Learning to Mask: Cross-Modal Noise Modulation for Hallucination Mitigation in Multi-modal Large Language Models

Zijian Song[1], Chunlei Wang[2], and Kun He[1†]

[1] School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China
[2] China Information Security Research Institute Company Ltd., Beijing, China
songzijian88@hust.edu.cn, wangchl@cecr.ac.cn, brooklet60@hust.edu.cn

Abstract. Multi-modal Large Language Models have demonstrated impressive capabilities in vision-language tasks, yet they remain highly susceptible to hallucinations, in which generated text diverges from the actual visual evidence. Current mitigation strategies, such as self-refinement and specialized decoding, often incur significant latency due to multi-round iterations or reliance on high-performance auxiliary models, limiting their real-time practical utility. In this paper, we systematically analyze the hidden states of MLLMs across the entire inference pipeline and reveal that hallucination-inducing redundant features persist not only in image tokens during the prefill stage but also propagate into Image and Answer Value Caches during decoding. Motivated by the observation that Gaussian noise injection can effectively suppress these redundancies, we propose HalMask (**Hal**lucination Feature **Mask**ing), a lightweight and real-time self-correction framework. HalMask employs specialized Image and Text Noise Modulation Modules, optimized via a dynamic regularization term, to accurately discern and mask hallucination-triggering features. During inference, it applies a dynamic Value Cache masking mechanism that continuously filters redundant information across both generation phases without disrupting critical semantic cues. Extensive experiments on multiple benchmarks demonstrate that HalMask significantly reduces hallucinations while introducing only marginal computational overhead.

Keywords: Multi-modal Large Language Models · Hallucination Mitigation

1 Introduction

Multi-modal Large Language Models (MLLMs) have achieved remarkable breakthroughs in tasks such as image captioning, visual question answering, and multimodal reasoning [21, 22, 33, 36]. Despite their proficiency in parsing visual content, these models remain vulnerable to hallucinations—a phenomenon where

† Corresponding author.

generated textual descriptions deviate from the actual visual evidence [5, 11, 18]. Consequently, addressing hallucinations is a pivotal challenge in achieving robust and trustworthy multimodal reasoning [8, 10, 19, 24].

To mitigate this issue, existing literature has predominantly focused on self-refinement [7,29,31] and decoding strategies [14–16,32]. However, self-refinement techniques typically necessitate high-performance auxiliary models, while specialized decoding strategies often rely on multi-round iterations and repetitive backtracking. These dependencies introduce significant latency, compromising inference throughput and practical utility. Therefore, there is an urgent need for an efficient, real-time solution capable of self-correction during the decoding process to suppress hallucinations.

While the hidden states in MLLMs encapsulate rich semantic and contextual information to facilitate expressive generation, it remains unclear whether they also harbor redundant information that induces hallucinations. While prior works [3,6,37] examined how image tokens during the prefill stage may contribute to hallucinations, a comprehensive analysis of hidden states throughout the entire inference pipeline, specifically concerning hallucination-triggering redundancies, remains lacking. We conduct a systematic and holistic analysis of hidden states during both the prefill and decoding phases of MLLMs. Our findings reveal three key insights (see Section 3.1 and Section 3.2): **(1) Hallucination-inducing redundant features are present not only in the image tokens of the prefill stage but also within the Image Value Cache and Answer Value Cache during the decoding phase; (2)Injecting Gaussian noise into these redundant features effectively suppresses the generation of hallucinated content; (3) The control coefficient for noise injection plays a decisive role in mitigation efficacy. While an optimal coefficient significantly reduces hallucinations, an improperly tuned one may obscure salient information, thereby inadvertently exacerbating the hallucination effect.**

Inspired by these observations, we propose **Hal**lucination Feature **Mask**ing (HalMask), a lightweight framework designed to suppress hallucination-inducing features via Gaussian noise injection. At its core, HalMask employs specialized Image and Text Noise Modulation Modules, which are trained on a hallucination ground truth text dataset to discern hallucination-inducing redundant features within multimodal embeddings. During the training phase, we introduce a dynamical regularization term to constrain the modulation modules, forcing the learned feature dynamics to converge toward a standard normal distribution, thereby facilitating precise stochastic masking of hallucination-inducing features. For inference, we introduce a tailored masking strategy across two distinct generation stages of MLLMs: (1) **Prefill stage**: we inject Gaussian noise into the hallucination-prone features within the image tokens; (2) **Decoding stage**: we implement a dynamic Value Cache masking mechanism. Specifically, as each new answer token is generated, it is aggregated with preceding tokens and processed by the Noise Modulation Modules to derive image and text modulation coeffi-

cients. These coefficients then govern the injection of Gaussian noise into the image and text Value Caches across all layers of the LLM decoder.

To evaluate the efficacy of HalMask in mitigating hallucinations, we conduct extensive experiments on CHAIR, POPE, OPOPE, AMBER, MMBench, and ScienceQA. Experimental results demonstrate that HalMask significantly reduces hallucinations while maintaining a marginal and acceptable computational overhead. Our main contributions are as follows:

- We propose HalMask, a novel framework designed to mitigate hallucinations by selectively masking hallucination-inducing features within both visual and textual modalities.
- HalMask comprises four core components that operate synergistically: the Image Noise Modulation Module and the Text Noise Modulation Module, which generate adaptive control coefficients to inject Gaussian noise into image and text features; and the Dynamic Regularization Term, which prevents the suppression of critical semantic cues; and the Dynamic Value Cache Masking mechanism, which eliminates redundant information at the cache level.
- Extensive experiments across diverse benchmarks and MLLMs validate HalMask's effectiveness, demonstrating significant performance gains with minimal and acceptable computational overhead.

2 Related Work

Multi-modal Large Language Models. Recent advances have led to the emergence of MLLMs that incorporate visual inputs to enhance their capabilities. LLaVA-1.5 [20] improves instruction-following via visual instruction tuning; MiniGPT-4 [36] bridges vision encoders with LLMs for unified multimodal understanding; and Qwen2.5-VL enhances spatial reasoning and visual grounding across diverse resolutions.

Hallucination in MLLMs. Despite their progress, MLLMs still suffer from hallucinations—generating descriptions misaligned with the visual content. To address this, recent works focus on self-correction mechanisms: Woodpecker [31] detects and rectifies hallucinations through visual verification; OPERA [14] exploits attention patterns for self-correction; and HALC [7], SID [15], and DeGF [32] mitigate hallucinations by providing self-feedback through contrastive decoding to verify and correct the initial response.

3 Preliminaries

3.1 Text-Induced Hallucination Mechanisms in Autoregressive Generation

Multi-modal Large Language Models (MLLMs) typically adopt an autoregressive generation paradigm, in which each new token is generated conditioned on previously generated tokens. This generation mechanism implies that earlier tokens

can propagate errors and potentially induce hallucinated content in subsequent outputs. To investigate whether prior textual tokens trigger hallucinations in newly generated tokens, we generate hallucinated texts using 500 images from COCO2014. These hallucinated texts, together with the prompt "Please describe this image in detail", are fed into LLaVA-1.5-7B. The CHAIR metric is used to evaluate the severity of hallucinations in the resulting image descriptions.

To further analyze this effect, we inject noise into the hallucinated texts by controlling a coefficient α, enabling us to examine how different levels of hallucinated information influence the generated descriptions:

$$X_{hal}^{control} = \alpha \cdot X_{hal} + (1 - \alpha) \cdot \epsilon, \tag{1}$$

where X_{hal} denotes the hallucinated text tokens and $\varepsilon \sim \mathcal{N}(0, 1)$ represents Gaussian noise. When $\alpha = 1$, the complete hallucinated text is fed into LLaVA-1.5-7B; when $\alpha = 0$, the hallucinated text is fully replaced by Gaussian noise.

As shown in Figure 1, comparing the cases $\alpha = 1$ and $\alpha = 0$ reveals that hallucinated texts significantly induce hallucinations in MLLMs. As α decreases from 1 to 0, the $\mathbf{CHAIR}_S$ score decreases from 72.8 to 20.8, while $\mathbf{CHAIR}_I$ decreases from 22.6 to 6.7, demonstrating a gradual reduction in hallucinations as Gaussian noise replaces hallucinated text. These results suggest that hallucinated textual tokens propagate and amplify hallucinations in MLLMs, while perturbing these tokens with Gaussian noise effectively mitigates this effect.

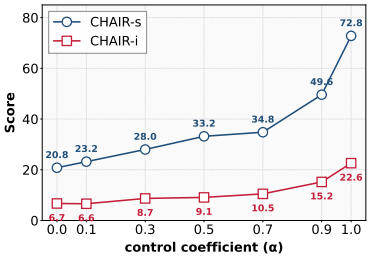


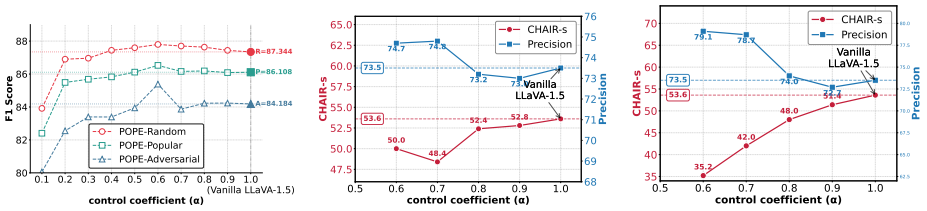
Fig. 1: Impact of the control coefficient on hallucination behavior. Decreasing α reduces hallucination (CHAIR), suggesting that hallucinations can be mitigated by controlling the strength of Gaussian noise injected into hallucination-inducing text.

The above results raise a critical question: **Do MLLMs contain hidden states that can induce hallucinations during inference?**

3.2 Multimodal Hallucination Mitigation via Redundant Feature Suppression in Prefill and Decoding Stages

MLLMs typically comprise two distinct stages during inference: the prefill stage and the decoding stage.

Prefill stage. During the prefill stage, image tokens and instruction text tokens are jointly processed by the LLM to generate the initial response tokens. As instruction texts are user-provided inputs (e.g., "Please describe this image in detail"), they are assumed to be hallucination-free. To investigate whether image tokens contain hallucination-inducing features, we compute the cosine similarity



(a) (b) (c)

Fig. 2: Accuracy comparison under Gaussian noise injection at different stages.**(a)** POPE F_1 scores during the prefill stage after injecting Gaussian noise into the bottom 30% of image tokens with the lowest similarity (controlled by coefficient α). **(b)** and **(c)** show the trends for $CHAIR_s$ and Precision during the decoding stage when noise is injected into the bottom 30% of the image Value Cache and the bottom 10% of the answer text Value Cache with the lowest similarity, respectively.

between image tokens and instruction text tokens using LLaVA-1.5-7B on the POPE dataset. We then select the bottom 30% of image tokens with the lowest similarity and inject Gaussian noise as:

$$z_v = \alpha \cdot x_v^{\text{bottom30\%}} + (1 - \alpha) \cdot \epsilon, \tag{2}$$

where $x_v^{\text{bottom30\%}}$ denotes the least similar image tokens. As illustrated in Figure 2(a), results show that multiple α settings outperform the baseline ($\alpha = 1.0$, Vanilla LLaVA-1.5) across POPE benchmarks. Notably, $\alpha = 0.6$ achieves a 0.69% improvement on the average POPE score. These findings indicate that image tokens contain redundant features, and suppressing them through Gaussian noise injection can effectively mitigate hallucinations in MLLMs.

Decoding stage. To investigate whether Value Cache redundancy triggers hallucinations, we analyze LLaVA-1.5-7B on the CHAIR dataset by injecting Gaussian noise into the bottom 30% of image tokens and 10% of answer text tokens (based on cosine similarity).

As shown in Figure 2(b-c), the decrease in CHAIR_S after noise injection confirms that these redundant features contain hallucination-inducing information. These results indicate: (1) Redundant hallucination-inducing features exist both in image tokens during the prefill stage and in the Value Cache during decoding; (2) Injecting Gaussian noise into these features can effectively mitigate hallucinations; (3)The effectiveness of this strategy is highly sensitive to the noise control coefficient: while a well-chosen coefficient reduces hallucinations, an inappropriate one may obscure useful information and even exacerbate hallucinations.

Therefore, we aim to develop a method that can automatically regulate the noise coefficient to inject properly calibrated Gaussian noise into redundant features, thereby reducing hallucinations without sacrificing useful information or introducing excessive computational overhead.

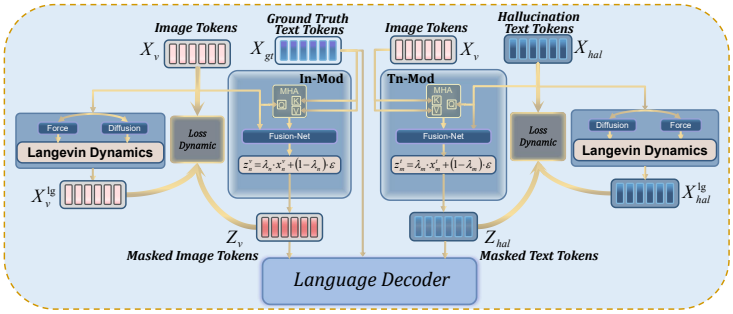


Fig. 3: The illustration of the proposed HalMask training framework.

4 The Proposed Method

4.1 Overview

To eliminate redundant information that causes hallucinations, we use control coefficients Λ_{image} and Λ_{text} to inject Gaussian noise into image features and text features, respectively. Specifically, we utilize a hallucination ground truth text dataset to train the Image Noise Modulation Module (In-Mod) and Text Noise Modulation Module (Tn-Mod) to generate Λ_{image} and Λ_{text}. The training architecture is illustrated in Figure 3. During inference, in the prefill stage, we use Λ_{image} to inject Gaussian noise into image tokens. In the decoding stage, we use Λ_{image} and Λ_{text} to inject Gaussian noise into the image value cache and the answer text value cache, respectively. The inference architecture is illustrated in Figure 4.

In the following sections, we introduce the training method of the two Noise Modulation Modules in Section 4.3, and describe the inference process of Noise Modulation during the prefill and decoding stages in Section 4.4.

4.2 Cross-Modal Noise Modulation

To eliminate redundant information in features, we introduce two modality-specific noise modulation modules: **Image Noise Modulation Module (In-Mod)** and **Text Noise Modulation Module (Tn-Mod)**. These modules adaptively inject Gaussian noise into multimodal features to mask redundant information, thereby preventing such redundancy from interfering with the prediction of subsequent tokens. Let $\mathbf{X}_v = [x_1^v, \ldots, x_N^v] \in \mathbb{R}^{N \times d}$ denote the image tokens extracted by the image encoder, where N is the number of image tokens and d is the feature dimension. Let $\mathbf{X}_{gt} = [x_1^{gt}, \ldots, x_M^{gt}] \in \mathbb{R}^{M \times d}$ and $\mathbf{X}_{hal} = [x_1^{hal}, \ldots, x_K^{hal}] \in \mathbb{R}^{K \times d}$ denote the Ground Truth (GT) text tokens and Hallucination text tokens, respectively, where M and K are the numbers of GT text tokens and Hallucination text tokens, respectively.

Image Noise Modulation Module. To enable the model to adaptively mask redundant information in image tokens based on variations in autoregressive text during inference, we incorporate Ground Truth text information during training. This allows In-Mod to fully leverage textual cues and perform precise, targeted masking of redundant image token information.

Due to the differing lengths of image tokens and GT text tokens, we employ an attention mechanism to project the GT text tokens onto the length of the image tokens, guided by cross-modal correlation. Specifically, we use $\mathbf{X}_v$ as the attention query, and $\mathbf{X}_{gt}$ as the attention keys and values. The projected GT text tokens are computed as follows:

$$\mathbf{X}_{gt}^{attn} = Softmax\left(\frac{\mathbf{X}_v \mathbf{X}_{gt}^T}{\sqrt{d}}\right)\mathbf{X}_{gt}, \tag{3}$$

where d denotes the dimension of $\mathbf{X}_v$.

After obtaining the projected GT text tokens $\mathbf{X}_{gt}^{attn} \in \mathbb{R}^{N\times d}$ and the image tokens $\mathbf{X}_v \in \mathbb{R}^{N\times d}$, we fuse them along the feature dimension. Specifically, we concatenate the textual and visual features to derive a multimodal representation:

$$\mathbf{X}_{merge} = \text{Concat}\left(\mathbf{X}_{gt}^{attn}, \mathbf{X}_v\right) \in \mathbb{R}^{N\times 2d}. \tag{4}$$

Subsequently, the multimodal representation is fed into Fusion-Net, which comprises five linear layers interspersed with ReLU activation functions, concluding with a sigmoid activation that outputs a control coefficient $\Lambda_{image} = \{\lambda_1^v, \lambda_2^v, \ldots, \lambda_N^v\} \in \mathbb{R}^{N\times d}$.

Text Noise Modulation Module. To accurately mask hallucination text tokens $\mathbf{X}_{hal}$, we employ an attention mechanism to integrate information between image tokens and hallucination text tokens. Specifically, $\mathbf{X}_{hal}$ serves as the attention query, while $\mathbf{X}_v$ functions as both the attention keys and values. The mapped image tokens are represented as follows:

$$\mathbf{X}_{v}^{attn} = Softmax\left(\frac{\mathbf{X}_{hal} \mathbf{X}_{v}^T}{\sqrt{d}}\right)\mathbf{X}_{v}, \tag{5}$$

where d denotes the dimension of $\mathbf{X}_{gt}$.

After obtaining the mapped image tokens $\mathbf{X}_v^{attn} \in \mathbb{R}^{K\times d}$ and the hallucination text tokens $\mathbf{X}_{hal} \in \mathbb{R}^{K\times d}$, we fuse them along the feature dimension by concatenating the textual and visual features. This yields a multimodal fusion representation:

$$\mathbf{X}_{merge} = \text{Concat}\left(\mathbf{X}_{v}^{attn}, \mathbf{X}_v\right) \in \mathbb{R}^{K\times 2d}. \tag{6}$$

Subsequently, the multimodal representation is passed through the Fusion-Net to generate a control coefficient $\Lambda_{text} = \{\lambda_1^t, \lambda_2^t, \ldots, \lambda_K^t\} \in \mathbb{R}^{K\times d}$.

Injecting Gaussian Noise for Masking Redundant Features. After obtaining the control coefficients Λ_{text} and Λ_{image} corresponding to image tokens and hallucination text tokens through In-Mod and Tn-Mod, we further utilize these control coefficients to mask redundant information within the image tokens and hallucination text tokens that may induce hallucinations. The masked token feature z_n is defined as:

$$z_n = \lambda_n \cdot x_n + (1 - \lambda_n) \cdot \epsilon, \tag{7}$$

where $\epsilon \sim \mathcal{N}(0, 1)$ is Gaussian noise, and $\lambda_n \in [0, 1]$ is a learnable control coefficient. This formulation enables token feature information compression: when λ_n is close to 0, the original feature x_n is largely replaced by noise, effectively masking redundant information. Conversely, when λ_n is close to 1, the feature is preserved. Finally, we obtain the masked image tokens $\mathbf{Z}_v$ and hallucination text tokens $\mathbf{Z}_{\text{hal}}$.

4.3 Training Phase

Hallucination-Ground Truth Text Dataset Construction. Following [35], we construct hallucination-Ground Truth text dataset comprising triplets of images, Ground Truth (GT) descriptions, and synthetic hallucinations. Starting with strictly aligned image-text pairs as GT, we synthesize hallucinated counterparts by introducing three targeted perturbations: co-occurrence, object uncertainty, and spatial positioning.

Dynamical Regularization Term. Existing methods [1, 2, 12, 17, 28, 34] often mask redundant features by setting the optimization objective to drive the feature distribution toward a Gaussian. However, imposing a simplistic Gaussian prior can lead to excessive feature regularization [13], suppressing not only redundant but also potentially valuable information. To address this issue, we adopt a dynamical systems perspective to perform feature regularization. Specifically, we employ a **dynamical regularization term** to encourage In-Mod and Tn-Mod to learn the dynamical properties that drive features toward a standard normal distribution, rather than directly forcing the features themselves to match a simple normal distribution.

In this paper, we employ the overdamped Langevin equation as the dynamical system [9]. The overdamped Langevin dynamics is given by:

$$dx = (-M(x)\nabla V(x) + \text{div}(M(x)))\, dt + \sqrt{2}M^{1/2}(x)dW_t, \tag{8}$$

where $M = [M_1, \ldots, M_d]$ denotes the diffusion matrix, V is the potential energy, W_t is a standard multidimensional Brownian motion, and $\text{div}M = (\text{div}M_1, \ldots, \text{div}M_d)^T$ represents the divergence of the i-th column of the diffusion matrix M.

We discretize the overdamped Langevin dynamics using the Euler-Maruyama method with a time step $dt = 1$, resulting in the following expression:

$$\Delta x_i = x_i^{\text{lg}} - x_i = M_i(x_i)v_i(x_i) + \text{div}(M_i(x_i)) + \sqrt{2}M_i^{1/2}(x_i)\epsilon, \tag{9}$$

where the force Net $v(x) = -\nabla V(x)$ and the diffusion Net $M_i(x)$ each consist of three linear layers with activation functions between layers, and $\epsilon \sim \mathcal{N}(0,1)$.

Given the sample triples $\{x_i^{\text{lg}}, x_i^v, x_i^{\text{hal}}\}$, during training we set $x_i^{lg} \sim \mathcal{N}(0,1)$. To enable the dynamical model to learn dynamics where features tend toward a standard normal distribution, we can update the parameters of force Net and diffusion Net by maximizing the likelihood function, or equivalently, by minimizing the following loss:

$$\mathcal{L}_{base} = \frac{1}{2N}\sum_{i=1}^{N}\left[\log M_i^v(x_i^v) + \frac{(r_i^v)^2}{2M_i^v(x_i^v)}\right] + \frac{1}{2K}\sum_{i=1}^{K}\left[\log M_i^{hal}(x_i^{hal}) + \frac{(r_i^{hal})^2}{2M_i^{hal}(x_i^{hal})}\right], \tag{10}$$

where $r_i^v = x_i^{lg} - x_i^v - M_i^v(x_i^v)v_i^v(x_i^v) - \text{div}(M_i^v(x_i^v))$, $r_i^{hal} = x_i^{lg} - x_i^{hal} - M_i^{hal}(x_i^{hal})v_i^{hal}(x_i^{hal}) - \text{div}(M_i^{hal}(x_i^{hal}))$.

To enable In-Mod and Tn-Mod to effectively learn dynamic characteristics, we introduce the Sliced Wasserstein Distance to characterize the dynamic change features of injected noise at the distributional level. Based on this, we construct the dynamic loss function:

$$\mathcal{L}_{\text{Dynamic}} = \frac{1}{L}\sum_{l=1}^{L}\left[\frac{1}{N}\sum_{n=1}^{N}\left(a_{(n),l}^v - b_{(n),l}^v\right)^2 + \frac{1}{K}\sum_{k=1}^{K}\left(a_{(k),l}^{hal} - b_{(k),l}^{hal}\right)^2\right], \tag{11}$$

where $a_{i,l}^v = \theta_l \cdot \left(x_i^{lg,v} - x_i^v\right)$, $b_{i,l}^v = \theta_l \cdot (z_i^v - x_i^v)$, $a_{i,l}^{hal} = \theta_l \cdot \left(x_i^{lg,hal} - x_i^{hal}\right)$, and $b_{i,l}^{hal} = \theta_l \cdot \left(z_i^{hal} - x_i^{hal}\right)$. $\{\theta_l\}_{l=1}^L$ denote L randomly sampled directions from a uniform distribution on the $(d-1)$-sphere $\mathbb{S}^{d-1}$. $a_{(n),l}^v$ and $b_{(n),l}^v$ denote the n-th sorted values of the image branch samples, respectively. Similarly, $a_{(k),l}^{hal}$ and $b_{(k),l}^{hal}$ denote the k-th sorted values of the hallucination text branch samples.

During training, to ensure the model accurately injects noise into redundant rather than critical information, we introduce a supervision term. This term is computed by calculating the cosine similarity between the masked image features, hallucination text features, and ground truth text features, respectively:

$$\mathcal{L}_{sup} = \frac{1}{NM}\sum_{n=1}^{N}\sum_{m=1}^{M}(h_n^v)^\top h_m^{gt} + \frac{1}{KM}\sum_{k=1}^{K}\sum_{m=1}^{M}(h_k^{hal})^\top h_m^{gt}, \tag{12}$$

where $\mathbf{H}_v = [h_1^v, \ldots, h_N^v] \in \mathbb{R}^{N\times d}$, $\mathbf{H}_{hal} = [h_1^{hal}, \ldots, h_K^{hal}] \in \mathbb{R}^{K\times d}$, and $\mathbf{H}_{gt} = [h_1^{gt}, \ldots, h_M^{gt}] \in \mathbb{R}^{M\times d}$ are the features obtained by passing the masked image features, hallucination text features, and ground truth text features, respectively, through the LLM decoder.

Consequently, the overall loss function is:

$$\mathcal{L}_{all} = \beta * \mathcal{L}_{dynamic} + \mathcal{L}_{base} + \mathcal{L}_{sup} + \mathcal{L}_{ce} \tag{13}$$

where β is a hyperparameter and $\mathcal{L}_{ce}$ denotes the standard cross-entropy loss function, commonly employed in training MLLMs.

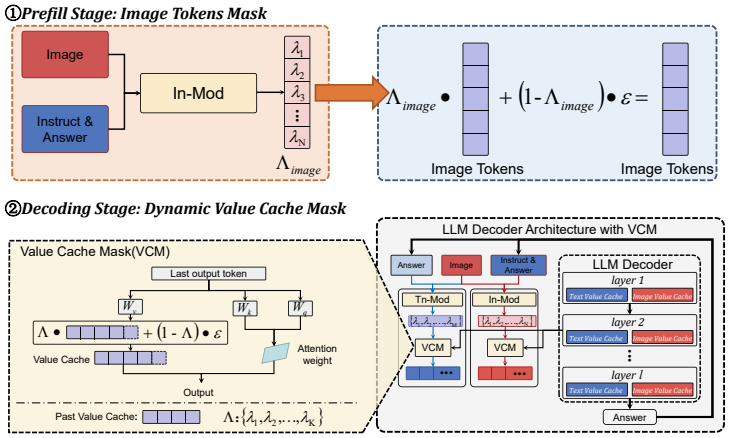


Fig. 4: The illustration of the proposed HalMask inference framework.

4.4 Inference Phase

As illustrated in Figure 4, we introduce customized mask strategies tailored to the two generation stages of MLLMs: (1) during the **prefill stage**, we filter out redundant information from the input image tokens fed into the LLM decoder; (2) during the **decoding stage**, we mask potential redundancy within the past value cache of the LLM decoder.

Prefill Stage. To eliminate redundancy in the image tokens input to the LLM decoder, we first fuse the text tokens $\mathbf{X}_t$ and the image tokens $\mathbf{X}_v$. These fused representations are then processed by the In-Mod module f_{In-Mod} to generate Λ_{image}. Subsequently, a masking operation is applied to the image tokens $\mathbf{X}_v$, resulting in the masked image tokens $\mathbf{Z}_v$. The computation can be formulated as follows:

$$\begin{aligned} \Lambda_{image} &= f_{In-Mod}(\mathbf{X}_t, \mathbf{X}_v), \\ \mathbf{Z}_v &= \Lambda_{image} \cdot \mathbf{X}_v + (1 - \Lambda_{image}) \cdot \epsilon. \end{aligned} \tag{14}$$

Decoding Stage. During the autoregressive generation process, we introduce a **dynamic Value Cache mask mechanism**. Specifically, when generating a new answer token, the model aggregates it with previously generated answer tokens to form $\mathbf{X}_a$, which is then passed through In-Mod and Tn-Mod to obtain Λ_{image} and Λ_{text}. These representations are subsequently fed into each layer of the LLM decoder, providing an updated compact Value Cache representation to facilitate the generation of the next token. This dynamic masking procedure continues throughout the autoregressive generation, enabling the cached information to preserve essential content while effectively suppressing redundancy.

Table 1: CHAIR evaluation results on MSCOCO dataset. Lower $\mathbf{CHAIR}_S$ and $\mathbf{CHAIR}_I$ indicate less object hallucination. Higher BLEU indicates better caption quality. We use 64 as the maximum token length in this experiment.

Methods	LLaVA-1.5			MiniGPT-4		
	$\mathbf{CHAIR}_S\downarrow$	$\mathbf{CHAIR}_I\downarrow$	BLEU↑	$\mathbf{CHAIR}_S\downarrow$	$\mathbf{CHAIR}_I\downarrow$	BLEU↑
Greedy	21.20	6.84	16.25	32.40	12.20	14.57
Beam Search	19.40	6.55	17.24	19.50	6.84	**15.99**
Woodpecker	23.85	7.50	17.05	28.87	10.20	15.30
LURE	19.48	6.50	15.97	27.88	10.20	15.03
OPERA	17.50	6.07	16.02	29.70	11.96	14.82
HALC	16.90	5.72	16.02	25.20	9.42	14.91
HACL	18.27	5.90	17.09	24.47	9.57	15.84
DeGF	18.40	6.10	–	–	–	–
Nullu	15.20	5.30	15.69	21.40	8.99	14.81
HalMask	**7.60**	**3.06**	**19.48**	**16.20**	**5.96**	15.40

The computation of the value cache mask proceeds as follows:

$$\begin{aligned} \Lambda_{image} &= f_{In-Mod}\big(\mathbf{X}_v, \mathrm{Concat}(\mathbf{X}_t, \mathbf{X}_a)\big), \\ \Lambda_{text} &= f_{Tn-Mod}(\mathbf{X}_t, \mathbf{X}_a), \\ \mathbf{C}_v &= \Lambda_{image} \cdot \mathbf{C}_v + (1 - \Lambda_{image}) \cdot \epsilon, \\ \mathbf{C}_t &= \Lambda_{text} \cdot \mathbf{C}_t + (1 - \Lambda_{text}) \cdot \epsilon. \end{aligned} \tag{15}$$

where $\mathbf{C}_v$ and $\mathbf{C}_t$ denote the image value cache and the answer text value cache, respectively. It should be noted that, during the autoregressive generation process, the Value Cache stores unmasked value representations. This design choice stems from the fact that, in the autoregressive decoding phase, each newly generated answer token is associated with a distinct parameter Λ. By avoiding the progressive accumulation of noise from repeated masking, this strategy not only preserves the fidelity of the cached representations but also enables the model to dynamically assign a new Λ to each generated token, thereby ensuring both the stability and expressiveness of the representations throughout the generation process.

5 Experiments

5.1 Experimental Settings

Baselines. Since our method proposes a self-correction approach, we compare it with six existing self-correction methods: Woodpecker [31], LURE [35], OPERA [14], HALC [7], SID [15], and DeGF [32]. We also compare with other state-of-the-art methods, such as HACL [16] and Nullu [30].

Implementation Details. We employ three representative MLLMs: LLaVA-1.5 [20], MiniGPT-4 [36] and Qwen2.5-VL, all at the 7B scale. We conduct experiments on six benchmarks, comprising three standard hallucination benchmarks

Table 2: Experimental results on OPOPE dataset.

Methods	Minigpt-4						LLaVA-1.5					
	Random		Popular		Adversarial		Random		Popular		Adversarial	
	Precision↑	$F_{0.2}$ ↑	Precision↑	$F_{0.2}$ ↑	Precision↑	$F_{0.2}$ ↑	Precision↑	$F_{0.2}$ ↑	Precision↑	$F_{0.2}$ ↑	Precision↑	$F_{0.2}$ ↑
Greedy	98.49	94.25	92.01	88.53	90.65	87.32	98.41	96.42	91.17	89.71	86.36	85.22
Beam	98.70	94.51	93.35	89.78	92.06	88.63	98.67	96.67	91.98	90.47	86.92	85.75
OPERA	98.77	94.55	92.82	89.28	91.63	88.22	98.57	96.58	91.80	90.30	86.40	85.26
HALC	98.62	94.26	93.80	90.01	92.22	88.61	98.21	95.89	91.87	90.05	89.44	87.81
Nullu	99.06	94.82	**96.08**	**92.19**	92.73	89.21	98.05	96.05	94.06	92.36	88.27	86.97
HalMask	**99.46**	**95.62**	95.30	91.91	**93.95**	**90.70**	**99.53**	**96.67**	**95.26**	**92.79**	**92.25**	**90.04**

(CHAIR [26], POPE [18], and OPOPE [7]) and three complex benchmarks (AMBER [27], MMBench [23], and ScienceQA [25]). During training, only the In-Mod and Tn-Mod modules are trainable, while all other parameters remain frozen. Ablation studies on the β hyperparameter in the loss function are presented in Appendix.

5.2 Main Results

Results on CHAIR. CHAIR quantifies the extent of object hallucination (OH) in image descriptions. This metric consists of two indicators: $\mathbf{CHAIR}_S$, which operates at the sentence level, and $\mathbf{CHAIR}_I$, which operates at the instance level. Lower CHAIR scores denote reduced OH. Furthermore, BLEU is employed to evaluate the quality of generated image descriptions, where higher BLEU scores indicate superior quality.

In Table 1, we present the evaluation results on the CHAIR benchmark across two representative MLLMs. Our method consistently outperforms existing approaches on both $\mathbf{CHAIR}_S$ and $\mathbf{CHAIR}_S$ metrics. Specifically, for the LLaVA-1.5 model, the $\mathbf{CHAIR}_S$ score decreases from 21.20 to 7.60, achieving a relative reduction of **64.15%**. Furthermore, our method substantially enhances image captioning quality, with BLEU scores improving from 16.25 to 19.48. These results demonstrate that our approach not only effectively mitigates hallucination in MLLMs but also enables high-quality text generation. As shown in Table 5, we conducted experiments on the Qwen2.5-VL model, and the results demonstrate that hallucination was significantly mitigated.

Results on OPOPE. Following the HALC experimental protocol, we evaluate hallucination using the Offline POPE (OPOPE) benchmark. As shown in Table 2, our method achieves the highest precision and $F_{0.2}$ score across most settings and data splits, demonstrating that our approach accurately describes objects in images while avoiding hallucinated descriptions.

Results on POPE. We conducted experiments on the POPE benchmark using the LLaVA-1.5 model. As shown in Table 3, our method achieves the highest average accuracy and average F_1 score. Specifically, our method improves the average F_1 score by 1.64% over LLaVA-1.5. The experimental results demonstrate the effectiveness of our method in mitigating hallucinations.

Table 3: Experimental results on POPE dataset. All baseline methods are implemented based on the LLaVA-1.57B model.

Methods	Random		Popular		Adversarial		Average Accuracy	Average F_1
	Accuracy↑	F_1 ↑	Accuracy↑	F_1 ↑	Accuracy↑	F_1 ↑		
Greedy	88.16	87.34	87.23	86.11	85.13	84.18	86.84	85.88
HACL	88.59	88.70	87.94	87.36	**86.54**	**85.73**	87.69	87.26
SID	89.46	89.62	85.13	85.94	83.24	82.21	85.94	85.92
DeGF	89.03	88.74	86.63	86.28	81.63	81.94	85.76	85.65
Nullu	89.45	89.20	85.37	85.63	79.40	80.88	84.74	85.24
HalMask	**90.21**	**89.93**	**88.57**	**88.15**	84.37	84.47	**87.72**	**87.52**

Table 4: Experimental results on complex benchmarks. All methods are based on the LLaVA-1.57B model.

Methods	AMBER				MMBench↑	SQA-IMG↑
	CHAIR↓	Cover↑	Hal↓	Cog↓		
LLaVA-1.5	7.3	**50.9**	34.1	4.2	64.3	66.2
HalMask	**2.9**	50.1	**22.5**	**1.5**	**65.2**	**68.5**

Table 5: CHAIR evaluation results on MSCOCO dataset. We use 512 as the maximum token length in this experiment.

Methods	CHAIR_S↓	CHAIR_I↓
Qwen2.5-VL	36.1	9.4
HalMask	**23.7**	**6.5**

Results on AMBER. As shown in Table 4, our method significantly mitigates hallucinations while maintaining coverage of ground-truth objects. Specifically, for LLaVA-1.5, our approach substantially reduces CHAIR, Hal, and Cog scores (where lower values indicate fewer hallucinations), while maintaining the Cover metric (the proportion of correctly mentioned ground-truth objects).

Results on MMBench. MMBench systematically evaluates twenty capability dimensions of MLLMs. We present the evaluation results using LLaVA-1.5 as the baseline in Table 4, where the overall scores increase from 64.3 to 65.2, demonstrating that our method enhances the multimodal capabilities of MLLMs.

Results on ScienceQA. ScienceQA is a dataset designed for complex scientific reasoning, comprising question-answer style multiple-choice questions derived from elementary and middle school science curricula. As shown in Table 4, our method improves LLaVA-1.5's accuracy by 2.3% compared to the baseline.

5.3 Ablation Study

To better understand the contribution of each component, we conduct comprehensive ablation studies on the POPE, OPOPE, and CHAIR benchmarks. As summarized in Tables 6 and 7, we evaluate the impact of the following modules: the Image Noise Modulation Module (In-Mod), the Text Noise Modulation Module (Tn-Mod), and the Dynamical Regularization Term.

Table 6: Ablation results on the CHAIR and OPOPE benchmarks.

Methods	CHAIR		OPOPE			
	$\mathbf{CHAIR}_s\downarrow$	$\mathbf{CHAIR}_I\downarrow$	Random↑	Popular↑	Adversarial↑	Average $F_{0.2}\uparrow$
LLaVA-1.5	21.20	6.84	96.42	89.71	85.22	90.45
+In-Mod	12.80	4.45	96.52	91.98	89.69	92.73
+Tn-Mod	20.40	6.74	**96.79**	89.31	87.30	91.13
+Tn-Mod+In-Mod	**7.60**	**3.06**	96.67	**92.79**	90.04	**93.17**
MiniGPT-4	32.40	12.20	94.25	88.53	87.32	90.03
+In-Mod	20.80	7.50	95.06	89.54	89.11	91.24
+Tn-Mod	23.00	8.13	**95.71**	88.84	87.32	90.62
+Tn-Mod+In-Mod	**16.20**	**5.96**	95.62	**91.91**	**90.70**	**92.74**

Effects of In-Mod and Tn-Mod. As shown in Table 6, the introduction of both In-Mod and Tn-Mod improves the performance of LLaVA-1.5 and MiniGPT-4 on CHAIR and OPOPE. Notably, in the MiniGPT-4 model, In-Mod reduces $\mathbf{CHAIR}_S$ by **11.6**, and Tn-Mod reduces $\mathbf{CHAIR}_S$ by **9.4**. This indicates that both In-Mod and Tn-Mod can significantly reduce hallucinations in MLLMs.

Table 7: Ablation results on POPE dataset.

Methods	POPE-Random		POPE-Popular		POPE-Adversarial		Average Recall↑	Average F1 Score↑	RTD↓
	Recall↑	$F_1\uparrow$	Recall↑	$F_1\uparrow$	Recall↑	$F_1\uparrow$			
w/o. Dynamical Regularization Term	82.67	88.86	83.00	87.95	82.60	**84.51**	82.76	87.11	56.13
HalMask	**84.87**	**89.93**	**85.07**	**88.15**	**85.00**	84.47	**84.98**	**87.52**	**50.15**

Effects of Dynamical Regularization Term. Dynamical Regularization Term prevents valuable information from being masked by Gaussian noise. As shown in Table 7, incorporating the Dynamical Regularization Term improves LLaVA-1.5's performance on the average F_1 score. Notably, the average recall score increases by 2.22%, indicating that informative features are effectively preserved. Additionally, we employ the Representation Topology Divergence (RTD) metric [4] and observe a reduction of 5.98 when using the Dynamical Regularization Term, which corroborates that this regularization mechanism helps preserve meaningful topological structure.

Speed-Accuracy Analysis. We conducted comparative experiments using the LLaVA-v1.5-7B backbone and evaluated performance on the POPE benchmark using the POPE average F_1 score as the metric. By eliminating additional decoding stages, our method achieves a superior trade-off between performance and efficiency. As shown in Table 8, compared to the typical contrastive decoding method HALC, our approach is 8.3 times faster while improving the POPE average F_1 score by 10.8%. This demonstrates that HalMask achieves substantial performance gains with only marginal and acceptable computational cost.

Table 8: Comparison of inference time and POPE benchmark performance across different methods.

	LLaVA-1.5	HALC	OPERA	SID	HalMask
Time (s/token)	0.021	0.283	0.136	0.137	0.034
POPE-Average↑	85.877	76.727	85.384	85.923	**87.517**

6 Conclusion

We proposed HalMask, a lightweight and efficient framework that mitigates hallucinations by selectively masking redundant features via Gaussian noise injection. Our approach operated across two critical phases of inference: at the prefill stage, we injected Gaussian noise into the hallucination-prone features within the image tokens; at the decoding stage, a dynamic Value Cache masking mechanism derived adaptive coefficients to inject noise into both image and text caches. Furthermore, our Dynamical Regularization Term explicitly constrained the modulation process to preserve essential semantic cues and maintain latent topological structure. Extensive experiments across diverse benchmarks demonstrated that HalMask significantly mitigated hallucinations with only marginal and acceptable computational overhead.

Acknowledgments

This work is supported by National Natural Science Foundation of China(U22B2017), and International Cooperation Foundation of Hubei Province, China (2024EHA032).

References

1. Achille, A., Soatto, S.: Information dropout: Learning optimal representations through noisy computation. IEEE transactions on pattern analysis and machine intelligence **40**(12), 2897–2905 (2018)
2. Alemi, A.A., Fischer, I., Dillon, J.V., Murphy, K.: Deep variational information bottleneck. arXiv preprint arXiv:1612.00410 (2016)
3. Bai, J., Guo, H., Peng, Z., Yang, J., Li, Z., Li, M., Tian, Z.: Mitigating hallucinations in large vision-language models by adaptively constraining information flow. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 23442–23450 (2025)
4. Barannikov, S., Trofimov, I., Balabin, N., Burnaev, E.: Representation topology divergence: A method for comparing neural network representations. Proceedings of Machine Learning Research **162**, 1607–1626 (2022)
5. Biten, A.F., Gómez, L., Karatzas, D.: Let there be a clock on the beach: Reducing object hallucination in image captioning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1381–1390 (2022)
6. Che, L., Liu, T.Q., Jia, J., Qin, W., Tang, R., Pavlovic, V.: Hallucinatory image tokens: A training-free eazy approach to detecting and mitigating object hallucinations in lvlms. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 21635–21644 (2025)

7. Chen, Z., Zhao, Z., Luo, H., Yao, H., Li, B., Zhou, J.: Halc: Object hallucination reduction via adaptive focal-contrast decoding. In: International Conference on Machine Learning. pp. 7824–7846. PMLR (2024)
8. Dai, W., Liu, Z., Ji, Z., Su, D., Fung, P.: Plausible may not be faithful: Probing object hallucination in vision-language pre-training. arXiv preprint arXiv:2210.07688 (2022)
9. Fathi, M., Stoltz, G.: Improving dynamical properties of metropolized discretizations of overdamped langevin dynamics. Numerische Mathematik **136**(2), 545–602 (2017)
10. Guan, T., Liu, F., Wu, X., Xian, R., Li, Z., Liu, X., Wang, X., Chen, L., Huang, F., Yacoob, Y., et al.: Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14375–14385 (2024)
11. Gunjal, A., Yin, J., Bas, E.: Detecting and preventing hallucinations in large vision language models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 18135–18143 (2024)
12. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-VAE: Learning basic visual concepts with a constrained variational framework. In: International Conference on Learning Representations (2017), `https://openreview.net/forum?id=Sy2fzU9gl`
13. Hoffman, M.D., Johnson, M.J.: Elbo surgery: yet another way to carve up the variational evidence lower bound. In: Workshop in advances in approximate Bayesian inference, NIPS. vol. 1 (2016)
14. Huang, Q., Dong, X., Zhang, P., Wang, B., He, C., Wang, J., Lin, D., Zhang, W., Yu, N.: Opera: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13418–13427 (2024)
15. Huo, F., Xu, W., Zhang, Z., Wang, H., Chen, Z., Zhao, P.: Self-introspective decoding: Alleviating hallucinations for large vision-language models. arXiv preprint arXiv:2408.02032 (2024)
16. Jiang, C., Xu, H., Dong, M., Chen, J., Ye, W., Yan, M., Ye, Q., Zhang, J., Huang, F., Zhang, S.: Hallucination augmented contrastive learning for multimodal large language model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 27036–27046 (2024)
17. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
18. Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W.X., Wen, J.R.: Evaluating object hallucination in large vision-language models. arXiv preprint arXiv:2305.10355 (2023)
19. Liu, F., Lin, K., Li, L., Wang, J., Yacoob, Y., Wang, L.: Mitigating hallucination in large multi-modal models via robust instruction tuning. arXiv preprint arXiv:2306.14565 (2023)
20. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 26296–26306 (2024)
21. Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., Lee, Y.J.: Llavanext: Improved reasoning, ocr, and world knowledge (2024)
22. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36**, 34892–34916 (2023)

23. Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., et al.: Mmbench: Is your multi-modal model an all-around player? In: European conference on computer vision. pp. 216–233. Springer (2024)
24. Lovenia, H., Dai, W., Cahyawijaya, S., Ji, Z., Fung, P.: Negative object presence evaluation (nope) to measure object hallucination in vision-language models. arXiv preprint arXiv:2310.05338 (2023)
25. Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.W., Zhu, S.C., Tafjord, O., Clark, P., Kalyan, A.: Learn to explain: Multimodal reasoning via thought chains for science question answering. Advances in neural information processing systems **35**, 2507–2521 (2022)
26. Rohrbach, A., Hendricks, L.A., Burns, K., Darrell, T., Saenko, K.: Object hallucination in image captioning. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 4035–4045 (2018)
27. Wang, J., Wang, Y., Xu, G., Zhang, J., Gu, Y., Jia, H., Wang, J., Xu, H., Yan, M., Zhang, J., et al.: Amber: An llm-free multi-dimensional benchmark for mllms hallucination evaluation. arXiv preprint arXiv:2311.07397 (2023)
28. Wang, Y., Rudner, T.G., Wilson, A.G.: Visual explanations of image-text representations via multi-modal information bottleneck attribution. Advances in Neural Information Processing Systems **36**, 16009–16027 (2023)
29. Wu, J., Liu, Q., Wang, D., Zhang, J., Wu, S., Wang, L., Tan, T.: Logical closed loop: Uncovering object hallucinations in large vision-language models. In: Findings of the Association for Computational Linguistics: ACL 2024. pp. 6944–6962 (2024)
30. Yang, L., Zheng, Z., Chen, B., Zhao, Z., Lin, C., Shen, C.: Nullu: Mitigating object hallucinations in large vision-language models via halluspace projection. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 14635–14645 (2025)
31. Yin, S., Fu, C., Zhao, S., Xu, T., Wang, H., Sui, D., Shen, Y., Li, K., Sun, X., Chen, E.: Woodpecker: Hallucination correction for multimodal large language models. Science China Information Sciences **67**(12), 220105 (2024)
32. Zhang, C., Wan, Z., Kan, Z., Ma, M.Q., Stepputtis, S., Ramanan, D., Salakhutdinov, R., Morency, L.P., Sycara, K., Xie, Y.: Self-correcting decoding with generative feedback for mitigating hallucinations in large vision-language models. arXiv preprint arXiv:2502.06130 (2025)
33. Zhang, Z., Zhang, A., Li, M., Zhao, H., Karypis, G., Smola, A.: Multimodal chain-of-thought reasoning in language models. arXiv preprint arXiv:2302.00923 (2023)
34. Zhao, S., Song, J., Ermon, S.: Infovae: Information maximizing variational autoencoders. arXiv preprint arXiv:1706.02262 (2017)
35. Zhou, Y., Cui, C., Yoon, J., Zhang, L., Deng, Z., Finn, C., Bansal, M., Yao, H.: Analyzing and mitigating object hallucination in large vision-language models. arXiv preprint arXiv:2310.00754 (2023)
36. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: Enhancing vision-language understanding with advanced large language models. arXiv preprint arXiv:2304.10592 (2023)
37. Zhuang, X., Zhu, Z., Xie, Y., Liang, L., Zou, Y.: Vasparse: Towards efficient visual hallucination mitigation via visual-aware token sparsification. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 4189–4199 (2025)