A Fine-Grained Hallucination Detection Framework for Text-to-Image Models

Anonymous ACL submission

Abstract

With the surge in popularity of Text-to-Image (TTI) models it has become crucial that we are able to quantify the reliability of such models. This "reliability" is closely related to how strictly these models are able to adhere to a given prompt and not generate incorrect/unnecessary details, also called "hallucinations". Although alot of work has gone into classifying coarse-grained hallucinations, efforts still have to be made in detecting and mitigating fine-grained or attribute-level hallucinations such as colour, number and position have not been looked at. To this end, in this paper, our contribution is multi-fold: (i) we first formalize our proposed definition of fine-grained hallucinations and describe its various types; (ii) subsequently, we propose a modularized fine-grained hallucination detection framework to detect hallucinations (including fine-grained) in TTIMs and; (iii) propose a novel metric for quantifying these hallucinations. Our pipelined framework for automatically detecting these attribute-level hallucinations consists of four sub-modules: (i) a detection and segmentation module, (ii) a dense captioning module, for generating captions for targeted regions of the image, (iii) a meta-model, which comprises of an LLM, to cohesively reconstitute the dense captions and (iv) finally, a tree-matching module, which computes targeted attribute level metrics using the syntax trees of both the input prompt and the generated meta-caption. Through extensive experiments with open-source TTIMs, using well-known datasets, we establish the efficacy and adaptability of our proposed methodology.

1 Introduction

The phenomenon of hallucination has been well-documented in recent literature, particularly within vision-language models (Rawte et al., 2024). From image captioning tasks (He et al., 2020), where models may fabricate details not present in the visual input, to generative models (Goodfellow et al., 2020; Rombach et al., 2022) like diffusion models, hallucinations manifest in varied ways. In this work, we take a deeper look at hallucinations elicited by text-to-image models. With such a surge in their popularity it becomes increasingly crucial to be able to quantify how effectively these models align with user intents and faithfully translate textual prompts into visual outputs. This necessitates the development of robust evaluation metrics to assess the fidelity, relevance, and consistency of the generated images, particularly in identifying and mitigating issues like hallucinations and misalignment with the input prompts.

Numerous efforts have been made to detect hallucinations in text-to-image models, but the majority of these focus on object-level hallucinations (Rohrbach et al., 2018), (Li et al., 2023) where the model hallucinates objects that are absent or incorrectly identifies objects. However, less attention has been given to attribute-level hallucinations also known as fine-grained hallucinations, which involve a more subtle misalignment (Li et al., 2023). Text-to-image models usually elicit a lot of hallucinated artefacts due to the task of image generation requiring a "creative license", to fill in details not mentioned in the prompt (input). But, this also opens up an avenue for unnecessary, misleading or completely wrong generations. These fine-grained hallucinations may include errors in colour, size, or other properties of an object, which can be just as detrimental, particularly in tasks that demand fine-grained precision.

To this end, in this work, we explore different fine-grained hallucinations within text-to-image models, identifying the different types that commonly occur and their impact. Subsequently, we propose a modularized framework to detect fine-grained hallucinations. Furthermore, we introduce a novel metric to quantify these hallucinations, aiming to provide a more nuanced evaluation of model performance in generating aligned and co-

Figure 1: Example of fine-grained hallucinations from TTIMs and its different types: (a) Color & Number; (b) Number; (c) Position; and (d) Text

herent outputs. Through extensive experiments using open-source text-to-image models spanning well-known datasets and ablation performance we demonstrate the efficacy of our proposed approach.

The **contributions** of this paper is multi-fold: (i) we first formalize our proposed definition of fine-grained hallucinations and describe its various types; (ii) subsequently, we propose a modularized methodology to detect hallucinations (including fine-grained) in TTIMs; (iii) propose a novel metric for quantifying these hallucinations; (iv) Through extensive experiments using open-source TTI-models, we establish the efficacy and adaptability of our proposed methodology.

2 Related Works

Hallucinations in Large Vision-Language Models (LVLMs) (Liu et al., 2024) have garnered significant attention due to their impact on the reliability and utility of generated content in real-world applications. Vision-language hallucinations refer to cases where the generated visual output includes elements or attributes that were not specified in the input prompt. This phenomenon is problematic for tasks like image generation (Goodfellow et al., 2020; Rombach et al., 2022), captioning (He et al., 2020), and multimodal understanding (Yue et al., 2024), where maintaining semantic alignment between text and image is crucial.

Previous research has explored hallucinations in various forms, however most of them focus on image captioning task. Several studies have identified common categories of hallucinations (Rawte et al., 2024), such as incorrect object generation, identity incongruity, and the inclusion of extraneous details not aligned with the input prompt. Models such as DALL-E (Ramesh et al., 2021), Imagen (Saharia et al., 2022), and Stable Diffusion (Rombach et al., 2022) have shown to occasionally produce fine-grained hallucinations, where specific attributes like color, size, or orientation deviate from the original prompt description. For instance, Saharia et al. highlighted the challenges of controlling object attributes in diffusion-based models, pointing to the limitations in current image-text alignment mechanisms.

Recent approaches to mitigating these visual-hallucinations have focused on object instance level metrics, although they prove to be a good signal to quantify how well the model follows the given prompt, larger concerns arise when the hallucination is on the attribute level instead of object level. Object level metrics focus on quantifying how many objects the TTIM has incorrectly generated (Rohrbach et al., 2018) or by repeatedly querying a large multi-modal model about the image (Li et al., 2023; Chen et al.). More recent works like (Chen et al., 2024) focus on using powerful closed-source model APIs as multi-modal question answering tools. Using these, they query a very large model about the contents of the generated image at the attribute level. Although effective, the use of such closed-source models make these pipelines prohibitive to researchers and users with resource constraints.

Despite these advances, existing techniques often fail to capture and quantify hallucinations at a fine-grained level and the methods which try to are very expensive to use. Most prior methods have centered around overall alignment metrics, which do not provide detailed insights into individual attribute-level hallucinations, such as color, shape, or position mismatches. Our proposed fine-grained hallucination detection framework/metric builds on these foundations by offering a more granular perspective, targeting specific attributes to

identify and quantify hallucinations in TTIMs.

3 Definition: Fine-Grained Visual Hallucinations

Text-to-image models, such as DALL-E (Ramesh et al., 2021) and Stable Diffusion (Rombach et al., 2022), have gained significant attention for their ability to generate high-quality images from textual descriptions. Despite their impressive capabilities, these models often exhibit a critical limitation: *"hallucination"* refers to instances where the output image contains elements that do not align with the input text/prompt or are misrepresented.

Fine-grained hallucinations or attribute-level hallucinations are subtle and intricate discrepancies between the generated image and the textual prompt. These inaccuracies can manifest in various forms, including erroneous object attributes, misinterpreted spatial relationships, incorrect textures, and compositional inconsistencies. For example, a model may misrepresent the color of an object, introduce unnecessary elements, or misplace objects relative to one another. These hallucinations are particularly problematic in cases where precision is crucial, such as medical imaging, architectural design, and scientific visualization. Understanding and addressing fine-grained hallucination is critical for improving the fidelity of text-to-image models. To this end, we categorise the various types of fine-grained hallucinations commonly seen in such models (taken into consideration in this paper). This is by no means an exhaustive list, these kinds of hallucinations are highly context-dependent, making it difficult to categorize them into fixed types. However, a few common types are as follows:

• **Color:** This type of hallucination occur when the model misinterprets the hue or colour of an object. For example, a model might generate "a red apple" when the input text specifies "a green apple". This type of hallucination can also manifest in gradients, with objects appearing in unexpected or unrealistic shades that deviate from the prompt's intended palette. Figure 1a shows an example of this type.

• **Number:** This type of hallucination involves the model generating an incorrect number of objects based on the prompt. For example, when a prompt specifies "two dogs", the model might produce "one", "three", or "an indeterminate number". This discrepancy is particularly noticeable in prompts requiring exact counts, such as "a row of five trees," where these models struggle to meet the specified quantity. Figure 1a & b show examples of this type.

• **Position:** Position-based hallucinations refer to inaccuracies in the spatial placement of objects within the image. This can include incorrect positioning relative to other objects, such as "placing a chair on top of a table" instead of "beside it". Misunderstanding spatial relationships between objects such as- "above", "below", or "next to"—can break the coherence of the generated image, especially in complex scenes. Figure 1c shows an example of this type.

• **Text:** Hallucinations involving text are common when models are tasked with generating written language within images, such as labels, signs, or captions. Models often produce gibberish, misspellings, or irrelevant words that bear little resemblance to the input prompt. Even when the model attempts to capture the meaning, the specific text can be flawed due to poor understanding of the symbolic representations or limitations in rendering readable typography. Figure 1d shows an example of this type.

4 Methodology

We propose a modularised pipeline to detect and quantify fine-grained visual hallucinations in an image generated from a TTIM given a text input. As fine-grained hallucinations are quite varied in their occurrence and depend heavily on the user, it is not possible to formulate a unified method of quantifying all of them. However, our proposed pipeline can easily be expanded to fit niche use-cases as well. We also provide guidelines to formulating an algorithm to detect a specified type of hallucination. Subsequently, we also introduce a novel metric to measure fine-grained hallucination. Our method exclusively uses open-sourced models for each module, making it accessible to anybody. This also means that for each module users can decide on how big of a model to use as per their resource constraints.

4.1 Fine-Grained Hallucination Detection Framework

The proposed pipelined framework is based on the textual mapping of the original text input to the generated description of the hallucinated image (which is generated using TTIMs). Our pipeline consists of four submodules: (1) Detection & Seg-

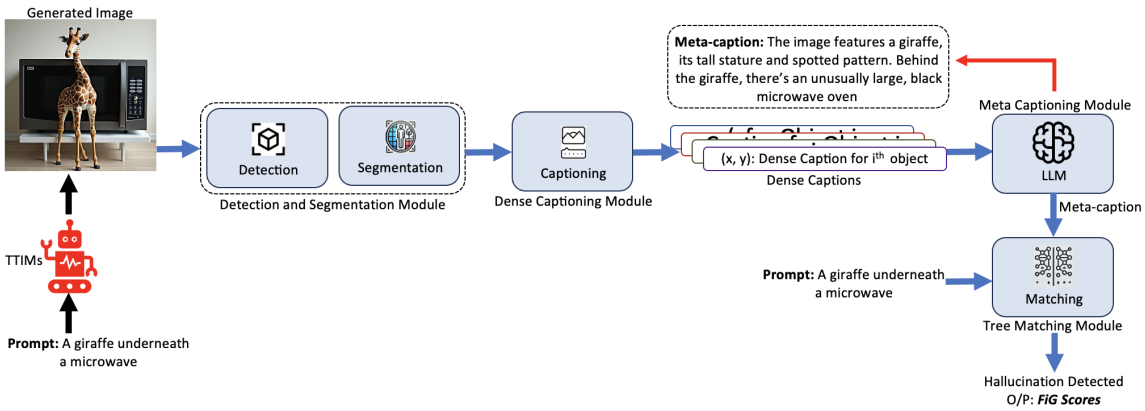

Figure 2: The architectural representation of the proposed fine-grained hallucination detection framework

mentation Module; (2) Dense Captioning Module; (3) Meta-Captioning Module; and (4) Tree Matching Module. Our hypothesis behind this pipelined methodology is that the dense caption being richer in information produce more detailed captions and also lead to a reduction in hallucination in by the captioning models as well. To facilitate creation of these dense captions we use a detection module, to create segments of the image which contain reference to objects mentioned in the input prompt.

The details of each sub-module is discussed below.

• **Detection & Segmentation Module.** This module consists of two models, namely, the detector and the segmenter. These models work in succession to produce segments of the image. These targeted segments reduce the amount of distractor-information for the subsequent modules. Given a *generated image*, I of size (nxn), and an accompanying prompt S, we define the segments created as: (i) Noun phrases, $P_1, P_2, \ldots, P_k$ are extracted from prompt, S using a NER model. Each noun phrase P_i is input into a detection model, D to obtain a corresponding bounding box, $B_i = D(P_i)$. Finally, each bounding box, B_i is passed into the segmentation model, M to obtain the corresponding segments, $S_i = M(B_i)$. For extracting noun phrases from prompts, we used a small instruction model, Phi-3-mini (Abdin et al., 2024), to ensure no target observation was missed. These noun-phrases were then fed into the detection model (prompted ones) to create bounding-boxes for their localisation. For creating segments of the detected objects, we use the Segment Anything Model from Meta (Kirillov et al., 2023). This is because this model takes bounding box coordinates as inputs to create the segments.

• **Dense Captioning Module.** For each segment, $S_i \in S$, captions are generated, $DenseCaption_i = DCM(P_1, S_i)$, using an image-to-text captioning model, where P_1 refers to the prompt given to the model. These captions are rich in information about the target object as a lot of unnecessary details have been removed by the first module. The model used in this section is prompted to generate detailed captions with regards to all the fine-grained features considered in this paper. Although, these captioning models do tend to hallucinate irrelevant details, we observed that giving them targeted segments of an image mitigates this issue. This is, in essence, due to them being less informative to hallucinate on account of only one object being present.

• **Meta-Captioning Module.** Once the dense captions have been generated for each object in the image, the task at hand is to stitch these together into a coherent caption accurately and intricately describing the image. Along with this, a signal is passed for the position of the object in the image. This is done by prompting the LLM with the generated caption and the median of each object. For each input sequence of the form $F_i = ((x_i, y_i), D_i) \in F$, where $D_i \in D$ and its corresponding median (x_i, y_i), $MetaCaption_i = MCM(P_2, F)$. This facilitates the underlying LLM to spatially reason about the entire image and thus, produce a caption that captures both visual and spatial details of the objects. The medians are calculate for each segment S in the following way:

$$\text{Median}_x = \frac{1}{|S|} \sum_{(x,y) \in S} x$$

$$\text{Median}_y = \frac{1}{|S|} \sum_{(x,y) \in S} y$$

where $\left|S\right|$ is the number of pixels in each segment S

• **Tree Matching Module.** The resultant meta-caption now contains a fine-grained description of the original image which is parsed into its syntax tree via any natural language processing library (spaCy (Altinok, 2021) is used in our case). Once both the meta-caption and original prompt have been converted into their tree structure, we can quantify the similarity and/or hallucination between both.

4.2 Fine-Grained Hallucination Score (FiG)

We propose a **Fi**ne-**G**rained Hallucination score, named ***FiG*** to quantify and measure attribute-level hallucinations in TTIMs. Our metric is computed on three levels of granularity, we call them Stage-1, Stage-2 and Stage-3 metrics which are detailed below:

• **Stage-1 Metric:** This deals with the coarse hallucinations typically objects that is found in the TTIM generated outputs. It signals how many objects mentioned in the prompt were correctly generated by the model, and is computed as the object level recall. More formally,

$$S1(orig, meta) = \frac{|objects(orig) \cap objects(meta)|}{|objects(orig)|}$$

where $objects(\cdot)$ is the set of distinct objects mentioned in the sentence. This metric indicates how well the image-generator model has done in representing each individual object that was mentioned in the prompt.

• **Stage-2 Metric:** The Stage-2 metric is designed to capture fine-grained inconsistencies between two descriptions (original and generated) by providing a set of scores for each attribute under analysis. This approach allows for targeted evaluation of specific attributes, such as color, position etc. in the generated image. For instance, if we aim to detect hallucinations related to color and position, we can compute two separate Stage-2 metric scores—one for each attribute. Algorithm 1 & 2 outline the general procedure for calculating the Stage-2 metric for any given attribute. In this paper, we focus on four common types of fine-grained hallucinations observed in generative models: color, position, text, and number. However, the proposed metric is flexible and can be extended to capture hallucinations related to other attributes, depending on the use case.

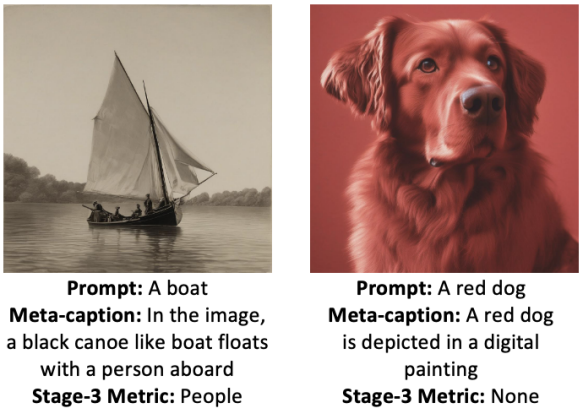

Figure 3: Demonstrating the usability of Stage-3 FiG Metric

• **Stage-3 Metric:** This metric plays a critical role in flagging potential hallucinations by listing extra, unprompted objects found in the meta-caption (i.e., the generated caption that describes the TTIMs output). Unlike numerical metrics that directly quantify performance, this metric serves as a qualitative signal, aimed at ensuring that generative models do not introduce a disproportionate number of extraneous elements into their outputs. The Stage-3 metric is computed by extracting all noun phrases from the meta-caption using NER techniques. These noun phrases, representing objects, are then compared against those mentioned in the original prompt. Any object or entity appearing in the meta-caption but absent from the original prompt is flagged. Although the number of such objects is not directly tied to a performance score, their presence can be indicative of a model's tendency to hallucinate irrelevant details. This is evident in Figure 3, where the Stage-3 metric correctly extracted "people" as a possible hallucination. More formally, it can be calculated as the set difference between objects in the meta-caption and the original prompt.

$$S3(orig, meta) = objects(meta) - objects(orig)$$

5 Experimentation Details

In this section, we deliberate on the experimentation that were carried out with regards to the proposed framework and the metric, i.e., FiG Score, the datasets used and the different models taken into consideration. To the best of our knowledge, no other metric directly tackles fine-grained hallucinations effectively, thus making comparisons

Table 1: Performance of popular open-source text-to-image models on the proposed framework. (↑)-Higher score is better, (↓)-Lower Score is better

Model(s)	DrawBench						MSCOCO					
	Stage-1 (↑)	Stage-2 (↑)				$CHAIR_i$ (↓)	Stage-1 (↑)	Stage-2 (↑)				$CHAIR_i$ (↓)
	Object FiG	*Colour FiG*	*Number FiG*	*Positional FiG*	*Text FiG*		*Object FiG*	*Colour FiG*	*Number FiG*	*Positional FiG*	*Text FiG*	
SDXL 1.0	39.39	47.43	31.12	10.12	28.57	50.10	35.90	41.30	29.20	15.12	N/A	48.10
SD-2	42.79	34.61	32.03	5.23	9.52	50.10	25.70	24.12	19.96	10.30	N/A	50.10
Flux-Dev	46.92	62.82	40.51	20.83	23.80	24.70	34.05	28.80	26.80	12.30	N/A	43.90

Table 2: Impact in performance by the variation of Detection Models. (↑)-Higher score is better, (↓)-Lower Score is better

Model(s)	DrawBench						MSCOCO					
	Stage-1 (↑)	Stage-2 (↑)				$CHAIR_i$ (↓)	Stage-1 (↑)	Stage-2 (↑)				$CHAIR_i$ (↓)
	Object FiG	*Colour FiG*	*Number FiG*	*Positional FiG*	*Text FiG*		*Object FiG*	*Colour FiG*	*Number FiG*	*Positional FiG*	*Text FiG*	
Grounding-DINO	39.39	47.43	31.12	10.12	28.57	50.1	42.12	48.47	37.82	15.23	N/A	44.1
YOLOv8	32.49	46.43	26.10	10.12	14.28	55.12	40.89	47.33	38.23	15.23	N/A	43.2

difficult. However, we report the $CHAIR_i$ metric to evaluate the quality of meta-captions generated during the pipeline.

5.1 Datasets

To evaluate the performance of our proposed fine-grained hallucination framework & metric, we use two well-known datasets: the DrawBench dataset (Saharia et al., 2022) and the MSCOCO Captions dataset (Chen et al., 2015).

The DrawBench dataset (Saharia et al., 2022) consists of 200 prompts that are specifically curated to provoke fine-grained hallucinations in TTIMs. Each prompt is carefully designed to challenge models in capturing subtle details such as color, shape, and spatial relationships, making it an ideal benchmark for assessing fine-grained inconsistencies. This dataset provides a controlled environment where the hallucination of certain attributes can be systematically measured.

In addition to DrawBench, we leverage the MSCOCO Captions dataset (Chen et al., 2015). Although this dataset is traditionally used for image captioning, it can be effectively repurposed for image generation by treating the captions as prompts. The dataset contains over 500,000 captions, offering a vast variety of scenarios that could lead to attribute-based hallucinations. For the purposes of our experiments, we randomly select a subset of 50,000 captions, ensuring that the dataset remains diverse while being computationally manageable.

Together, these datasets form a robust testing ground for detecting fine-grained hallucinations across different generative models. The controlled nature of the DrawBench prompts, combined with the scale and diversity of MSCOCO captions, ensures that our metric is evaluated on a wide range of potential hallucination cases.

5.2 Evaluation

Along with the FiG metric introduced in this paper, we also report results on the $CHAIR_i$ metric (Rohrbach et al., 2018) on all the experiments performed. The $CHAIR_i$ score is an hallucination detection metric for image captioning tasks. Given the ground truth objects in the image, CHAIR calculates the proportion of objects that appear in the caption but not the image. Existing work commonly adopts its two variants, i.e., $CHAIR_I$ and $CHAIR_S$, which evaluate the hallucination degree at the object instance level and sentence level respectively. They are be formulated as:

$$CHAIR_i = \frac{|\{\text{hallucinated objects}\}|}{|\{\text{all mentioned objects}\}|}$$

$$CHAIR_s = \frac{|\{\text{captions with hallucinated objects}\}|}{|\{\text{all captions}\}|}$$

As the CHAIR metric is used for detecting hallucinations in image captioning tasks, it can be used to judge the quality of the generated meta-captions. However, as this metric only works on object level, it cannot be directly compared to the FiG metric proposed in this paper.

6 Results and Analysis

In this section, we analyze the performance of our proposed fine-grained hallucination detection framework using the FiG and CHAIR metrics, on several open-source TTIMs focusing on the impact of model selection within the different modules of our proposed pipeline. We assess the effectiveness

Table 3: Impact in performance by the variation of Dense Captioning Models. (↑)-Higher score is better, (↓)-Lower Score is better

Model(s)	DrawBench						MSCOCO					
	Stage-1 (↑)	Stage-2 (↑)				$CHAIR_i$ (↓)	Stage-1 (↑)	Stage-2 (↑)				$CHAIR_i$ (↓)
	Object FiG	*Colour FiG*	*Number FiG*	*Positional FiG*	*Text FiG*		*Object FiG*	*Colour FiG*	*Number FiG*	*Positional FiG*	*Text FiG*	
InstructBLIP-Vicuna 7b	46.92	62.82	40.51	10.23	23.80	24.7	44.22	59.64	37.82	12.23	N/A	43.90
mPLUG-OWL	44.41	28.73	29.43	2.81	47.61	50.1	40.21	24.33	28.32	12.53	N/A	40.32
BLIP2	46.65	32.18	45.07	19.74	42.85	24.7	47.68	30.24	42.07	19.01	N/A	50.21

Table 4: Impact in performance by the variation of Meta-Captioning Models. (↑)-Higher score is better, (↓)-Lower Score is better

Model(s)	DrawBench						MSCOCO					
	Stage-1 (↑)	Stage-2 (↑)				$CHAIR_i$ (↓)	Stage-1 (↑)	Stage-2 (↑)				$CHAIR_i$ (↓)
	Object FiG	*Colour FiG*	*Number FiG*	*Positional FiG*	*Text FiG*		*Object FiG*	*Colour FiG*	*Number FiG*	*Positional FiG*	*Text FiG*	
Qwen-7b	46.92	62.82	40.51	20.83	23.80	24.70	34.05	28.80	26.80	12.30	N/A	43.90
Mistral-v2 7b	44.83	59.77	32.71	19.48	52.38	22.12	32.82	27.23	25.13	11.24	N/A	40.21
Llama-3.1 8b	41.07	45.97	32.30	19.23	52.38	23.90	31.23	29.21	22.12	16.23	N/A	44.54
NeMo-Minitron 8b	49.23	57.47	27.29	12.12	52.38	24.45	31.75	35.71	28.01	9.82	N/A	50.10

of these models in capturing specific types of hallucinations in TTIMs and highlight areas where performance is suboptimal. Additionally, we provide a detailed analysis of the FiG metric's robustness across various generative models, identifying key shortcomings and offering insights into potential improvements for future iterations. By examining the interplay between model choice and hallucination detection, we discovered the best performing pipeline to consist of: (1) Grounding-DINO as the detection model, SAM as the segmentation model; (2) Instruct-BLIP as the dense captioning model; and (3) Qwen-7b as the meta-captioning model.

Comparison of Different Text-To-Image Models. We performed experiments on three TTIMs: SDXL (Podell et al., 2023), Stable Diffusion 2.0 (SD-2) (Rombach et al., 2021), and FLUX[1], each of which varies in architecture, training data, and fine-tuning strategies.

As evident by Table 1, FLUX outperformed the other two models by consistently generating images with minimal fine-grained hallucinations. FLUX exhibited strong alignment with the original prompts, particularly in attributes like color, number, and position. Its robust performance suggests that its underlying architecture and training setup allow it to handle intricate prompt details better, leading to fewer erroneous visual objects in the generated images. In contrast, SD-2 struggled the most in our evaluation. The high hallucination rate in SD-2 may be attributed to its training data, indicating that while it performs well in generating aesthetically pleasing images, it may not be as well-equipped to manage the nuanced, fine-grained relationships required for precise text-to-image alignment.

Impact of Detection Models. We experimented with two detection models: Grounding-DINO and YOLOv8. Grounding-DINO consistently outperformed YOLOv8 in our experiments (see Table 2). This performance boost can be attributed to the design of Grounding-DINO, which allows prompting the model to specifically target and detect each object of interest. This capability of focusing on specific objects aligns well with the goals of fine-grained hallucination detection, where the goal is not only to detect objects but to ensure precise alignment with the original prompt. In contrast, YOLOv8 operates as a general object detector, attempting to detect all objects in an image regardless of the task at hand. While YOLOv8 is robust in general object detection tasks, it lacks the fine-tuned capability to prioritize specific objects of interest. As a result, it occasionally misses objects that are present but were not part of its original training. This generalization gap causes YOLOv8 to struggle with certain nuanced cases of object hallucinations, which require a more targeted approach for accurate detection. Thus, for applications involving the fine-grained detection of hallucinations, particularly where precise alignment between the prompt and generated image is crucial, Grounding-DINO proves to be the superior model.

Impact of Dense Captioning Models. The dense captioning model is responsible for capturing the

[1] https://huggingface.co/black-forest-labs/FLUX.1-dev

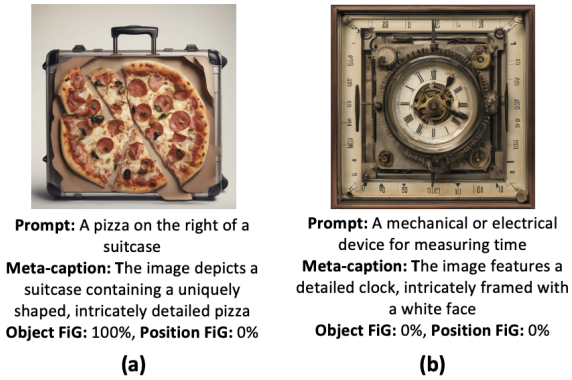

Figure 4: Use Case: (a) Highlighting the strength of the proposed framework/metric; (b) Depicting the weakness of the proposed method

detailed attributes of each segmented object, and these details are crucial for generating accurate meta-captions, which our metric uses to assess hallucinations. In our experiments, we evaluated three dense captioning models: InstructBLIP, mPLUG-OWL, and BLIP2. Amongst these, InstructBLIP and BLIP2 demonstrated the best overall performances, although each model excelled in different metrics, as shown in Table 3. InstructBLIP achieved higher scores in Object FiG and Colour FiG scores, indicating that it is more effective at accurately identifying and describing objects and their colors within generated images. This can be attributed to InstructBLIP's stronger focus on object recognition and attribute-level captioning. On the other hand, BLIP2 outperformed InstructBLIP in terms of Number and Text FiG scores. These attributes are critical for detecting hallucinations in scenarios where precise spatial and numerical alignment with the prompt is essential.

Impact of Meta-Captioning Models. The meta-captioning model plays a crucial role in our hallucination detection pipeline, as it synthesizes individual object captions into a coherent meta-caption, capturing the overall scene and relationships between objects. This model is particularly important for calculating the Positional FiG score, which relies on accurately deducing the relative positions of objects based on their median coordinates from the segmented output. We evaluated four models for this task: Mistral 7b, Llama 3.1 8b, NeMo 8b, and Qwen 7b. Amongst these, Qwen 7b consistently outperformed the others in terms of accurately generating cohesive meta-captions and deducing the relative positions of objects. This can be seen clearly in Table 4.

Error Analysis. The pipeline, though effective still is not perfect. To provide a clearer understanding of the strengths and limitations of our fine-grained hallucination detection pipeline, we present two example cases: one where the pipeline successfully detected all hallucinations and another where it failed to capture certain fine-grained details. As seen in Figure 4(a), our proposed FiG metric is successfully able to capture an incorrect generation by the model. The prompt specifies the relative positions of both the pizza and suitcase, however, the underlying TTIM ignored that. Due to this, the FiG Score for the Positional attribute computes to 0% correctly signaling an erroneous generation. However, our pipeline is not full-proof as seen in Figure 4(b), the input prompt was a description of a clock, but as it did not explicitly mention the word "clock", our metric was not able to capture the correct generation and gave it a poor FiG score. This outlines a drawback of our pipeline, namely, that it cannot match words/phrases which are semantically similar to each other but not exactly the same. Refer to the appendix A to see more examples and analysis of our pipeline.

7 Conclusion

In this paper, we introduce a novel fine-grained hallucination detection framework and metric for text-to-image models, addressing a critical gap in the evaluation of visual hallucinations in generative models. Our approach offers a detailed, attribute-level analysis of hallucinations, moving beyond traditional, coarse-grained metrics. By breaking down hallucinations into categories such as object presence, color, position, number, and text, our metric provides a more comprehensive understanding of the types of hallucinations that occur in generated images. Through extensive experimentation, we demonstrated the versatility of our approach by testing it across various generative models, dense captioning models, detection models, and meta-captioning systems. Our findings shows that model choice in each of the modules of the pipeline significantly impacts the overall performance. We also conducted an error analysis to highlight both the strengths and limitations of our pipeline, revealing that while it is effective in detecting a wide range of fine-grained hallucinations, there are still challenges in areas like shape detection and segmentation.

8 Limitations

While the FiG Metric proves to be effective at quantifying the fine-grained hallucinations in TTIMs, it does not completely eliminate the issue. The use of multiple modules makes it prone to the biases of each of the models used. While the pipeline is effective, finding each model to fit your use-case can be an arduous task. Our pipeline also faces difficulty when multiple objects overlap each other thereby causing their gaps or fractures in their segmentations. This prevents the dense captioning model from understanding what the original object was and therefore causing a drop in the Object FiG score. Another limitation we observed occurring in during our experimentation was that the captioning models refuse to generate names of institutions, famous characters or personalities. This could be because these models have been trained not to generate certain words due to licensing issues. Due to this, our pipeline gives erroneous results when the input prompt contains these types of nouns.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Duygu Altinok. 2021. *Mastering spaCy: An end-to-end practical guide to implementing NLP applications using the Python ecosystem*. Packt Publishing Ltd.

Xiang Chen, Chenxi Wang, Yida Xue, Ningyu Zhang, Xiaoyan Yang, Qiang Li, Yue Shen, Lei Liang, Jinjie Gu, and Huajun Chen. 2024. Unified hallucination detection for multimodal large language models. *Preprint*, arXiv:2402.03190.

Xiang Chen, Chenxi Wang, Ningyu Zhang, Yida Xue, YUE SHEN, GU Jinjie, Huajun Chen, et al. Unified hallucination detection for multimodal large language models. In *ICLR 2024 Workshop on Reliable and Responsible Foundation Models*.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. 2015. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144.

Sen He, Wentong Liao, Hamed R Tavakoli, Michael Yang, Bodo Rosenhahn, and Nicolas Pugeault. 2020. Image captioning through image transformer. In *Proceedings of the Asian conference on computer vision*.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 292–305.

Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou, Rongjun Li, and Wei Peng. 2024. A survey on hallucination in large vision-language models. *arXiv preprint arXiv:2402.00253*.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2023. SDXL: improving latent diffusion models for high-resolution image synthesis. *CoRR*, abs/2307.01952.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr.

Vipula Rawte, Anku Rani, Harshad Sharma, Neeraj Anand, Krishnav Rajbangshi, Amit Sheth, and Amitava Das. 2024. Visual hallucination: Definition, quantification, and prescriptive remediations. *arXiv e-prints*, pages arXiv–2403.

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

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2021. High-resolution image synthesis with latent diffusion models. *Preprint*, arXiv:2112.10752.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep

language understanding. *Advances in neural information processing systems*, 35:36479–36494.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567.

A Appendix

A.1 Algorithms

Here, we expand on the way we compute the FiG Metric as described in the paper. The first algorithm 1 is how we extract $(noun, attribute)$ pairs from a given statement. The second algorithm 2 describes how these extracted pairs are used to compute the FiG metric for each attribute.

Algorithm 1 Attribute Pair Extraction

Input: Attribute A, Sentence S
Output: List of attribute pairs `attr_pairs`
`attr_pairs` $\leftarrow$ []
for each noun n in `nouns`(S) **do**
    if A exists as a child in the subtree of n **then**
        Append (n, A) to `attr_pairs`
    end if
end for
return `attr_pairs`

Algorithm 2 Fine-Grained Attribute Consistency

Input: List of target attributes `attributes`, Meta-caption `meta`, Original caption `orig`
Output: Consistency scores for each attribute
`scores` $\leftarrow \{\}$
for each attribute A in `attributes` **do**
    `meta_attr` $\leftarrow$ `Algorithm1`$(A,$ `meta`$)$
    `orig_attr` $\leftarrow$ `Algorithm1`$(A,$ `orig`$)$
    `scores`$[A] \leftarrow \frac{|\texttt{orig\_attr} \cap \texttt{meta\_attr}|}{|\texttt{orig\_attr}|}$
end for
return `scores`

A.2 Additional Examples

In this subsection, we include a few additional examples of the successes and failures of our proposed pipeline to provide a better view of how the framework functions.

In Figure 5, we see that the FiG metric is correctly able to identify when the generated image follows the input prompt correctly. More specifically, we see in 5(c), that the Object FiG and Positional FiG are correctly able to indicate a mistake in the generation.

Figure 6 shows the limitations of our pipeline, in 6(a) we see that even though the Text FiG is 0%, the meta-caption consists of the word "Google", even though it is not written in the picture anywhere. This may lead to a mis-identification of correct words by the Stage-3 metric. This might be happening because the captioning model is trying to fill in the gaps and assume what was supposed to be written in the text by identifying the letters individually. In the second example 6(b), we see even though the model has correctly generated an image of 'Darth Vader', the captioning model is not able to identify him. We have seen through our experiments that captioning models rarely ever generate proper nouns or names of certain institutions and franchises.

Prompt: A yellow vase placed on the right of a red book
Meta-caption: The image displays a red book, catching attention with its distinct color and rectangular form, complemented by a yellow vase on its right with a singularly detailed shape
Colour FiG: 100%, **Position FiG:** 100%

(a)

Prompt: A pink car
Meta-caption: The image shows a distinctive pink classic car with a large front grille
Colour FiG: 100%

(b)

Prompt: A wine glass on top of a dog
Meta-caption: The image showcases a bottle of wine, and a wine glass filled with red liquid
Object FiG: 50%, **Position FiG:** 0%

(c)

Figure 5: Image showing examples where the FiG metric was able to correctly quantify the inconsistencies between the image and input prompt

Prompt: A sign that says "Google Research Pizza Cafe"
Meta-caption: The image shows a street light against a brick wall, with "Google Rea Cafe Rescue" written in contrasting white
Text FiG: 0%

(a)

Prompt: Darth Vader playing with raccoon in Mars during sunset
Meta-caption: The image depicts a man in a black and white illustration, wearing a cape and wielding a lightsaber while seated on the ground
Object FiG: 0%

(b)

Figure 6: Image showing examples where the FiG metric was not able to correctly quantify the inconsistencies between the image and input prompt