Enhancing Vision-Language Model Reliability with Uncertainty-Guided Dropout Decoding

Yixiong Fang[*]
Carnegie Mellon University
yixiongf@cs.cmu.edu

Ziran Yang[*]
Princeton University
zirany@princeton.edu

Zhaorun Chen
University of Chicago
zhaorun@uchicago.edu

Zhuokai Zhao[†]
University of Chicago
zhuokai@uchicago.edu

Jiawei Zhou[†]
Stony Brook University
jiawei.zhou.1@stonybrook.edu

Abstract

Large vision-language models (LVLMs) excel at multimodal tasks but are prone to misinterpreting visual inputs, often resulting in hallucinations and unreliable outputs. We present DROPOUT DECODING, a novel inference-time approach that quantifies the uncertainty of visual tokens and selectively masks uncertain tokens to improve decoding. Our method measures the uncertainty of each visual token by projecting it onto the text space and decomposing it into *aleatoric* and *epistemic* components. Specifically, we focus on *epistemic* uncertainty, which captures perception-related errors more effectively. Inspired by dropout regularization, we introduce uncertainty-guided *token dropout*, which applies the dropout principle to input visual tokens instead of model parameters, and during inference rather than training. By aggregating predictions from an ensemble of masked decoding contexts, we can robustly mitigate errors arising from visual token misinterpretations. Evaluations on benchmarks including CHAIR, THRONE, and MMBench demonstrate that DROPOUT DECODING significantly reduces object hallucinations (OH) and enhances both reliability and quality of LVLM outputs across diverse visual contexts. Code is released at `https://github.com/kigb/DropoutDecoding`.

1 Introduction

Recent advancements in large vision-language models (LVLMs) have demonstrated impressive capabilities [1, 2, 3] in tasks such as image captioning, visual question answering (VQA), and multimodal reasoning [4, 5, 6, 7]. However, LVLMs still face challenges in accurately perceiving and interpreting visual inputs, leading to inaccurate outputs and hallucinations [8]. These issues often stem from LVLMs misrepresenting key image elements or overlooking critical details [9, 10]. In practice, LVLMs typically process visual inputs token by token [11], which we refer to as *visual tokens*.[3] This 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 [12, 13, 14], 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 [15, 16, 17, 18, 19], or using additional supervision signals especially at lower level to guide the model [20, 21]. However, these approaches are resource-intensive

[*]Equal contribution. Work done during their research internship at Stony Brook University.
[†]Joint last author.
[3]We specifically refer to the tokens that are already in the input prompt to the text decoder. Concrete definition is in §3.1.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

and not easily extensible to new tasks. Alternative inference-time strategies rely on attention or logits-based mechanisms but typically use heuristic designs and increase inference cost [22, 23, 24, 25]. Therefore, enhancing the trustworthiness of LVLMs [26] and reducing hallucinations [24] require more principled methods that can more effectively emphasize the most informative parts of the visual input.

To address this challenge, we propose a novel approach that quantifies uncertainty in visual token contexts and removes uncertain tokens, both directly at inference time to improve the reliability of LVLM outputs. Inspired by traditional dropout [27] techniques—typically applied to model parameters but difficult to implement directly in pretrained LVLMs [28, 29]—we introduce *token dropout*, which applies the dropout principle to input context tokens instead of model parameters. Furthermore, it is applied to regularize the inference process instead of training, by introducing randomness in decoding contexts to reduce overfitting to noisy visual tokens.

Our method measures the uncertainty of each visual token by projecting it into the text token space *through the text decoder directly*, and decomposing this uncertainty into two components: *aleatoric* (data-related) and *epistemic* (model-related) [30, 31, 32]. By focusing on epistemic uncertainty, which reflects the model's lack of knowledge, we identify visual tokens with high uncertainty and selectively target them for suppression. At inference time, we adjust the visual inputs by selectively suppressing tokens with high epistemic uncertainty. Specifically, we create an *ensemble of predictions* by generating multiple subsets of visual inputs, each with different combinations of high-uncertainty tokens dropped out. These subsets are processed independently, and their corresponding outputs are aggregated using majority voting to produce the final prediction.

Our method, termed DROPOUT DECODING, enhances the reliability and accuracy of LVLM outputs without modifying the underlying model parameters or requiring additional training. Experiments on LVLM decoding benchmarks including CHAIR [33], THRONE [34], and MMBench [35] demonstrate the effectiveness of our approach. In summary, we make the following contributions. First, we introduce a novel approach that quantifies and decomposes uncertainty on tokens in the visual inputs at inference time without additional supervision, by projecting visual input tokens onto text token interpretations. Second, we propose 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 and during inference. And finally, comprehensive experiments are conducted on various benchmarks, showing significant reductions in OH and improved fidelity in pre-trained LVLMs without additional fine-tuning.

2 Related Work

Reliable Generation. Hallucinations in LLMs—where models generate irrelevant or incorrect information [36, 37, 38]—arise from data [19], training, and inference issues [39], with attention mechanisms exacerbating them [40]. To address this, factual-nucleus sampling [41] balances diversity and accuracy. While [42] guide decoding with quantified uncertainty, our approach quantifies uncertainty at the visual input level, not requiring model ensembles.

OH in LVLMs. Object hallucination (OH) is common in LVLMs, where models generate incorrect object descriptions. CHAIR [33] and POPE [43] evaluate OH, while THRONE [34] offers a more holistic approach. We use CHAIR and THRONE to assess OH in our work.

OH Reduction. Methods addressing OH in LVLMs include internal signal guidance (e.g., OPERA [44]), contrastive decoding (e.g., VCD [45]), and selective information focusing (e.g., HALC [24]). AGLA [46] mitigates hallucinations by enhancing visual grounding through global and local attention, while Memory-Space Visual Retracing [47] refines multimodal alignment via iterative visual reference retrieval. In contrast, DROPOUT DECODING 1) selects visual tokens during generation, 2) uses uncertainty for token selection without external models, and 3) employs a token-level majority voting strategy.

3 Preliminaries

3.1 Vision-Language Model Decoding

Widely adopted LVLM architectures [48, 49, 17] typically include a vision encoder, a vision-text interface module, and a Transformer-based LLM decoder. As we mostly focus on the decoder side inference, we assume the LLM decoder parameterized by θ.

The visual input, such as an image, is segmented into patches and processed by the vision encoder,[4] followed by the vision-text interface module, to produce a sequence of *visual tokens* $x^v = (x_1^v, x_2^v, \dots, x_N^v)$. 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 $x^t = (x_1^t, x_2^t, \dots, x_M^t)$. 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 $y = (y_1, y_2, \dots)$ as continuation from prompt x, following the conditional probability distribution

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

where $y_{<j} = (y_1, \dots, y_{j-1})$ is the sequence of previously generated tokens, f_θ denotes 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.

3.2 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 §5, and drawing inspiration from classical uncertainty decomposition [31, 50, 32]. 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[5]

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

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

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

The epistemic uncertainty, expressed as the KL divergence between candidate models' predictive distributions and the BMA, has proven effective in various applications [51, 52, 28, 53]. Our approach, adopts a similar formulation for uncertainty quantification, calculating the KL divergence between candidate prediction distributions on individual visual tokens and an aggregated average distribution.

4 Textual Interpretation of Visual Tokens

As discussed in §1, identifying the visual tokens that carry significant information and quantifying their uncertainty is critical for improving the reliability of LVLMs. We propose a supervision-free approach that maps visual tokens to text token space for improving LVLM reliability by identifying significant visual tokens and quantifying their uncertainty. This mapping leverages the LVLM's inherent ability to align visual and textual contexts.

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*.

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

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

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,[6] we obtain its textual projected distribution over the vocabulary $\mathcal{V}$ from the last layer of the LLM decoder in the LVLM as:

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

where h_i^v is the LLM decoder top-layer hidden representation aligned at the i-th visual token positions, $x_{\le i}^v$ denotes the visual tokens up until index i.[7] This approach is also generally referred to as logit lens [54] in mechanistic interpretability for LLMs.

Here, q_i^{proj}, which we refer to as *visual-textual distribution*, represents the projection of the visual input onto the text space. It 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" [15] representation of the visual content. As we will demonstrate in §6, this heuristic method serves as an effective proxy for uncertainty estimation.

An illustrative example with projection uncertainty. Figure 4 demonstrates our projection method by processing an image into patches and projecting five selected patches into the text space, retrieving their top-5 text tokens. Informative patches yield specific tokens like "*Berlin*," "*computer*," or "*map*," which are less frequent in the vocabulary and capture unique visual contexts. In contrast, patches producing common words (e.g., "*a*," "*the*," "on") convey less specific information. This suggests that projected text tokens effectively proxy the information content of visual tokens.

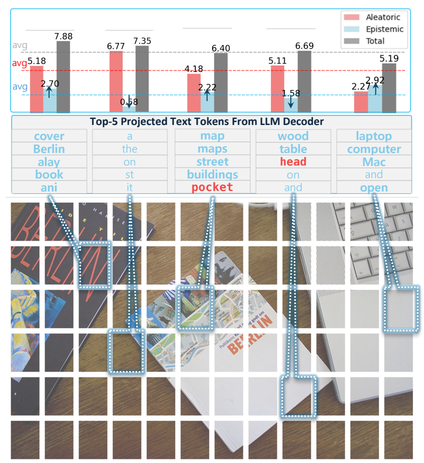


Figure 1: An illustrative example where visual tokens are projected into the text space, **bold** words indicate highly informative projections, and red words mark misalignments. Dotted lines show average uncertainties; high epistemic uncertainty correlates with informative patches.

Leveraging this, we introduce uncertainty measures from the textual projection distributions q_i^{proj} to quantify each visual token's uncertainty, as depicted in the figure. Following classical uncertainty quantification (§3.2), we decompose total uncertainty into *aleatoric* (data-related) derived directly from q_i^{proj}, and *epistemic* (model-related) by comparing q_i^{proj} to an average distribution (§5.1). As illustrated, epistemic uncertainty aligns well with the information content of visual tokens: high epistemic uncertainty corresponds to informative patches (e.g., "*Berlin*"), and vice versa (e.g., "*the*"). In contrast, aleatoric and total uncertainty do not show this correlation. This finding motivates our focus on epistemic uncertainty as a reliable indicator of the significance of visual information.

5 Method

We propose DROPOUT DECODING, which leverages visual uncertainty to selectively drop out visual tokens and guide decoding. As shown in Fig. 2 and Algorithm 1, our approach comprises two stages: uncertainty quantification (§5.1) before decoding and uncertainty-guided token generation (§5.2) for decoding.

5.1 Uncertainty Quantification Before Decoding

Average visual-textual distribution. We begin by defining the averaged distribution q^{proj}, which represents the overall projection of the entire visual input (e.g. an image) into the text space. Using the projected distribution defined in Eq. (3), we define the average projection distribution over all

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

[7]For the models we use, the visual tokens x^v are all placed before the text tokens x^t in the concatenated sequence x, so $x_{\le i}^v$ are purely visual tokens. But our approach also applies to other cases.

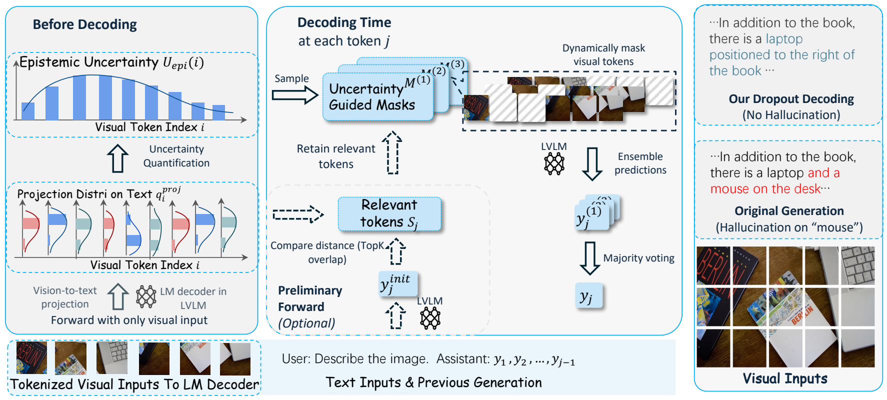


Figure 2: An overview of our DROPOUT DECODING. The method includes uncertainty measurement of visual tokens (under "*Before Decoding*") and uncertainty-guided visual context dropout decoding algorithm (under "*Decoding Time*"). The pseudocode is in Algorithm 1.

visual tokens as:

$$q^{\text{proj}} = \mathbb{E}_i[q_i^{\text{proj}}] = \frac{1}{N}\sum_i^N q_i^{\text{proj}} \tag{4}$$

where q_i^{proj} represents the text-space projection of the i-th visual token, and N is the total number of visual tokens. Note that the subscript i indicates different distributions rather than elements within a single distribution. This provides us with a "baseline" representation of the visual input, against which we can quantify the surprisal of a specific visual token. This idea is grounded in classical uncertainty decomposition where a Bayesian average distribution is needed to quantify epistemic uncertainty [31, 32].

Uncertainty measurement 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 §3.2, we refer to ours as *perception uncertainty*. We start by quantifying the *perception **total** uncertainty* of the visual input as the entropy of the average visual-textual distribution $\mathbb{H}\left[q^{\text{proj}}\right]$. Then, to attribute this total uncertainty to individual visual tokens, we decompose it (details in Appendix A) by:

$$U_{\text{total}} = \mathbb{H}\left[q^{\text{proj}}\right] = \mathbb{E}_i\left[\text{CE}\left(q_i^{\text{proj}}, q^{\text{proj}}\right)\right] \tag{5}$$

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

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

Here we have the *perception **aleatoric** uncertainty* of the i-th visual token $U_{\text{ale}}(i) = \mathbb{H}\left[q_i^{\text{proj}}\right]$, capturing the inherent noise or ambiguity of the i-th token, and the *perception **epistemic** uncertainty*—

$$U_{\text{epi}}(i) = D_{\text{KL}}\left(q_i^{\text{proj}} \parallel q^{\text{proj}}\right) \tag{6}$$

quantifying the divergence between the visual token's textual projection and the overall projection. It indicates 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.

5.2 Uncertainty-Guided Decoding

During the text decoding process, we leverage the computed uncertainty measures to guide the generation of each token. Our method involves two main steps for each generated *text* 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 more relevant visual tokens.

Identifying relevant visual tokens (optional). We selectively retain only the most relevant visual tokens from the context, which are excluded for dropout. When generating each output text token, y_j, we first perform a preliminary forward pass to generate an initial prediction token y_j^{init}:

$$y_j^{\text{init}} \sim p_\theta(\cdot \mid x^v, x^t, y_{<j}) \tag{7}$$

Next, we determine the set of visual tokens that are relevant to this initial prediction. Specifically, a visual token x_i^v is considered relevant if the initial prediction y_j^{init} appears among the top-k tokens of its visual-textual projection q_i^{proj}. Formally, the set of relevant visual tokens for the j-th generation is:

$$\mathcal{S}_j = \left\{ x_i^v \;\middle|\; y_j^{\text{init}} \in \text{TopK}(q_i^{\text{proj}}) \right\} \tag{8}$$

where $\text{TopK}(\cdot)$ denotes the function returning the top-k entries of a given distribution.

To illustrate the intuition behind this step, consider an image depicting a cat. Suppose the model correctly predicts the token "*cat*" during the preliminary forward pass. In that case we retain the visual tokens associated with "*cat*" and drop out among the remaining visual content. Conversely, if the model incorrectly predicts "*dog*" or unrelated tokens irrelevant to an object, these predictions will not align with the top text projections of any q_i^{proj} if the visual interpretation is accurate. In such cases, no visual tokens are retained due to a lack of clear relevance, and dropout is applied across the entire visual context as the best alternative.

It is worth noting that this step is optional. Omitting it can improve efficiency by reducing the computational overhead of the preliminary forward pass. As shown by the ablation studies in §7, while skipping this step may lower performance on certain benchmarks like THRONE [34], it still achieves comparable results on others such as CHAIR [33].

Visual token dropout with uncertainty guidance. Using the epistemic uncertainty measurements $U_{\text{epi}}(i)$ from Eq. (6), we introduce dropout masks over visual tokens. As illustrated in Fig. 4, the projected visual-textual distributions sometimes misalign with the image content, and regions of high information can lead to substantial errors, resulting in hallucinations. Based on this intuition, we selectively target visual tokens with high epistemic uncertainties for dropout.

Specifically, we formulate a controllable series of sample distributions for visual token dropout based on $U_{\text{epi}}(i)$, for each visual position i:

$$P_{\text{dropout}}^{(k)}(x_i^v) = \gamma^{(k)} \left(\frac{U_{\text{epi}}(i) - U_{\text{epi}}^{\min}}{U_{\text{epi}}^{\max} - U_{\text{epi}}^{\min}} \right) + \delta^{(k)} \tag{9}$$

where $U_{\text{epi}}^{\min}$, $U_{\text{epi}}^{\max}$ are the minimum and maximum epistemic uncertainty values across all visual tokens, and $\gamma^{(k)}$ and $\delta^{(k)}$ are hyperparameters controlling the probability range of the dropout. By adjusting the values of $\gamma^{(k)}$ and $\delta^{(k)}$, we can modulate the intensity of visual token dropout. For further discussion on hyperparameters, see §7.2.

With the dropout distributions, we can sample dropout masks for each visual token independently. Denote the binary mask as $M^{(k)} \in \{0,1\}^N$, consisting of a binary indicator $M_i^{(k)}$ for each visual token x_i^v, where the conrresponding visual token is retained if $M_i^{(k)} = 1$, and dropped if $M_i^{(k)} = 0$. The dropout mask sampling follows $P(M_i^{(k)} = 0) = P_{\text{dropout}}^{(k)}(x_i^v)$, and the sampling is done for each visual token position independently. A higher value of $P_{\text{dropout}}(x_i^v)$ indicates that x_i^v is more likely to be dropped out. If we performed the optional preliminary forward pass to identify relevant visual token set $\mathcal{S}_j$, these visual tokens are never dropped, i.e.,, $\forall x_i^v \in \mathcal{S}_j$, set $M_i^{(k)} = 1$ directly.

Ensemble-based reliable generation. Our inference-time context dropout introduces stochasticity, so we employ an ensemble decoding approach by independently sampling K distinct dropout masks, $\{M^{(k)}\}_{k=1}^K$, to enhance generation quality. Since the masks are independent, the text generative distribution from K masks can be efficiently computed in a parallel forward pass

$$y_j^{(k)} \overset{\text{Decoding}}{\sim} p_\theta(\cdot \mid x_{/M^{(k)}}^v, x^t, y_{<j}) \tag{10}$$

where $x_{/M^{(k)}}^v$ denotes the visual tokens after applying dropout mask $M^{(k)}$, and $\overset{\text{Decoding}}{\sim}$ denotes invariance to the decoding algorithm used (e.g., greedy search in our implementation, though others are applicable).

Algorithm 1 Pseudocode of 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 visual text projecting distributions q_i^{proj}. {Eq (3)}
Compute average distribution q^{proj}. {Eq. (4)}
Compute epistemic uncertainty $U_{\text{epi}}(i)$. {Eq. (6)}
for $j = 1$ to L **do**
  Identifying relevant visual tokens (optional):
  Generate preliminary token y_j^{init}. {Eq. (7)}
  Get relevant tokens $\mathcal{S}_j$ with y_j^{init} and q_i^{proj}. {Eq. (8)}
  Visual token dropout with uncertainty-guidance:
  Get K dropout prob $P^{(k)}$ with $U_{\text{epi}}(i)$. {Eq. (9)}
  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. (10)}
  Majority voting on $y_j^{(k)}$ and get y_j.
end for
Return Generated sequence y

Each $y_j^{(k)}$ serves as a *candidate prediction* for the next text token, with the final token y_j selected via majority voting among the K masked inputs. In case of a tie, we choose the prediction from the forward pass with the fewest dropped tokens, as it retains the most information and is deemed more reliable. By forming an ensemble of predictions derived from various subsets of the visual input, enabled through token dropout, we diversify the model's perspective on the visual content. This diversity mitigates the impact of any single misinterpretation, ultimately leading to more reliable and robust generation, which is also observed in other ensemble-based methods [24, 55, 56, 57, 58].

6 Experiments

We evaluate the proposed DROPOUT DECODING from two aspects: OH reduction and overall generation quality. For OH, we use the CHAIR [33] and THRONE [34] metrics to assess the performance of different decoding methods on the MSCOCO dataset. Additionally, we employ MMBench [35] to evaluate the overall generation quality and general ability of these methods.

6.1 Experimental Setup

Base LVLMs. We evaluate all methods on three representative LVLMs: LLaVA-1.5 [49], InstructBLIP [59] and LLaVA-NEXT [16]. LLaVA-1.5 and LLaVA-NEXT use hundreds to thousands of visual tokens for detailed representation, while InstructBLIP employs just 32 tokens but with higher information density. This showcases the flexibility of our approach, effective across models with varying token counts.

Hallucination reduction baselines. In addition to the original LVLM outputs, we compare our method with beam search as well as two state-of-the-art decoding methods: VCD [45], which contrasts original and distorted visuals to reduce hallucinations, and OPERA [44], which applies penalties and token adjustments for better grounding.

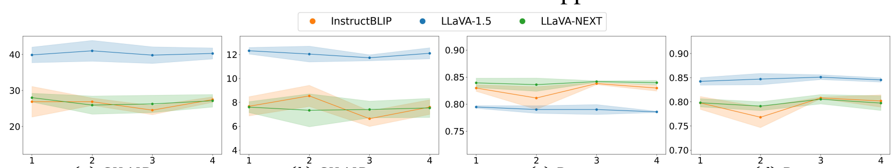
6.2 CHAIR

CHAIR [33] is a benchmark for evaluating object hallucination in image captioning. It includes two metrics: the sentence-level CHAIR$_S$, measuring the frequency of captions with hallucinated objects, and the object-level CHAIR$_I$, calculating the proportion of hallucinated objects among all objects.

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$_I$ and CHAIR$_S$ improve by 16% and 12% respectively over the second-best method. Furthermore, DROPOUT DECODING reduces the generation of hallucinated objects without compromising the inclusion of relevant objects. These improvements align with expectations that token dropout reduces generated objects.

Model	Method	CHAIR		THRONE			
		CHAIR$_S$ ↓	CHAIR$_I$ ↓	F_{all}^{1}↑	$F_{\text{all}}^{0.5}$↑	P_{all}↑	R_{all}↑
LLaVA-1.5	Greedy	42.20$_{\pm 2.86}$	12.83$_{\pm 0.36}$	0.795$_{\pm 0.006}$	0.784$_{\pm 0.009}$	0.772$_{\pm 0.015}$	0.847$_{\pm 0.010}$
	Beam Search	46.33$_{\pm 1.10}$	13.9$_{\pm 0.60}$	0.790$_{\pm 0.007}$	0.772$_{\pm 0.004}$	0.759$_{\pm 0.003}$	**0.862**$_{\pm 0.009}$
	OPERA	41.47$_{\pm 0.92}$	12.37$_{\pm 0.72}$	0.802$_{\pm 0.003}$	0.791$_{\pm 0.004}$	0.782$_{\pm 0.009}$	0.854$_{\pm 0.011}$
	VCD	49.20$_{\pm 0.88}$	14.87$_{\pm 0.47}$	0.786$_{\pm 0.012}$	0.771$_{\pm 0.017}$	0.759$_{\pm 0.020}$	0.854$_{\pm 0.015}$
	DROPOUT DECODING	39.80$_{\pm 2.3}$	**11.73**$_{\pm 0.25}$	**0.804**$_{\pm 0.002}$	**0.796**$_{\pm 0.006}$	0.790$_{\pm 0.009}$	0.851$_{\pm 0.005}$
	DROPOUT DECODING (w/o prelim)	**39.73**$_{\pm 2.15}$	12.20$_{\pm 0.70}$	0.799$_{\pm 0.002}$	0.794$_{\pm 0.004}$	**0.791**$_{\pm 0.007}$	0.843$_{\pm 0.005}$
InstructBLIP	Greedy	27.87$_{\pm 1.32}$	7.90$_{\pm 0.63}$	0.809$_{\pm 0.001}$	0.826$_{\pm 0.003}$	0.832$_{\pm 0.006}$	0.803$_{\pm 0.007}$
	Beam Search	25.87$_{\pm 2.77}$	6.93$_{\pm 0.569}$	0.809$_{\pm 0.002}$	0.827$_{\pm 0.006}$	0.836$_{\pm 0.005}$	0.807$_{\pm 0.015}$
	OPERA	28.07$_{\pm 1.75}$	8.23$_{\pm 0.53}$	0.805$_{\pm 0.004}$	0.824$_{\pm 0.003}$	0.830$_{\pm 0.004}$	0.798$_{\pm 0.008}$
	VCD	39.33$_{\pm 2.70}$	19.10$_{\pm 0.30}$	0.737$_{\pm 0.008}$	0.746$_{\pm 0.012}$	0.751$_{\pm 0.020}$	0.757$_{\pm 0.007}$
	DROPOUT DECODING	**24.53**$_{\pm 1.26}$	**6.63**$_{\pm 0.65}$	**0.814**$_{\pm 0.008}$	**0.833**$_{\pm 0.004}$	**0.838**$_{\pm 0.002}$	**0.808**$_{\pm 0.016}$
	DROPOUT DECODING (w/o prelim)	26.2$_{\pm 2.40}$	7.10$_{\pm 0.854}$	0.807$_{\pm 0.008}$	0.823$_{\pm 0.006}$	0.827$_{\pm 0.010}$	0.804$_{\pm 0.010}$
LLaVA-NEXT	Greedy	28.80$_{\pm 2.12}$	8.10$_{\pm 0.92}$	0.815$_{\pm 0.012}$	0.832$_{\pm 0.009}$	0.830$_{\pm 0.007}$	0.799$_{\pm 0.008}$
	Beam Search	28.06$_{\pm 1.30}$	**7.10**$_{\pm 0.20}$	0.816$_{\pm 0.007}$	0.834$_{\pm 0.006}$	0.834$_{\pm 0.004}$	0.801$_{\pm 0.002}$
	OPERA	29.06$_{\pm 1.89}$	8.06$_{\pm 1.07}$	0.814$_{\pm 0.011}$	0.832$_{\pm 0.011}$	0.831$_{\pm 0.006}$	0.799$_{\pm 0.007}$
	VCD	33.19$_{\pm 0.52}$	8.10$_{\pm 0.91}$	0.818$_{\pm 0.004}$	0.822$_{\pm 0.003}$	0.808$_{\pm 0.005}$	**0.822**$_{\pm 0.003}$
	DROPOUT DECODING	**26.26**$_{\pm 2.4}$	7.39$_{\pm 0.69}$	**0.821**$_{\pm 0.010}$	**0.840**$_{\pm 0.009}$	**0.842**$_{\pm 0.002}$	0.805$_{\pm 0.010}$
	DROPOUT DECODING (w/o prelim)	27.0$_{\pm 1,80}$	7.53$_{\pm 0.643}$	0.814$_{\pm 0.009}$	0.835$_{\pm 0.007}$	0.837$_{\pm 0.003}$	0.793$_{\pm 0.008}$

Table 1: Comparison of methods on CHAIR$_S$, CHAIR$_I$, F_{all}^{1}, $F_{\text{all}}^{0.5}$, P_{all}, and R_{all} metrics for LLaVA-1.5, InstructBLIP, and LLaVA-NEXT. Details of the experimental setup and the interpretation of the standard deviation can be found in the appendix. Details of the experimental setup and the interpretation of the standard deviation can be found in the appendix.



(a) CHAIR$_S$ (b) CHAIR$_I$ (c) P_{all} (d) R_{all}

Figure 3: Comparison of CHAIR$_S$, CHAIR$_I$, P_{all} and R_{all} scores with standard deviations across different candidate numbers.

6.3 THRONE

THRONE [34] 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 [43]). It uses P_{all} (Precision), R_{all} (Recall), F_{all}^{1}, and $F_{\text{all}}^{0.5}$. Additionally, it employs F$_\beta$, 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 1 illustrate that DROPOUT DECODING surpasses nearly all baseline methods across various metrics, highlighting its effectiveness in reducing 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}. Across models, 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 large increase in $F_{\text{all}}^{0.5}$ further shows its comprehensiveness.

6.4 MMBench

MMBench [35] is a comprehensive benchmark designed to evaluate the multimodal capabilities of LVLMs across various tasks and data types. Since the prompt length limits in MMBench exceed InstructBLIP's token allowance, we report results only on LLaVA-1.5 and LLaVA-NEXT.

Results. As shown in Table 2, DROPOUT DECODING outperforms all the other baselines on LLaVA-1.5, which demonstrates its robustness and adaptability across a broader range of multimodal tasks.

Method	Original	VCD	OPERA	DROPOUT DECODING
LLaVA-1.5	71.86	72.35	73.86	**74.01**
LLaVA-NEXT	**74.57**	69.65	74.54	74.31

Table 2: Results on MMBench. Higher is better.

7 Analysis and Ablation Studies

7.1 Efficiency Analysis

We conducted a thorough analysis of computational overhead, measuring throughput and wall-time to evaluate efficiency. Table 7.1 summarizes these results. Our method introduces additional overhead primarily in two aspects: (1) a preliminary forward pass for identifying relevant visual tokens, (2) performing K parallel forward passes using varied dropout masks.

The preliminary forward pass, though beneficial, is optional. Omitting it results in only approximately 7% throughput reduction compared to greedy decoding, while still consistently improving performance metrics across benchmarks. Furthermore, the method efficiently handles the K parallel passes by batching identical inputs with distinct dropout masks into a single batched operation, significantly reducing additional computational overhead.

In terms of GPU memory, we verified efficiency under realistic conditions. Using vLLM and LLaVA-1.5 on 4×A800 80GB GPUs, GPU memory usage was 38.12 GB with efficient KV caching, unchanged between greedy decoding and our method without preliminary passes. Confirmatory experiments under Huggingface Transformers similarly demonstrated minimal GPU memory increase (from 14.02 GB to 15.31 GB), indicating negligible impact on inference constraints.

The cost of computing uncertainty metrics is explicitly included in our benchmarks and remains negligible. With LLaVA-1.5 and 576 image tokens, computing uncertainty adds only 73.30 ms per input, a minor cost amortized across the batched forward passes.

Overall, our approach effectively balances efficiency and performance.

Metric	Greedy	Ours w/o prelim	Ours w/ prelim
Throughput (tok/s) ↑	37.0	34.1 (-7.8%)	20.1 (-45.7%)
Wall-time (per 50 tok) ↓	1.35	1.47 (+8.9%)	2.49 (+84.4%)

Table 3: Computational overhead analysis.

7.2 Parallel Dropouts Hyperparameters

As in §5.2, we generate K candidate predictions using token dropout. This section examines how varying the hyperparameters impacts generation quality. We fix $\delta^{(k)} = 0.1$ and adjust $\gamma^{(k)}$ based on a predefined order: $\gamma^{(1)} = 0.3$, $\gamma^{(2)} = 0.5$, and $\gamma^{(3)} = 0.7$. However, setting $\gamma^{(4)} = 0.9$ excessively drops visual tokens and degrades InstructBLIP's performance, so we set $\gamma^{(4)} = 0.1$. Moreover, our majority voting favors candidates with fewer dropped tokens in ties. To avoid identical outputs when comparing only two candidates, we remove Candidate 1 in the second round.

As shown in Fig. 3 (a) and (b), both CHAIR_S and CHAIR_I scores peak at $K = 3$ for LLaVA-1.5 and InstructBLIP. Increasing K to 4 introduces a less-masked candidate that slightly negatively impact our method's effectiveness in reducing hallucinations. Conversely, using fewer candidates (e.g. , only Candidate 1/2) lacks the balance needed for stable voting outcomes, resulting in increased randomness. Similarly, Fig. 3 (c) and (d) shows that THRONE's R_{all} and P_{all} metrics also perform best at $K = 3$. Overall, we find that selecting three candidates strikes the optimal balance between increased certainty from additional votes and the controlled uncertainty introduced by candidate dropout probability, allowing DROPOUT DECODING to achieve more trustworthy and stable generation results.

7.3 Preliminary Forward Pass

As discussed in §5.2, DROPOUT DECODING may employ a preliminary forward pass to retain most relevant objects during generation, which helps reduce hallucinated objects while maintaining high-quality outputs. In contrast, bypassing this step risks masking relevant visual tokens during the token dropout phase, potentially degrading overall performance. However, incorporating a preliminary forward pass roughly doubles the computational cost per generation. Specifically, our goals are: 1) to confirm the effectiveness of the preliminary forward pass, and 2) to explore a more efficient alternative when computational resources are limited.

As shown in Table 1, including the preliminary forward pass consistently improves most metrics, with particular notable gains in the F_{all} score on THRONE. Interestingly, for LLaVA-1.5 and LLaVA-NEXT, the variant without the preliminary pass still outperforms other baselines in most metrics. We hypothesize that this discrepancy arises from differences in the abundance of visual tokens, as LLaVA-1.5 and LLaVA-NEXT have hundreds or thousands of visual tokens. While InstructBLIP only has 32, making each token's contribution more critical. Consequently, omitting the preliminary forward pass in InstructBLIP risks losing critical information, lowering performance. These findings suggest that while a preliminary forward pass is highly beneficial for LVLMs, models with more tokens may achieve better efficiency and performance by skipping this step.

7.4 Necessity of Uncertainty Guidance on Masking

As discussed in §5.1, DROPOUT DECODING incorporates epistemic uncertainty in the masking process. To validate the necessity of this approach, we compare it with a random masking strategy, which replaces uncertainty with a random method. As shown in table Table 4, although random masking performs better on the CHAIR metric, it struggles with BLEU and fails to compute the THRONE metric. We find that random masking causes repetitive token generation (e.g., "*apple apple apple...*"), artificially inflating the CHAIR score. This happens because random masking disrupts contextual information, leading to faulty generation. In contrast, our uncertainty-based approach selectively masks uncertain tokens, preserving the context and ensuring more coherent and accurate sequences.

Model	Method	CHAIR		BLEU	THRONE
		$\text{CHAIR}_S \downarrow$	$\text{CHAIR}_I \downarrow$	BLEU $\uparrow$	THRONE $\uparrow$
LLaVA-1.5	Greedy	$42.20_{\pm 2.86}$	$7.90_{\pm 0.63}$	$11.62_{\pm 0.09}$	Table 1
	Uncertainty-guided	$39.80_{\pm 2.3}$	$11.73_{\pm 0.25}$	$11.64_{\pm 0.12}$	Table 1
	Random	$35.93_{\pm 0.90}$	$8.57_{\pm 0.63}$	$11.51_{\pm 0.12}$	Error
InstructBLIP	Greedy	$27.87_{\pm 1.32}$	$7.90_{\pm 0.63}$	$12.70_{\pm 0.54}$	Table 1
	Uncertainty-guided	$24.53_{\pm 1.26}$	$6.63_{\pm 0.65}$	$12.30_{\pm 0.18}$	Table 1
	Random	$23.80_{\pm 1.28}$	$5.07_{\pm 0.71}$	$10.88_{\pm 0.08}$	Error

Table 4: Comparison of masking strategies on CHAIR_S, CHAIR_I, BLEU and THRONE metrics for LLaVA-1.5, InstructBLIP.

7.5 High-Confidence Token Masking Analysis

To further investigate the robustness of our uncertainty-guided masking, we conducted an additional ablation experiment focusing on the opposite condition—masking high-confidence tokens instead of low-confidence ones.

Following the identical experimental setup as in Table 1 (our main CHAIR and THRONE evaluation), we replaced low-confidence masking with high-confidence masking. The results are summarized in Table 7.5.

Model	$\text{CHAIR}_S \downarrow$	$\text{CHAIR}_I \downarrow$	$F_{\text{all}} \uparrow$	$F_{0.5,\text{all}} \uparrow$	$P_{\text{all}} \uparrow$	$R_{\text{all}} \uparrow$
LLaVA	41.62	12.00	0.798	0.784	0.774	0.858
LLaVA-NEXT	27.61	7.82	0.803	0.823	0.828	0.789
InstructBLIP	29.50	9.01	0.808	0.825	0.825	0.794

Table 5: Effect of masking high-confidence tokens on CHAIR and THRONE metrics.

As shown, the results are generally worse than masking low-confidence tokens (i.e., masking high-uncertainty tokens as in our proposed method) and remain close to the greedy decoding baseline. This suggests that dropping high-confidence tokens has limited influence on generation quality—these tokens correspond to regions where the model is already certain, typically associated with background or redundant patches. Consequently, their removal produces only minor perturbations.

In contrast, masking low-confidence tokens directly influences the model's generative process, as these tokens are uncertain yet potentially informative (often corresponding to salient or ambiguous visual regions). Masking them introduces meaningful variability, thereby improving robustness and reducing hallucination frequency. This further validates our uncertainty-guided masking strategy as both effective and theoretically grounded.

8 Conclusion

We introduce DROPOUT DECODING, a novel uncertainty-guided context selective decoding approach aimed at enhancing the reliability of LVLMs. After quantifying the uncertainty in visual inputs, DROPOUT DECODING accordingly drops out visual tokens to regularize uncertainty and employs an ensemble-based decoding approach to stabilize generation. Extensive experiments on CHAIR, THRONE, and MMBench validate the effectiveness with consistent improvements over existing methods in both hallucination reduction and general multimodal capability.

References

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

[2] Wanhua Li, Renping Zhou, Jiawei Zhou, Yingwei Song, Johannes Herter, Minghan Qin, Gao Huang, and Hanspeter Pfister. 4d langsplat: 4d language gaussian splatting via multimodal large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22001–22011, 2025.

[3] 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.

[4] 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.

[5] 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.

[6] Devaansh Gupta, Siddhant Kharbanda, Jiawei Zhou, Wanhua Li, Hanspeter Pfister, and Donglai Wei. Cliptrans: transferring visual knowledge with pre-trained models for multimodal machine translation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2875–2886, 2023.

[7] Wanhua Li, Zibin Meng, Jiawei Zhou, Donglai Wei, Chuang Gan, and Hanspeter Pfister. Socialgpt: Prompting llms for social relation reasoning via greedy segment optimization. *Advances in Neural Information Processing Systems*, 37:2267–2291, 2024.

[8] 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.

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

[10] 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.

[11] Yanhong Li, Zixuan Lan, and Jiawei Zhou. Text or pixels? evaluating efficiency and understanding of llms with visual text inputs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, Suzhou, China, November 2025. Association for Computational Linguistics.

[12] 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.

[13] Yixiong Fang, Tianran Sun, Yuling Shi, and Xiaodong Gu. Attentionrag: Attention-guided context pruning in retrieval-augmented generation, 2025.

[14] Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. See what you are told: Visual attention sink in large multimodal models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[15] 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.

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

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

[18] 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.

[19] Mingyang Song, Xiaoye Qu, Jiawei Zhou, and Yu Cheng. From head to tail: Towards balanced representation in large vision-language models through adaptive data calibration. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9434–9444, 2025.

[20] 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.

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

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

[23] 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.

[24] 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*.

[25] 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.

[26] Tanqiu Jiang, Jiacheng Liang, Rongyi Zhu, Jiawei Zhou, Fenglong Ma, and Ting Wang. Robustifying vision-language models via dynamic token reweighting. *arXiv preprint arXiv:2505.17132*, 2025.

[27] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[28] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR.

[29] 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.

[30] 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.

[31] 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.

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

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

[34] 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.

[35] 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.

[36] 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.

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

[38] 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.

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

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

[41] 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.

[42] 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.

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

[44] 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.

[45] 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.

[46] Wenbin An, Feng Tian, Sicong Leng, Jiahao Nie, Haonan Lin, QianYing Wang, Guang Dai, Ping Chen, and Shijian Lu. Agla: Mitigating object hallucinations in large vision-language models with assembly of global and local attention, 2024.

[47] Xin Zou, Yizhou Wang, Yibo Yan, Sirui Huang, Kening Zheng, Junkai Chen, Chang Tang, and Xuming Hu. Look twice before you answer: Memory-space visual retracing for hallucination mitigation in multimodal large language models, 2024.

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

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

[50] 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.

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

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

[53] Zhuokai Zhao, Yibo Jiang, and Yuxin Chen. Direct acquisition optimization for low-budget active learning. *arXiv preprint arXiv:2402.06045*, 2024.

[54] Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens, 2023.

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

[56] 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.

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

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

[59] 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.

A Details of Uncertainty Decomposition

A detailed derivation of Eq. (5):

$$\begin{aligned}
U_{\text{total}} &= \mathbb{H}\left[q^{\text{proj}}\right] \\
&= -\sum_{y\in\mathcal{V}} q^{\text{proj}}(y)\log q^{\text{proj}}(y) \\
&= -\sum_{y\in\mathcal{V}} \left(\mathbb{E}_i\left[q_i^{\text{proj}}(y)\right]\right)\log q^{\text{proj}}(y) \\
&= \mathbb{E}_i\left[-\sum_{y\in\mathcal{V}} q_i^{\text{proj}}(y)\log q^{\text{proj}}(y)\right] \\
&= \mathbb{E}_i\left[\mathrm{CE}\left(q_i^{\text{proj}}, q^{\text{proj}}\right)\right] \\
&= \mathbb{E}_i\left[\mathbb{H}\left[q_i^{\text{proj}}\right] + D_{\mathrm{KL}}\left(q_i^{\text{proj}} \parallel q^{\text{proj}}\right)\right] \\
&= \mathbb{E}_i\left[U_{\text{ale}}(i) + U_{\text{epi}}(i)\right]
\end{aligned} \tag{11}$$

B Implementation Details

Our experiment is conducted on the MSCOCO 2014 test set, where we randomly sample 500 images across 3 random seeds. The average and standard deviation across these seeds are reported in our result table. The prompt used for the images is "Describe the image."

The experimental setup of DROPOUT DECODING is shown in Table 6. 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	**CHAIR**	**THRONE**	**MMBench**
	512	512	256
Top-k		False	
Top-p		1	
Temperature τ		1	
Number Beams		1	

Table 6: Parameter settings used in our experiments.

In addition to general generation settings, DROPOUT DECODING includes hyperparameters specified in §5.2. The details of these hyperparameter settings are provided below:

Top-k in identifying relevant visual tokens. Before the decoding process, we first obtain q^{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 $\delta^{(k)} = 0.1, \gamma^{(k)} = 0.2 * k + 0.1; k = 1, 2, ..., K; K = 3$ in our experiment settings.

Moreover, we provide the hyperparameter settings of our baselines. OPERA's hyperparameters can be referred to Table 7; VCD's hyperparameters can be referred to Table 8.

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

Table 7: OPERA hyperparameter settings.

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

Table 8: VCD hyperparameter settings.

C Details of Masked Tokens

We measured the average masked visual tokens of the selected candidate in decoding, demonstrating the effectiveness of our method. For reference, the total number of visual tokens varies across models: LLaVA uses 576 visual tokens, InstructBLIP processes 32 visual tokens, while LLaVA-NEXT typically generates around 2,000 tokens depending on the resolution and patching strategy.

	LLaVA	**LLaVA-NEXT**	**InstructBLIP**
Masked Tokens	244.14	879.53	13.67

Table 9: Average number of masked tokens in the selected candidate

Impact Statement

This paper aims to advance the field of Machine Learning. While our work has potential societal implications, we do not identify any immediate concerns that require specific attention.

Limitations

While DROPOUT DECODING improves reliability by suppressing uncertain visual tokens during inference, our method has several limitations. First, the additional forward pass in decoding, which introduces computational overhead. Second, the effectiveness of epistemic uncertainty estimation relies on the quality of text-space projections, which may be influenced by the quality of projector.

D Generation Examples and Analysis

D.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.

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.

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.

D.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.

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.

D.3 LLaVA-NEXT Examples

Generated Caption:

Greedy Generated caption:

In the heart of a bustling pizza shop, a woman stands, her attention captivated by the slice of pizza she holds in her hand. The pizza, with its golden crust and vibrant toppings, is a tantalizing sight. She's dressed casually, her green jacket contrasting with the warm tones of the shop.The shop itself is a feast for the eyes. A large refrigerator stands in the background, its surface adorned with various signs and posters. The words "We guarantee the best pizza" are prominently displayed, a testament to the shop's commitment to quality. ...

Dropout Decoding Generated caption:

In the heart of a bustling pizza shop, a woman stands, her attention captivated by the slice of pizza she holds in her hand. The pizza, with its golden crust and vibrant toppings, is a tantalizing sight. She's dressed casually, her green jacket contrasting with the warm tones of the shop.The shop itself is a feast for the eyes. Boxes of pizza are neatly stacked on shelves, their white and red colors standing out against the darker tones of the shop. A sign on the wall proudly declares "We guarantee the best pizza in town", a testament to the shop's confidence in its offerings. ...

Generated Caption:

Greedy Generated caption:

The image depicts a cozy indoor setting with a focus on a wooden table. The table is set with a few items, including a plate of food, a glass of water, and a small bowl. There are also a couple of utensils, a fork and a knife, placed on the table. The table is positioned in front of a large window that offers a view of a patio area with a bamboo fence and a small table. The room has a warm and inviting atmosphere, with a wooden floor and a beige cushion visible in the foreground. The lighting suggests it might be daytime. There are no visible texts or distinctive brands in the image. The style of the image is a candid, real-life photograph with a focus on the table setting and the view outside the window.

Dropout Decoding Generated caption:

The image depicts a cozy indoor setting with a focus on a wooden table. The table is set with a few items, including a plate with a small portion of food, a glass of water, and a pair of chopsticks. There are also a couple of small bowls and a spoon on the table. The table is positioned in front of a large window that offers a view of a wooden deck outside. The deck is furnished with a bench and a small table, and there are bamboo plants and a tree visible in the background. The room has a warm and inviting atmosphere, with a wooden floor and a cushion on the floor to the left of the table. The lighting suggests it might be daytime, and the overall style of the image is natural and serene.

D.4 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.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our claims in the abstract and introduction accurately reflect the paper's contribution and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are listed in the appendix.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide the full set of assumptions and a complete proof.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We fully disclose all the information to reproduce our results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide open access to the data and code.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We include no training in our work, and we provide all test details in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report the error bars in the experiment.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the efficiency analysis for time of execution.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We follow the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our work has no societal impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original paper for each asset.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Our work doesn't release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We do not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We do not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: Our core method development does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.