From Uncertainty to Trust: Enhancing Reliability in Vision-Language Models with Uncertainty-Guided Context Dropout

Yixiong Fang[*♣] Ziran Yang[*♢] Zhaorun Chen[♠] Zhuokai Zhao[♠] Jiawei Zhou[♡]
[♣]Shanghai Jiaotong University, [♢]Peking University
[♠]University of Chicago, [♡]Stony Brook University

Abstract

Large vision-language models (LVLMs) have demonstrated impressive capabilities in multimodal tasks but remain prone to misinterpretations of visual inputs, often resulting in hallucinations and unreliable outputs. To address these limitations, we propose DROPOUT DECODING*, a novel inference-time approach that quantifies the uncertainty of input visual tokens and uses them to enhance decoding. Our method measures the uncertainty associated with each visual token by projecting it onto the text token space and decomposing the uncertainty into aleatoric and epistemic components. We focus on epistemic uncertainty as it more accurately captures the visual challenges for the model, related to perception-related errors. Then inspired by dropout techniques, we introduce uncertainty-guided token dropout, applying the idea of dropout to context tokens rather than model parameters. By token dropout,* DROPOUT DECODING *aggregates an ensemble of candidate predictions that robustly mitigates individual errors from the perception of visual tokens. Evaluations on established benchmarks, including CHAIR, THRONE, and MMBench, demonstrate the effectiveness of our approach in both reducing object hallucinations and improving the quality and reliability of model outputs, enabling LVLMs to perform accurately and consistently across diverse visual contexts.*

1. Introduction

Recent advancements in large vision-language models (LVLMs) have demonstrated impressive capabilities [11, 15, 50, 55, 57, 60], in tasks like image captioning [54], visual question answering (VQA) [1, 21], and multimodal reasoning [6, 35, 37, 58]. However, LVLMs still face challenges in accurately perceiving and interpreting visual inputs, leading to inaccurate outputs and hallucinations [33]. These issues often stem from LVLMs misrepresenting key image elements or overlooking critical details, compromising the reliability of their outputs in tasks demanding precise visual understanding [4, 5, 16, 53].

In practice, LVLMs typically process visual inputs token by token, which we refer to as *visual tokens*, which can fall short in effectively focusing on the most informative parts of the visual context. While attention mechanisms are designed to prioritize relevant information, they are not always perfect [43, 48], especially when the inputs are complex or ambiguous for the model, or in other words, of high *uncertainty*. Existing methods to address these challenges in the training stage often involve fine-tuning on specific tasks [31, 32, 45, 52], or using additional supervision signals especially at lower level to guide the model [7, 51]. However, these approaches are resource-intensive and not easily extensible to new tasks. Alternative inference-time strategies, such as attention-based or logits-based mechanisms on decoding correction [4, 20, 42, 47, 56], attempt to identify important regions in the input without additional training, but they typically rely on heuristic design choices and largely increase inference cost. Therefore, enhancing the trustworthiness of LVLMs and reducing hallucinations requires more principled methods that can more effectively emphasize the most informative parts of the visual input.

To address this, we propose a novel approach that quantifies uncertainty on the input visual tokens context directly at inference time. Inspired by traditional dropout techniques, which are challenging to apply to pretrained LVLMs [13, 23], we propose a novel approach: *token dropout*, applied the idea of dropout to the context tokens instead of model parameters. Our method measures the uncertainty associated with each visual token by projecting it onto the text token space and decomposing the uncertainty into *aleatoric* (data-related) and *epistemic* (model-related) components [19, 40, 46]. By focusing on the epistemic uncertainty, which reflects the model's lack of knowledge, we identify visual tokens that the model is less confident about.

During , we adjust the visual inputs to selectively suppress the influence of high-uncertainty visual tokens based

[*] equal contribution

on epistemic uncertainty. Specifically, we generate an ensemble of predictions by creating multiple subsets of the visual inputs, each time *dropping out* different combinations of high-epistemic-uncertainty tokens. Each subset is used to produce a candidate output, and the final prediction is obtained by aggregating these candidates through majority voting. This process, which we term **DROPOUT DECODING**, enhances the reliability and accuracy of the generated outputs without modifying the underlying model parameters or requiring additional training.

By introducing DROPOUT DECODING, our contribution can be summarized as follows:

- Introduction of a novel approach that quantifies and decomposes uncertainty on tokens in the visual inputs into aleatoric and epistemic components at inference time without additional supervision.
- A decoding strategy that uses epistemic uncertainty measurements to guide the selective dropout of high-uncertainty visual tokens in the context, analogous to performing dropout on the model but applied to the input tokens during inference.
- Comprehensive experiments showing significant reductions in object hallucinations and improved fidelity in pre-trained LVLMs without additional fine-tuning.

2. Preliminaries

2.1. Vision-Language Model Decoding

We consider a general LVLM architecture adopted widely [27, 30, 31], which typically includes a vision encoder, a vision-text interface module, and a Transformer-based LLM decoder. As we mostly focus on the decoder side inference optimization, we assume the LLM decoder is with parameter θ.

The visual input, such as an image, is segmented into patches and processed by the vision encoder,[1] followed by the vision-text interface module, to produce a sequence of *visual tokens* $x^v = (x_1^v, x_2^v, \ldots, x_N^v)$. Each token x_i^v is a contextualized embedding of an image patch, serving as the direct input to the text decoder. The text input such as a query or instruction is $x^t = (x_1^t, x_2^t, \ldots, x_M^t)$. The input to the text decoder is denoted as $x = [x^v, x^t]$, which is the concatenation of visual and text tokens. At this point, the visual and text tokens are aligned and serve as a sequential input to the LLM decoder.

During autoregressive decoding, the decoder generates output text tokens $y = (y_1, y_2, \ldots)$ as continuation from prompt x, following the conditional probability distribution

$$\begin{aligned} h_j &= f_\theta(x^v, x^t, y_{<j}) \\ p_\theta(y_j \mid x^v, x^t, y_{<j}) &= \mathrm{softmax}(W_\mathcal{V} h_j) \end{aligned} \tag{1}$$

where $y_{<j} = (y_1, \ldots, y_{j-1})$ is the sequence of previously generated tokens, f_θ represents the LLM forward pass to produce hidden states $h_j \in \mathbb{R}^d$ on top of the Transformer layers, $W_\mathcal{V} \in \mathbb{R}^{|\mathcal{V}| \times d}$ is the output projection matrix onto the text vocabulary $\mathcal{V}$, and $y_j \in \mathcal{V}$ the output token at j-th step.

2.2. Uncertainty Quantification

Our approach quantifies the information uncertainty of visual tokens used for decoding by adapting the concept of epistemic uncertainty for measurement, as detailed in §4, and drawing inspiration from classical uncertainty decomposition [19, 40, 41]. To provide the necessary background, we first introduce the concept of uncertainty decomposition.

Uncertainty decomposition separates the total uncertainty of a model's prediction into two components: *aleatoric* uncertainty, which is inherent to the data, and *epistemic* uncertainty, which relates to the model's lack of knowledge. The Bayesian framework offers a principled way to quantify uncertainty about some candidate model with weights w, through the posterior estimation over the hypothesis space for a given dataset $\mathcal{D}$. The Bayesian model average (BMA) predictive distribution is defined as[2]

$$p(y \mid x, \mathcal{D}) = \int_w p(y \mid x, w) p(w \mid \mathcal{D})\, dw. \tag{2}$$

The total information uncertainty is measured by the entropy of BMA: $\mathbb{H}[p(y \mid x, \mathcal{D})]$, which equals to the posterior expectation of the cross-entropy (CE) between the predictive distribution of the candidate model and the BMA distribution:

$$\begin{aligned} \underbrace{\mathbb{H}[p(y \mid x, \mathcal{D})]}_{\text{Total Uncertainty}} &= \mathbb{E}_{p(w|\mathcal{D})}\left[\mathrm{CE}[p(y \mid x, w), p(y \mid x, \mathcal{D})]\right] \\ &= \underbrace{\mathbb{E}_{p(w|\mathcal{D})}\left[\mathbb{H}(p(y \mid x, w))\right]}_{\text{Aleatoric Uncertainty}} \\ &\quad + \underbrace{\mathbb{E}_{p(w|\mathcal{D})}\left[D_{\mathrm{KL}}(p(y \mid x, w) \parallel p(y \mid x, \mathcal{D}))\right]}_{\text{Epistemic Uncertainty}} \end{aligned}$$

The epistemic uncertainty term takes the form of KL divergence between individual predictive distributions of candidate models and the BMA, and it finds success in many applications [3, 13, 36]. In our approach, we adopt a similar formulation for uncertainty quantification, calculating the KL divergence between candidate prediction distributions on individual visual tokens and an aggregated average distribution.

3. Motivation Study: Visual Token Insights

As discussed in §1, identifying the visual tokens that carry significant information and quantifying their uncertainty is

[1] We assume a general Transformer architecture for vision encoder as well. Our approach could also apply to other types of vision encoders.

[2] $p(y \mid x, w, \mathcal{D}) = p(y \mid x, w)$ because of conditional independence.

critical for improving the reliability of LVLMs. To address this, we propose a supervision-free, scalable approach that maps visual tokens to the text token space, effectively translating visual content into an interpretable text-based representation. This mapping acts as a heuristic for understanding visual tokens, leveraging the LVLM's inherent ability to compress and align sequential context across visual and text modalities.

Text-space projection of visual tokens. While LVLMs are trained to generate text *only after* processing all visual tokens x^v and text instruction tokens x^t, the hidden representations h on top of the text decoder layers inherently capture textual semantics. This is due to their proximity to the text vocabulary projection, even at visual token positions where the model is not explicitly trained to generate text.

Building on this intuition, we adopt a heuristic approach to interpret visual tokens by projecting them onto the text vocabulary at the top Transformer layers. In particular, for each visual token x_i^v at position i,[3] we obtain its textual projected distribution over the vocabulary $\mathcal{V}$ from the last layer of the LLM decoder in the LVLM as:

$$\begin{aligned} h_i^v &= f_\theta(x_{\leq i}^v) \\ p_i^{\text{proj}} &= p_\theta(\cdot \mid x_{\leq i}^v) = \text{softmax}(W_\mathcal{V} h_i^v) \end{aligned} \tag{3}$$

where h_i^v is the LLM decoder top-layer hidden representation aligned at the i-th visual token positions, $x_{\leq i}^v$ denotes the visual tokens up until index i.[4]

Here, p_i^{proj}, representing the projection of the visual input onto the text space, encapsulates the model's interpretation of the i-th visual token. This projection offers a text-based summarization, akin to an unordered caption or a "bag-of-words" representation of the visual content. As we will demonstrate in §5, this heuristic method serves as an effective proxy for uncertainty estimation.

A motivating example. To illustrate the effectiveness of this projection method and motivate our approach, consider the example shown in Fig. 1. The image is processed into patches, and for five selected patches, we compute their corresponding distribution over the text space. Then we obtain the top-5 predicted text tokens for each.

Some patches yield specific and informative text tokens, typically associated with significant visual content, such as *"Berlin"*, *"computer"*, or *"map"*. These tokens are relatively closer to the long tail in vocabulary, indicating that the corresponding visual tokens capture unique and informative visual context. In contrast, patches resulting in common words carry less specific information, because high-frequency words (e.g. *"a"*, *"the"*, or *"on"*) contribute less

[3]Note that i indexes are only used over visual tokens x^v, not text tokens x^t or generations y.

[4]For the models we use, the text tokens x^t are all placed behind x^v in the concatenated sequence x, but our approach also applies to other cases.

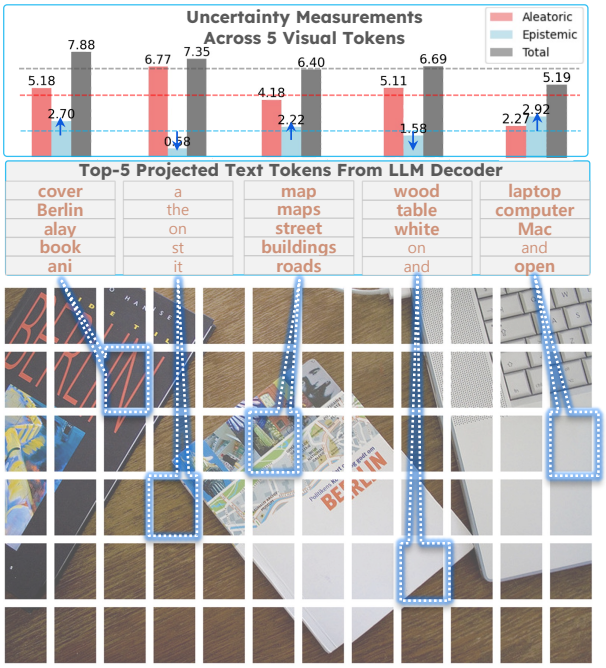

Figure 1. An illustrative example where visual tokens are projected into text space. We show 5 image patches and their projected top-5 text tokens. High epistemic uncertainty correlates well with high-information in visual tokens, whereas aleatoric and total uncertainty do not.

to the uniqueness of the visual content. This observation suggests that the projected text tokens can serve as a proxy for the information content of the visual tokens.

To further elaborate on these insights, we demonstrate the uncertainty measures based on p_i^{proj} associated with each visual token here quantified by our method (detailed computations in Sec. 4). Specifically, we decompose the total uncertainty into *aleatoric* (data-related) and *epistemic* (model-related) components. As shown in Fig. 1, the epistemic uncertainty accurately reflects the information content of the visual tokens: *visual tokens with high epistemic uncertainty correspond to patches with significant information* (e.g., *"Berlin"*), while those with low epistemic uncertainty correspond to less informative patches (e.g., *"the"*). In contrast, the aleatoric and the total uncertainty do not correlate well. This finding motivates our focus on epistemic uncertainty as a reliable indicator of the significance of visual information.

4. Method

Based on the above insights, we propose DROPOUT DECODING, leveraging uncertainty measurements to select visual information, by dropout tokens, to guide decoding. As illustrated in Fig. 2 and Algorithm 1, our approach comprises two main stages: uncertainty measurement before de-

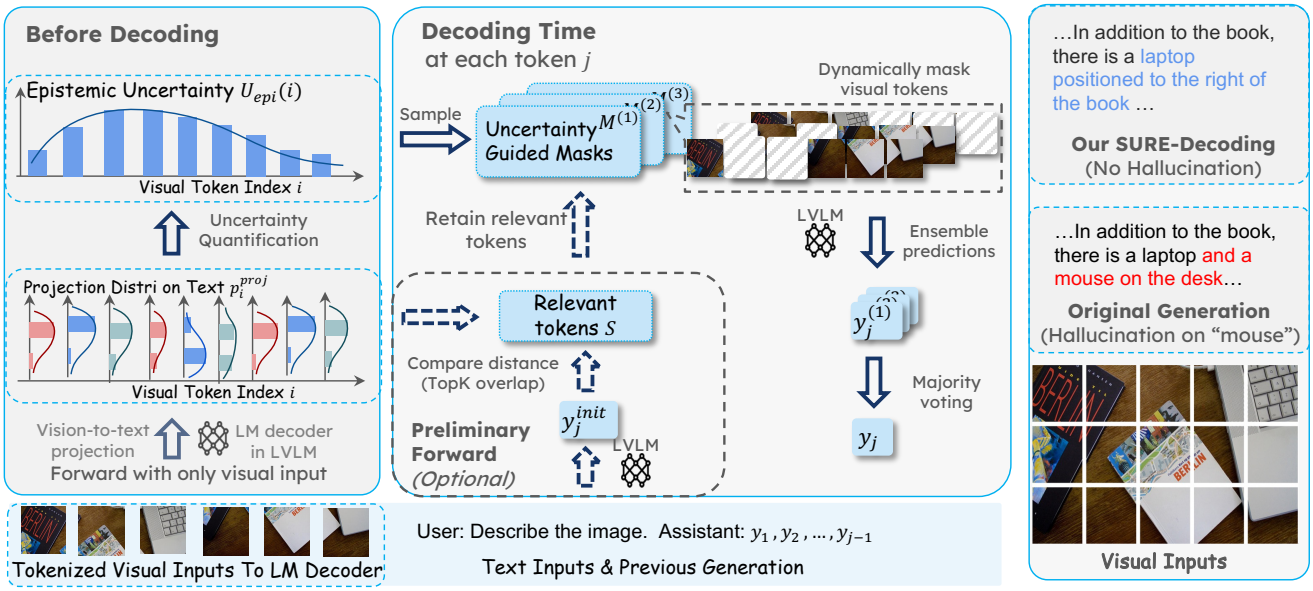

Figure 2. An overview of our DROPOUT DECODING. The method includes uncertainty measurement of visual tokens ("*Before Decoding*" in the image) and uncertainty-guided decoding algorithm ("*Decoding Time*" in the image). The pseudocode is in Algorithm 1.

coding and uncertainty-guided decoding during generation. Specifically, we introduce a supervision-free method to quantify the uncertainty associated with perceiving the visual inputs in §4.1. During decoding, we leverage these uncertainty measurements to dropout visual tokens and guide the token generation process (§4.2), resulting in more reliable and accurate outputs.

4.1. Before Decoding: Uncertainty Quantification

Bayesian visual token average distribution. Using the projected logits defined in Eq. (3), we define the average projection distribution over all visual tokens as:

$$p^{\text{proj}} = \mathbb{E}_i[p_i^{\text{proj}}] \tag{4}$$

where p_i^{proj} epresents the text-space projection of the i-th visual token. Note that the subscript i indicates different distributions rather than elements within a single distribution. The averaged distribution p^{proj} represents the overall projection of the entire visual input (e.g. an image) into the text space. This idea is grounded in classical uncertainty decomposition where a Bayesian average distribution is needed to quantify epistemic uncertainty [19, 40]. This provides us with a "baseline" representation of the visual input, against which we can quantify the surprisal of a specific visual token. By comparing each individual visual token's projection p_i^{proj} to this "baseline" p^{proj}, we can measure how much information that token adds beyond the average.

Uncertainty metrics for visual tokens. We aim to quantify the uncertainty associated with each visual token at inference time. To distinguish from those uncertainty terms in classical settings as introduced in §2.2, we refer to ours as *perception uncertainty*. We start by computing the *perception **total** uncertainty* of the visual input, quantified by the entropy of the Bayesian visual token average distribution:

$$U_{\text{total}} = \mathbb{H}\left[p^{\text{proj}}\right] = -\sum_{y\in\mathcal{V}} p^{\text{proj}}(y)\log p^{\text{proj}}(y)$$

To attribute this total uncertainty to individual visual tokens, we decompose it as follows:

$$\begin{aligned}
U_{\text{total}} &= -\sum_{y\in\mathcal{V}} p^{\text{proj}}(y)\log p^{\text{proj}}(y) \\
&= -\sum_{y\in\mathcal{V}} \left(\mathbb{E}_i\left[p_i^{\text{proj}}(y)\right]\right)\log p^{\text{proj}}(y) \\
&= \mathbb{E}_i\left[-\sum_{y\in\mathcal{V}} p_i^{\text{proj}}(y)\log p^{\text{proj}}(y)\right] \\
&= \mathbb{E}_i\left[\text{CE}\left(p_i^{\text{proj}}, p^{\text{proj}}\right)\right]
\end{aligned}$$

Further decomposing the cross-entropy, the perception total uncertainty can be expressed as:

$$\begin{aligned}
U_{\text{total}} &= \mathbb{E}_i\left[\mathbb{H}\left[p_i^{\text{proj}}\right] + D_{\text{KL}}\left(p_i^{\text{proj}} \parallel p^{\text{proj}}\right)\right] \\
&= \mathbb{E}_i\left[U_{\text{ale}}(i) + U_{\text{epi}}(i)\right]
\end{aligned}$$

where we have the *perception **aleatoric** uncertainty* of the i-th visual token $U_{\text{ale}}(i) = \mathbb{H}\left[p_i^{\text{proj}}\right]$, which captures the inherent noise or ambiguity at the position of i-th token.

Besides, the *perception **epistemic** uncertainty* is:

$$U_{\text{epi}}(i) = D_{\text{KL}}\left(p_i^{\text{proj}} \parallel p^{\text{proj}}\right) \tag{5}$$

which quantifies the divergence between the token's projection and the overall projection, indicating how much the model's belief about this token differs from its belief about the entire visual input. A higher $U_{\text{epi}}(i)$ suggests that the i-th visual token conveys information that is surprising or not well-represented in the overall visual content, which can be critical for identifying tokens that might introduce uncertainty in the decoding process.

4.2. Uncertainty-Guided Decoding

During the decoding process, we leverage the computed uncertainty metrics to guide the generation of each token. Our method involves two main steps for each generated token: (1) identifying relevant visual tokens (optional) and (2) performing *token dropout* with uncertainty-guided masking. The first step is optional, designed to enhance decoding by retaining the relevant visual tokens.

Identifying relevant visual tokens (optional). When generating each token, e.g. the j-th output token, we first perform a preliminary forward pass to generate an initial prediction y_j^{init}:

$$y_j^{\text{init}} \sim p_\theta(\cdot \mid x^v, x^t, y_{<j}) \tag{6}$$

We then identify the set of visual tokens that are relevant to this initial prediction. Specifically, we consider a visual token x_i^v as relevant if the initial prediction y_j^{init} appears among the top-k tokens in its projected distribution p_i^{proj}. Formally, the set of relevant visual tokens for j-th generation is:

$$\mathcal{S}_j = \left\{ x_i^v \;\middle|\; y_j^{\text{init}} \in \text{TopK}(p_i^{\text{proj}}) \right\} \tag{7}$$

where the TopK denotes the set of top-k entries of a distribution.

We retain these tokens in the context while considering dropping out the others in the following section, aiming to remove unrelated information and focus the model's attention on relevant visual content.

To illustrate the intuition behind this step, consider an image depicting a cat. If the model correctly predicts the token "*cat*" during the preliminary forward pass, we retain the visual tokens associated with "*cat*" and drop out the remaining information. Conversely, if the model incorrectly predicts "*dog*" or outputs unrelated words instead of an object's name—results that do not appear among the top predictions of any p_i^{proj} —we do not retain any tokens at this stage. Instead, we apply a dropout and ensemble-based method in the following section to improve reliability.

It is worth noting that this step is optional. When efficiency is a priority, omitting this preliminary forwarding can save computational cost, as we only need to perform one forward pass instead of two. While omitting this step may result in decreased performance on certain benchmarks (e.g., THRONE), it can still yield comparable performance on others (e.g., CHAIR), for details see §6.

Token dropout with uncertainty-guided masking. Using the epistemic uncertainty measurements $U_{\text{epi}}(i)$ from Eq. (5), we drop out visual tokens by sampling multiple masks. Tokens with higher epistemic uncertainty are more likely to be dropped out, encouraging diversity in the ensemble of predictions created by token dropout, which is known to be helpful in ensemble-based methods [12, 14, 24, 39].

We define K dropout probabilities across all each visual token x_i^v, $\forall i$ with the below linear transformation:

$$P_{\text{dropout}}^{(k)}(x_i^v) = \gamma^{(k)} \left(\frac{U_{\text{epi}}(i) - U_{\text{epi}}^{\min}}{U_{\text{epi}}^{\max} - U_{\text{epi}}^{\min}} \right) + \delta^{(k)} \tag{8}$$

where $\gamma^{(k)}$ and $\delta^{(k)}$ are hyperparameters controlling the probability range of dropout (for implementation see §5), and $U_{\text{epi}}^{\min}$, $U_{\text{epi}}^{\max}$ are the minimum and maximum epistemic uncertainty values across all visual tokens.

Then we generate K different dropout masks $\{M^{(k)}\}_{k=1}^{K}$ by sampling according to $P_{\text{dropout}}^{(k)}(x_i^v)$. Each dropout mask $M^{(k)}$ is a binary vector where a value of 1 indicates that the corresponding visual token is retained, and 0 indicates it is dropped out. Practically, the process is:

- Sample Dropout Masks: independently sample $M_i^{(k)} \sim P_{\text{dropout}}^{(k)}(x_i^v)\,, \forall k, i$. Higher $P_{\text{dropout}}(x_i^v)$ indicates more likely x_i^v being dropped out.
- Ensure Relevance: all relevant visual tokens are retained in each dropout mask, i.e. $M_i^{(k)} = 1$ if $x_i^v \in S$.

We then perform a parallel forward pass to obtain all the conditional probability distributions for each k with $M_i^{(k)}$:

$$y_j^{(k)} \overset{\text{Decoding}}{\sim} p_\theta^{(k)}(\cdot \mid x^v_{/M^{(k)}}, x^t, y_{<j}) \tag{9}$$

where $x^v_{/M^{(k)}}$ denotes the visual tokens after applying dropout mask $M^{(k)}$. The $\overset{\text{Decoding}}{\sim}$ indicates that the process is invariant to whatever decoding algorithm is used (we use greedy search in the implementation while others are also applicable). We aggregate the predictions from the multiple masked inputs using majority voting, selecting the token that appears most frequently among the ensemble of k *candidate predictions*. If there is a tie in the majority voting, we select the prediction from the forward pass with the least number of tokens being dropped out, considering it the most reliable due to the preservation of more information.

By creating an ensemble of predictions using different subsets of the visual input—achieved through token dropout—we can capture diverse perceptions of the same

Algorithm 1 Pseudocode for DROPOUT DECODING.

Input: visual tokens x^v, Text tokens x^t, Number of dropout masks K, Generation length L
Output: Generated sequence y

Before Decoding:
Obtain per-token projection distributions p_i^{proj}. ▷ Eq (3)
Compute average distribution p^{proj}. ▷ Eq. (4)
Compute epistemic uncertainty $U_{\text{epi}}(i)$. ▷ Eq. (5)
for $j = 1$ to L **do**
  Identifying relevant visual tokens (optional):
  Generate preliminary token y_j^{init}. ▷ Eq. (6)
  Get relevant tokens $\mathcal{S}_j$ with y_j^{init} and p_i^{proj}. ▷ Eq. (7)

  Token dropout with uncertainty-guided masking:
  Get K dropout prob $P^{(k)}$ with $U_{\text{epi}}(i)$. ▷ Eq. (8)
  Generate K dropout masks $M^{(k)}$ based on $P^{(k)}$ while retain relevant tokens $\mathcal{S}_j$.
  Forward candidates $y_j^{(k)}$ with masks $M^{(k)}$. ▷ Eq. (9)
  Majority voting on $y_j^{(k)}$ and get y_j.
end for
Return Generated sequence y

visual content. Encouraging diversity within the ensemble allows us to mitigate the impact of any single misinterpretation, leading to a more reliable generation.

5. Experiments

We evaluate the proposed DROPOUT DECODING from two aspects: object hallucination reduction and overall generation quality. For object hallucination, we use the CHAIR [38] and THRONE [22] metrics to assess the performance of different decoding methods on the MSCOCO dataset. Additionally, we employ MMBench [34] to evaluate the overall generation quality and general ability of these methods.

5.1. Experimental Setup

Baselines. In addition to the original LVLM outputs, we compare our method with two state-of-the-art decoding methods: VCD [26], which contrasts original and distorted visuals to reduce hallucinations and OPERA [18], which applies penalties and token adjustments for better grounding.

Base LVLMs. We evaluate all methods on LLaVA-1.5 [30] and InstructBLIP [9]. LLaVA-1.5 uses linear projection layers to align image and text features, generating 576 image tokens for detailed visual representation. InstructBLIP, in contrast, employs a Q-former with only 32 image tokens to bridge the modalities. This diversity highlights the flexibility of our approach, validating its efficacy across both high- and low-token-count models and confirming its robustness and adaptability.

Method	LLaVA-1.5		InstructBLIP	
	$\text{CHAIR}_S \downarrow$	$\text{CHAIR}_I \downarrow$	$\text{CHAIR}_S \downarrow$	$\text{CHAIR}_I \downarrow$
Greedy	$42.20_{\pm 2.86}$	$12.83_{\pm 0.36}$	$27.87_{\pm 1.32}$	$7.90_{\pm 0.63}$
VCD	$49.20_{\pm 0.88}$	$14.87_{\pm 0.47}$	$39.33_{\pm 2.70}$	$19.10_{\pm 0.30}$
OPERA	$41.47_{\pm 0.92}$	$12.37_{\pm 0.72}$	$28.07_{\pm 1.75}$	$8.23_{\pm 0.53}$
DROPOUT DECODING	$\mathbf{39.80}_{\pm 2.3}$	$\mathbf{11.73}_{\pm 0.25}$	$\mathbf{24.53}_{\pm 1.26}$	$\mathbf{6.63}_{\pm 0.65}$

Table 1. Comparison of methods on CHAIR_S and CHAIR_I metrics based on LLaVA-1.5 and InstructBLIP.

5.2. CHAIR

The Caption Hallucination Assessment with Image Relevance (CHAIR) [38] is a benchmark designed to evaluate object hallucination in image captioning situations. CHAIR provides two primary metrics to measure hallucination at different granularities: sentence-level and object-level. The sentence-level metric, CHAIR_S, calculates the proportion of captions that contain any hallucinated objects, giving an overall measure of hallucination frequency in captions. And the object-level metric, CHAIR_I, calculates the proportion of hallucinated objects out of all mentioned objects across captions, reflecting the prevalence of hallucination among the objects described.

Results. As shown in Table 1, DROPOUT DECODING consistently outperforms baseline approaches across various models, demonstrating its reliability and effectiveness in image captioning. Especially, on InstructBLIP, CHAIR_I is improved by approximately 16% over the second-best method, and CHAIR_S sees a gain of around 12%. These substantial improvements underscore the effectiveness of our approach, which aligns well with intuitive expectations that token dropout will reduce generated objects. Furthermore, DROPOUT DECODING reduces the generation of hallucinated objects without compromising the inclusion of relevant objects. This reduction in hallucinated content, as opposed to accurate content, is further validated by the recall metric (R_{all}) in THRONE.

5.3. THRONE

THRONE [22] assesses hallucinations in LVLM-generated responses, covering both "Type I" (mentions of non-existent objects, like CHAIR) and "Type II" (accuracy of object existence, like POPE [28]). It uses P_{all} (Precision), R_{all} (Recall), F_{all}^1, and $F_{\text{all}}^{0.5}$. Additionally, it employs F_β, which combines P_{all} and R_{all}, with the parameter β controlling the weight of R_{all} relative to P_{all}: $F_{\text{all}}^\beta = (1+\beta^2) \cdot \frac{P_{\text{all}} \times R_{\text{all}}}{(\beta^2 \times P_{\text{all}}) + R_{\text{all}}}$.

Results. The test results in Table 2 illustrate that DROPOUT DECODING surpasses nearly all baseline methods across various metrics, highlighting its effectiveness in reduc-

Method	LLaVA-1.5				InstructBLIP			
	$F^1_{\text{all}}\uparrow$	$F^{0.5}_{\text{all}}\uparrow$	$P_{\text{all}}\uparrow$	$R_{\text{all}}\uparrow$	$F^1_{\text{all}}\uparrow$	$F^{0.5}_{\text{all}}\uparrow$	$P_{\text{all}}\uparrow$	$R_{\text{all}}\uparrow$
Greedy	$0.795_{\pm 0.006}$	$0.784_{\pm 0.009}$	$0.772_{\pm 0.015}$	$0.847_{\pm 0.010}$	$0.809_{\pm 0.001}$	$0.826_{\pm 0.003}$	$0.832_{\pm 0.006}$	$0.803_{\pm 0.007}$
OPERA	$0.802_{\pm 0.003}$	$0.791_{\pm 0.004}$	$0.782_{\pm 0.009}$	$0.854_{\pm 0.011}$	$0.805_{\pm 0.004}$	$0.824_{\pm 0.003}$	$0.830_{\pm 0.004}$	$0.798_{\pm 0.008}$
VCD	$0.786_{\pm 0.012}$	$0.771_{\pm 0.017}$	$0.759_{\pm 0.020}$	$\mathbf{0.854}_{\pm 0.015}$	$0.737_{\pm 0.008}$	$0.746_{\pm 0.012}$	$0.751_{\pm 0.020}$	$0.757_{\pm 0.007}$
DROPOUT DECODING	$\mathbf{0.804}_{\pm 0.002}$	$\mathbf{0.796}_{\pm 0.006}$	$\mathbf{0.790}_{\pm 0.009}$	$0.851_{\pm 0.005}$	$\mathbf{0.814}_{\pm 0.008}$	$\mathbf{0.833}_{\pm 0.004}$	$\mathbf{0.838}_{\pm 0.002}$	$\mathbf{0.808}_{\pm 0.016}$

Table 2. Comparison of methods on F^1_{all}, $F^{0.5}_{\text{all}}$, P_{all}, and R_{all} metrics in THRONE for LLaVA-1.5 and InstructBLIP.

Method	Original	VCD	OPERA	***Dropout Decoding***
LLaVA-1.5	71.86	72.35	73.86	**74.01**

Table 3. Results on MMBench for different methods.

ing both Type I and Type II hallucinations. Specifically, DROPOUT DECODING demonstrates notable strengths in InstructBLIP, excelling in the P_{all} metric and achieving the highest performance in R_{all}. For LLaVA-1.5, P_{all} metric achieves larger improvement while the R_{all} score also exceeds that of the Greedy method, confirming that retaining overlap tokens effectively preserves relevant objects. The significant increase in $F^{0.5}_{\text{all}}$ further validates its comprehensive capability.

5.4. MMBench

MMBench [34] is a comprehensive benchmark designed to evaluate the multimodal capabilities of LVLMs across various tasks and data types, including image captioning, question answering, and object recognition. It provides a holistic view of a model's strengths and weaknesses in multimodal understanding. For our experiments, we used the "MMBench_en_dev" subset to assess performance. Since the prompt length limits in MMBench exceed InstructBLIP's token allowance, we report results only on LLaVA-1.5.

Results. As shown in Table 3, DROPOUT DECODING outperforms all the other baselines on LLaVA-1.5, which demonstrates not only its effectiveness in hallucination mitigation but also its robustness and adaptability across a broader range of multimodal tasks.

6. Analysis and Ablation Studies

6.1. Number of Candidates Predictions

As discussed in §4.2, we aggregate k candidate predictions from inputs after token dropout, with dropout masks $M^{(k)}$ In this section, we explore how varying k (from 1 to 4) impacts generation results. We set $\delta^{(k)} = 0.1$ and control $\gamma^{(k)}$ based on a predefined order: $\gamma^{(1)} = 0.3$, $\gamma^{(2)} = 0.5$, and $\gamma^{(3)} = 0.7$. Notably, setting $\gamma^{(k)}$ to 0.9 leads to excessive dropout of visual tokens and degraded performance in InstructBLIP, so we retain $\gamma^{(4)} = 0.1$. Additionally, since our majority vote algorithm selects the prediction with fewer tokens being dropped out in the event of a tie, for scenarios with only two candidates, this results in identical outputs. Therefore, we exclude Candidate 1 in the second round, leaving only Candidate 2.

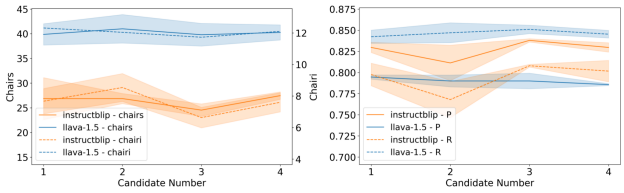

Figure 3. *Left*: Comparison of CHAIR_S and CHAIR_I scores with standard deviations across different candidate numbers. *Right*: Comparison of P_{all} and R_{all} scores with standard deviations across different candidate numbers.

As shown in Fig. 3 *Left*, both CHAIR_S and CHAIR_I scores peak at $k = 3$. When $k = 4$, adding a less-masked candidate reduces the method's ability to mitigate hallucination. Conversely, when fewer candidates (like candidate 1 and candidate 2) are used, the lower dropout probability minimizes randomness. Similarly, Fig. 3 *Right* shows that THRONE's R_{all} and P_{all} metrics also perform best at $k = 3$. Overall, we find a balance between increasing certainty with more voting and introducing controlled uncertainty through candidate dropout probability, where DROPOUT DECODING achieves trustworthy results by harmonizing these factors.

6.2. Preliminary Forward Pass

According to §4.2, DROPOUT DECODING leverages a preliminary forward pass to retain relevant objects during generation, effectively reducing hallucinated objects while maintaining high output quality. In contrast, skipping the preliminary forward pass can cause the token dropout to inadvertently mask relevant visual tokens, potentially leading to decreased performance. While including the preliminary forward pass can enhance accuracy, it requires at least double the computation time per generation. To balance performance with efficiency, we evaluate the model both with and

Model	Method	$\text{CHAIR}_S \downarrow$	$\text{CHAIR}_I \downarrow$	$R_{\text{all}} \uparrow$	$F^1_{\text{all}} \uparrow$
LLaVA-1.5	Greedy	$42.20_{\pm 2.860}$	$12.83_{\pm 0.360}$	$0.85_{\pm 0.010}$	$0.79_{\pm 0.006}$
	w/ preliminary	$39.80_{\pm 2.300}$	$\mathbf{11.73}_{\pm 0.250}$	$\mathbf{0.85}_{\pm 0.005}$	$\mathbf{0.80}_{\pm 0.002}$
	w/o preliminary	$\mathbf{39.70}_{\pm 2.150}$	$12.20_{\pm 0.700}$	$0.84_{\pm 0.005}$	$0.799_{\pm 0.002}$
InstructBLIP	Greedy	$27.87_{\pm 1.320}$	$7.90_{\pm 0.630}$	$0.80_{\pm 0.007}$	$0.81_{\pm 0.001}$
	w/ preliminary	$\mathbf{24.53}_{\pm 1.260}$	$\mathbf{6.63}_{\pm 0.650}$	$\mathbf{0.81}_{\pm 0.016}$	$\mathbf{0.82}_{\pm 0.008}$
	w/o preliminary	$26.20_{\pm 4.010}$	$7.10_{\pm 0.850}$	$0.80_{\pm 0.010}$	$0.81_{\pm 0.008}$

Table 4. Comparison of w/ and w/o preliminary forward on CHAIR_S, CHAIR_I, R_{all}, and F^1_{all} metrics for LLaVA-1.5 and InstructBLIP.

without this preliminary forward pass to 1) validate the efficacy of the preliminary forward pass; and 2) explore a potentially more efficient approach as a substitute when computational resources are limited.

As shown in Table 4, incorporating the preliminary forward pass consistently outperforms setups without it across most metrics, with a particularly notable improvement in R_{all}. While reducing the number of generated objects can lower hallucination and boost CHAIR metrics, it may also decrease R_{all} on the THRONE benchmark. Specifically, for LLaVA-1.5, the preliminary forward pass improves performance beyond the baseline, especially in CHAIR metrics. In contrast, for InstructBLIP, CHAIR remains competitive, but the THRONE metric is lower.

We hypothesis that this is due to the differences in visual token abundance: LLaVA-1.5 uses 576 visual tokens, diluting the significance of each, whereas InstructBLIP relies on only 32, making each token highly informative. As a result, omitting the preliminary forward pass in InstructBLIP can inadvertently drop out critical information, leading to reduced performance. These findings demonstrate the effectiveness of the preliminary forward pass in preserving relevant objects, particularly for models with abundant visual tokens. Conversely, models with fewer visual tokens may benefit from omitting it to optimize efficiency.

7. Related Work

Reliable generation. Reliable generation in LLMs is often challenged by hallucinations, where the model generates irrelevant or factually incorrect information [17, 44, 59]. These hallucinations stem from issues in data, training, and inference stages [49], with attention mechanisms exacerbating the problem as sequence lengths grow [8]. To mitigate these, methods like factual-nucleus sampling have been proposed to balance output diversity and factual accuracy [25]. Besides, while Arias et al. [2] leverage quantified uncertainty to guide the decoding process for LLM, our method differs significantly. We quantify uncertainty at the level of visual input context rather than of model ensemble which is heavy.

Object hallucination in LVLMs. Object hallucination is a common issue in LVLMs, where models generate descriptions containing objects, attributes, or relationships not present in the actual image. The CHAIR metric [38] is widely used to evaluate object hallucination, measuring the hallucination rate on the MSCOCO dataset [29]. Another benchmark, POPE [28], treats object hallucination as a binary classification task. More recently, THRONE [22] takes a more holistic approach, using open-ended, object-based image descriptions for evaluation. In our work, we use CHAIR and THRONE to assess object hallucination.

Object hallucination reduction. Recent methods addressing hallucination in LVLMs include internal signal guidance, contrastive decoding, and selective information focusing, all of which are inference-time strategies. OPERA [18] uses internal signals like attention patterns to refine outputs and improve alignment with visual content. Contrastive decoding methods, like VCD [26], enhance coherence by comparing image-specific outputs. Selective information focusing approaches, such as HALC [4], prioritize key image regions, while CDG [10] uses CLIP embeddings to align generation with visual input. In contrast, DROPOUT DECODING works with any LVLM by 1) selecting visual information from visual tokens during generation, unlike HALC which selects regions initially; 2) using uncertainty to guide visual information selection, requiring no external models, unlike HALC and CDG; 3) introducing a token-level majority voting strategy.

8. Conclusion

In this paper, we introduce DROPOUT DECODING, a novel uncertainty-guided selective decoding approach aimed at enhancing the reliability of LVLMs. After quantifying and decomposing uncertainty in visual inputs, DROPOUT DECODING employs an ensemble-based method that utilizes epistemic uncertainty to selectively mask high-uncertainty tokens. Extensive experiments on benchmarks including CHAIR, THRONE, and MMBench validate the effectiveness, demonstrating consistent performance improvements over existing methods in both hallucination reduction and general multimodal capability tasks.

References

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015. 1

[2] Esteban Garces Arias, Julian Rodemann, Meimingwei Li, Christian Heumann, and Matthias Aßenmacher. Adaptive contrastive search: Uncertainty-guided decoding for open-ended text generation. *arXiv preprint arXiv:2407.18698*, 2024. 8

[3] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation, 2018. 2

[4] Zhaorun Chen, Zhuokai Zhao, Hongyin Luo, Huaxiu Yao, Bo Li, and Jiawei Zhou. Halc: Object hallucination reduction via adaptive focal-contrast decoding. In *Forty-first International Conference on Machine Learning*. 1, 8

[5] Zhaorun Chen, Yichao Du, Zichen Wen, Yiyang Zhou, Chenhang Cui, Zhenzhen Weng, Haoqin Tu, Chaoqi Wang, Zhengwei Tong, Qinglan Huang, et al. Mj-bench: Is your multimodal reward model really a good judge for text-to-image generation? *arXiv preprint arXiv:2407.04842*, 2024. 1

[6] Zhaorun Chen, Zhuokai Zhao, Zhihong Zhu, Ruiqi Zhang, Xiang Li, Bhiksha Raj, and Huaxiu Yao. Autoprm: Automating procedural supervision for multi-step reasoning via controllable question decomposition. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1346–1362, 2024. 1

[7] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatial-rgpt: Grounded spatial reasoning in vision language model. *arXiv preprint arXiv:2406.01584*, 2024. 1

[8] David Chiang and Peter Cholak. Overcoming a theoretical limitation of self-attention. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7654–7664, Dublin, Ireland, 2022. Association for Computational Linguistics. 8

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

[10] Ailin Deng, Zhirui Chen, and Bryan Hooi. Seeing is believing: Mitigating hallucination in large vision-language models via clip-guided decoding, 2024. 8

[11] Yifan Du, Zikang Liu, Junyi Li, and Wayne Xin Zhao. A survey of vision-language pre-trained models. *arXiv preprint arXiv:2202.10936*, 2022. 1

[12] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019. 5

[13] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1050–1059, New York, New York, USA, 2016. PMLR. 1, 2

[14] Mudasir A Ganaie, Minghui Hu, Ashwani Kumar Malik, Muhammad Tanveer, and Ponnuthurai N Suganthan. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151, 2022. 5

[15] Akash Ghosh, Arkadeep Acharya, Sriparna Saha, Vinija Jain, and Aman Chadha. Exploring the frontier of vision-language models: A survey of current methodologies and future directions. *arXiv preprint arXiv:2404.07214*, 2024. 1

[16] Anisha Gunjal, Jihan Yin, and Erhan Bas. Detecting and preventing hallucinations in large vision language models. *arXiv preprint arXiv:2308.06394*, 2023. 1

[17] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions, 2023. 8

[18] Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. Opera: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation, 2024. 6, 8

[19] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506, 2021. 1, 2, 4

[20] Lai Jiang, Hongqiu Wu, Hai Zhao, and Min Zhang. Chinese spelling corrector is just a language learner. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 6933–6943, 2024. 1

[21] Ehsan Kamalloo, Nouha Dziri, Charles LA Clarke, and Davood Rafiei. Evaluating open-domain question answering in the era of large language models. *arXiv preprint arXiv:2305.06984*, 2023. 1

[22] Prannay Kaul, Zhizhong Li, Hao Yang, Yonatan Dukler, Ashwin Swaminathan, C. J. Taylor, and Stefano Soatto. Throne: An object-based hallucination benchmark for the free-form generations of large vision-language models, 2024. 6, 8

[23] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017. 1

[24] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017. 5

[25] Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale Fung, Mohammad Shoeybi, and Bryan Catanzaro. Factuality enhanced language models for open-ended text generation, 2023. 8

[26] Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. Mitigating object hallucinations in large vision-language models through visual contrastive decoding, 2023. 6, 8

[27] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022. 2

[28] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models, 2023. 6, 8

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. 8

[30] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 2, 6

[31] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2024. 1, 2

[32] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 1

[33] Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou, Rongjun Li, and Wei Peng. A survey on hallucination in large vision-language models. *arXiv preprint arXiv:2402.00253*, 2024. 1

[34] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player?, 2024. 6, 7

[35] Zixian Ma, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna. Crepe: Can vision-language foundation models reason compositionally? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10910–10921, 2023. 1

[36] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn, 2016. 2

[37] Denisa Roberts and Lucas Roberts. Smart vision-language reasoners. *arXiv preprint arXiv:2407.04212*, 2024. 1

[38] Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning, 2019. 6, 8

[39] Lior Rokach. Ensemble-based classifiers. *Artificial intelligence review*, 33:1–39, 2010. 5

[40] Kajetan Schweighofer, Lukas Aichberger, Mykyta Ielanskyi, and Sepp Hochreiter. Introducing an improved information-theoretic measure of predictive uncertainty. *arXiv preprint arXiv:2311.08309*, 2023. 1, 2, 4

[41] Kajetan Schweighofer, Lukas Aichberger, Mykyta Ielanskyi, Günter Klambauer, and Sepp Hochreiter. Quantification of uncertainty with adversarial models. *Advances in Neural Information Processing Systems*, 36:19446–19484, 2023. 2

[42] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 1

[43] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: visual explanations from deep networks via gradient-based localization. *International journal of computer vision*, 128:336–359, 2020. 1

[44] Chaoqi Wang, Zhuokai Zhao, Chen Zhu, Karthik Abinav Sankararaman, Michal Valko, Xuefei Cao, Zhaorun Chen, Madian Khabsa, Yuxin Chen, Hao Ma, et al. Preference optimization with multi-sample comparisons. *arXiv preprint arXiv:2410.12138*, 2024. 8

[45] Weiyun Wang, Zhe Chen, Wenhai Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Jinguo Zhu, Xizhou Zhu, Lewei Lu, Yu Qiao, and Jifeng Dai. Enhancing the reasoning ability of multimodal large language models via mixed preference optimization. *arXiv preprint arXiv:2411.10442*, 2024. 1

[46] Andrew G Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *Advances in neural information processing systems*, 33:4697–4708, 2020. 1

[47] Kelvin Xu. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2015. 1

[48] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention, 2016. 1

[49] Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. Hallucination is inevitable: An innate limitation of large language models, 2024. 8

[50] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*, 2023. 1

[51] Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and ground anything anywhere at any granularity. *arXiv preprint arXiv:2310.07704*, 2023. 1

[52] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it?, 2023. 1

[53] Bohan Zhai, Shijia Yang, Chenfeng Xu, Sheng Shen, Kurt Keutzer, and Manling Li. Halle-switch: Controlling object hallucination in large vision language models. *arXiv e-prints*, pages arXiv–2310, 2023. 1

[54] Haotian Zhang, Haoxuan You, Philipp Dufter, Bowen Zhang, Chen Chen, Hong-You Chen, Tsu-Jui Fu, William Yang Wang, Shih-Fu Chang, Zhe Gan, et al. Ferret-v2: An improved baseline for referring and grounding with large language models. *arXiv preprint arXiv:2404.07973*, 2024. 1

[55] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1

[56] Jianyi Zhang, Da-Cheng Juan, Cyrus Rashtchian, Chun-Sung Ferng, Heinrich Jiang, and Yiran Chen. Sled: Self logits evolution decoding for improving factuality in large language models. *arXiv preprint arXiv:2411.02433*, 2024. 1

[57] Yiming Zhang, Zhuokai Zhao, Zhaorun Chen, Zhili Feng, Zenghui Ding, and Yining Sun. Rankclip: Ranking-consistent language-image pretraining. *arXiv preprint arXiv:2404.09387*, 2024. 1

[58] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023. 1

[59] Zhuokai Zhao. Enhanced data utilization for efficient and trustworthy deep learning. 2024. 8

[60] Zhuokai Zhao, Harish Palani, Tianyi Liu, Lena Evans, and Ruth Toner. Multimodal guidance network for missing-modality inference in content moderation. In *2024 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pages 1–4. IEEE, 2024. 1

A. Generation Examples and Analysis

A.1. LLaVA-1.5 Examples

Generated Caption:

Greedy Generated caption:

The image features a black and white photograph of a small boat with three people rowing it down a river. The boat is being pulled by the people, and there are two swans swimming alongside the boat. The swans are swimming in the water near the boat, creating a peaceful and serene scene. The people in the boat are enjoying the ride, possibly on a leisurely outing.

DROPOUT DECODING Generated caption:

The image features a group of people in a small boat, paddling down a river. There are at least four people visible in the boat, with some of them sitting closer to the front and others towards the back. The boat is filled with people, and they seem to be enjoying their time together on the water.

Generated Caption:

Greedy Generated caption:

The image features a kitchen countertop with a microwave oven and a large, wrapped object sitting on top of it. The object is wrapped in aluminum foil, which is covering a large rectangular box or container. The microwave oven is positioned on the left side of the countertop, and the wrapped object is placed on the right side. The countertop is also adorned with a few utensils, such as a spoon and a fork.

DROPOUT DECODING Generated caption:

The image features a kitchen counter with a metal pan or tray sitting on top of it. The pan is covered in aluminum foil, which is commonly used to keep food warm or to prevent it from sticking to the pan. The kitchen counter appears to be a part of a larger countertop, and the scene suggests that the pan is being prepared for cooking or baking.

Generated Caption:

Greedy Generated caption:

The image features a dining table set with a white plate containing a variety of food items, including meat, vegetables, and bread. The plate is placed on a wooden table, and there is a glass of wine next to it. The table is set for a meal, with a fork and knife placed on the table. The food items on the plate are arranged in a visually appealing way, making the dish look appetizing.

DROPOUT DECODING Generated caption:

The image features a dining table with a white plate containing a delicious meal. The plate is topped with a variety of food items, including a piece of meat, a roll, and a side of vegetables. There are also a couple of carrots on the plate. In addition to the plate, there is a wine glass placed on the table, suggesting that the meal is accompanied by a glass of wine. A chair is positioned near the table, ready for someone to sit and enjoy the meal.

A.2. InstructBLIP Examples

Generated Caption:

Greedy Generated caption:

The image features a table with a laptop and a plate of food, including a bowl of salad and a plate of food. The food is arranged on the table, with a fork and a knife nearby. The laptop is placed on the table, likely used for work or entertainment. The table is set up for a comfortable dining experience, with a variety of food options available.

DROPOUT DECODING Generated caption:

The image features a plate of food on the table is aesthetically pleasing and well-balanced. There is a plate of food, including a salad, and a bowl of chicken, and a fork. The plate is placed on the table, which is also adorned with a laptop and a computer. The laptop is placed on the table, and the computer is positioned to the left of the plate.

Generated Caption:

Greedy Generated caption:

The image features a red fire hydrant placed in front of a stone wall. The fire hydrant is positioned in the middle of the scene, with a window above it. There are two benches placed near the fire hydrant, one on the left and another on the right side of the image. The benches are positioned close to the fire hydrant, with one closer to the window and the other further away.

DROPOUT DECODING Generated caption:

The image features a red fire hydrant is placed in front of a stone building, with a window above it. The window is located on the second floor of the building, and the fire hydrant is positioned in front of the window. There are two metal bars on either side of the fire hydrant, which are likely used to secure the hydrant in place. The overall scene is quite unique, with the fire hydrant and the window creating a distinctive composition.

Generated Caption:

Greedy Generated caption:

The image features a dining table with a white plate topped with a slice of pie, a spoon, and a cherry on top. The pie is surrounded by various utensils, including a fork, knife, and spoon. There are also two cups on the table, one on the left and one on the right, each containing a beverage. The dining table is set with a variety of dishes, including a bowl, a cup, and a spoon. The scene is set for a delicious meal, with the pie and beverages adding to the overall ambiance.

DROPOUT DECODING Generated caption:

The image features a dining table is set with a variety of dishes, including a plate with a slice of pie, a cherry, and a spoon. There are also several cups of coffee and a bowl of soup. The table is set for a meal, and the dishes are arranged in a way that suggests a person is about to enjoy their food.

A.3. Analysis

In our examples, we use red to denote hallucinations. Specifically, DROPOUT DECODING addresses hallucinations in two key ways: eliminating non-existent objects and correcting the generation of misidentified objects. For instance, in the third example of LLaVA-1.5, the Greedy method hallucinates a non-existent object, a knife, whereas DROPOUT DECODING effectively removes this hallucination. Similarly, in the second example of InstructBLIP, the Greedy method misidentifies metal bars as benches, but DROPOUT DECODING successfully corrects this, accurately recognizing the metal bars.

B. Implementation Details

The experimental setup of DROPOUT DECODING is shown in Table 5. We set the maximum new tokens to 512 to ensure the complete generation of models, therefore achieving more reliable results from CHAIR and THRONE. In MMBench, as all questions are single-choice questions, we set the maximum new tokens to 1 for a more precise evaluation. We set other parameters in generation to greedy for more stable and repeatable results.

Parameters	**Value**
Maximum New Tokens (CHAIR)	512
Maximum New Tokens (THRONE)	512
Maximum New Tokens (MMBench)	1
Top-k	False
Top-p	1
Temperature τ	1
Number Beams	1

Table 5. Parameter settings used in our experiments.

In addition to general generation settings, DROPOUT DECODING includes hyperparameters specified in §4.2. The details of these hyperparameter settings are provided below:

top-k in identifying relevant visual tokens. Before the decoding process, we first obtain p^{proj}, which is then used in the decoding process for generating the relevant visual tokens. The higher the top-k is, the more visual tokens are expected to be kept during the decoding process. In LLaVA-1.5, we set $k = 5$, and in InstructBLIP, we set $k = 10$. The difference of k between LLaVA-1.5 and InstructBLIP derives from the informative level of each visual token, where in LLaVA-1.5, each visual token carries less information than in InstructBLIP, which only contains 32 visual tokens.

Number of mask K. K refers to the number of predictions that will join the majority vote progress. We set $K = 3$ in our experiment settings.

$\gamma^{(k)}$ and $\delta^{(k)}$ in uncertainty-guided masking We set $\delta^{(k)} = 0.1, \gamma^{(k)} = 0.2 * k + 1; k = 1, 2, ..., K; K = 3$ in our experiment settings.

Parameters	**Value**
Self-attention Weights Scale Factor θ	50
Attending Retrospection Threshold	15
Beam Size	3
Penalty Weights	1

Table 6. OPERA Hyperparameter Settings

Parameters	Value
Amplification Factor α	1
Adaptive Plausibility Threshold	0.1
Diffusion Noise Step	500

Table 7. VCD Hyperparameter Settings

Moreover, we provide the hyperparameter settings of our baselines. OPERA's hyperparameters can be referred to Table 6; VCD's hyperparameters can be referred to Table 7.

C. Further Discussion on Ablation Studies

To further validate our uncertainty guidance' effectiveness, we select random masking strategy as an additional baseline to compare with DROPOUT DECODING's uncertainty-guided masking. The experimental setup remains identical, except that tokens are masked randomly, that is, candidate k masks each vision token at $\gamma^{(k)}$ instead of using uncertainty guidance. The generated results using the random masking strategy often suffer from issues, with models producing repeated tokens until reaching the maximum token limit. For instance, the model might repeatedly generate "skiers" hundreds of times (generation: "The image shows shows a a snowy snowy slope with a skiers skiers skiers skiers ..."); this occurred in approximately 20–25 out of 500 cases, an issue nearly never encountered with our method. This behavior likely stems from random masking disrupting essential context information within LVLMs. In contrast, our uncertainty-guided masking applies a lower masking rate to tokens that the LVLMs are less "surprised" by and a higher rate to tokens that elicit greater surprise. This allows the model to generate content in a rather "expected" manner, even though many informative vision tokens are masked. By preserving base context information, our approach effectively maintains the LVLMs' consistency and coherence.