we present results on THRONE, POPE and POPE-C when training with our object enumeration task using COCO or COCO and VisualGenome as object enumeration data. Approximately 33000 of the 81479 COCO images in the LLaVA visual instruction tuning data are contained in the VisualGenome dataset. When using COCO *and* VisualGenome data, we construct the object enumeration task for an image from VisualGenome data when possible and COCO otherwise—we do not combine COCO and VisualGenome annotations for any image.

5.4 Inference Details

In Table 3 of the main paper, we present results on THRONE, POPE and POPE-C when training with our object enumeration task *and* performing the object enumeration task at inference. In the next section we show the effect of not performing object enumeration during inference on THRONE and POPE, instead directly addressing the relevant task.

5.5 Ablation Results

Model	Obj. Enum. Data	Obj. Enum. Negatives	Obj. Enum. Inference	THRONE			POPE		
				P_{CLS}	R_{CLS}	$F^{0.5}_{\text{CLS}}$	P	R	F^1
LLaVA-v1.3	✗	N/A	N/A	79.9	65.3	76.5	58.0	98.4	73.0
	COCO	✗	✓	82.4	69.2	79.4	64.8	95.2	77.1
	COCO + VG	✗	✓	85.8	60.4	79.1	66.0	95.3	78.0
	COCO	✓	✓	83.2	68.8	79.9	73.2	88.2	80.0
	COCO + VG	✓	✓	86.2	67.0	81.5	83.0	82.5	82.8
LLaVA-v1.5	✗	N/A	N/A	69.9	56.4	66.8	81.9	90.8	86.1
	COCO + VG	✓	✗	79.3	76.1	78.6	83.2	86.4	84.8
	COCO + VG	✓	✓	86.1	77.0	84.1	89.8	83.7	86.7

Table 5. **Effect of Negatives and Inference:** Including negatives in our object enumeration task improves performance on THRONE and POPE in terms of precision and F-score. Performing the object enumeration task at inference time improves performance on THRONE and POPE, but hampers inference time as the object enumeration task can generate long sequences.

6 Limitations

In this paper we present THRONE which is a step towards measuring and mitigating hallucinations in LVLMs, nonetheless, our work has few key limitations which we list below.

1. THRONE is concerned with *only* measuring hallucinations in LVLM predictions in the form of a *false existence of an object in a closed set of classes*. As observed in LLMs, hallucinations are much more multifaceted and include not just objects outside a pre-defined vocabulary, but also many abstract concepts such as wrong reasoning relating to a visual scene as well as wrong attributes of a particular objects or person. These additional hallucinations are not possible to be measured with THRONE without modifications.
2. The presented method of THRONE only focused on *"Type-I hallucinations"* which does not paint a complete picture of the hallucinating behavior of an LVLM. Indeed we present POPE-C in Fig. 7 to extend hallucination measurements in both Type-I and Type-II hallucinations. We present POPE-C as an extension of POPE since we observe that POPE severely undercounts hallucinations in Type-II form.
3. Due to lack of general and exhaustive ground truth object label lists for a given image, our method relies on curated datasets such as MCOCO or Object365 that have detailed annotations that are complete on an image level, which are needed for our evaluation.
4. Our method focuses only on the hallucination bias of LVLMs but does not include measurements of other types of bias of LVLM generations (*e.g.* related to concepts of fairness in generation) which we leave for future work.

7 Ethical Considerations

We present THRONE which is a general evaluation pipeline for measuring hallucinations (specifically "Type-I" hallucinations) in Large-Vision-Language Models (LVLMs). Overall we believe that our contribution is ethically positive as it measures and shows that existing public LVLMs are not yet ready to be deployed in mission critical applications, as we observe that they still suffer from hallucinating objects to a large extent. In addition we believe our presented evaluation framework also provides for a "north-star" in measuring evaluation and can aid the field and practitioners alike in measuring and making progress towards reducing evaluations in LVLMs as well as electing to use one LVLM over another. We note that measuring societal bias in LVLMs is highly important pre-requisite before their deployment, however this is not investigated in the current work.

References

[1] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the Conference on Empirical Methods in Natural Language*, 2023. 3

[2] Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. In *Proceedings of the Conference on Empirical Methods in Natural Language*, pages 4035–4045, 2018. 4