SENSIEDIT: SENSITIVITY-GUIDED SPATIAL ATTENTION EDITING FOR HALLUCINATION MITIGATION IN LVLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large vision-language models (LVLMs) have demonstrated remarkable capabilities in multimodal reasoning and complex scene understanding. However, they still suffer from hallucination, generating descriptions that contradict visual facts. Attention heads differ in their sensitivity to image content, motivating selective modulation of visual attention. To exploit this variation, we propose **SensiEdit**, a lightweight framework for sensitivity-guided spatial attention editing. Specifically, we introduce *Attention-based Visual Sensitivity* (AVS) to characterize each head's response to changes in visual input and train a lightweight predictor to estimate this signal. The predicted sensitivity guides stronger visual attention modulation for less sensitive heads, while learnable spatial biases determine the editing strength at different visual positions. The LVLM backbone remains frozen throughout training, and the spatial biases are learned from only 3K images. Experiments on LLaVA-1.5-7B and Qwen2.5-VL-7B, together with supplementary results on three additional backbones, demonstrate the effectiveness of SensiEdit in mitigating object hallucination. On LLaVA-1.5-7B, SensiEdit achieves relative improvements of 56% and 68% on CHAIR_S and CHAIR_I, respectively, against the baseline in Table 7, while preserving general capabilities.

1 INTRODUCTION

Large Vision-Language Models (LVLMs) (Bai et al., 2023b; Dai et al., 2023; Liu et al., 2024; Zhu et al., 2024) generate fluent, context-aware language grounded in visual input, yet they hallucinate: the generated description asserts objects, attributes or relations that the image does not contain. Such failures matter in applications that require reliable visual descriptions, including medical reporting (Tanno et al., 2025).

Existing mitigation strategies fall into three families, each with a structural limitation. *Retraining* methods (Jiang et al., 2024; Fu et al., 2025; Zhao et al., 2023; Lu et al., 2023) construct hallucination-specific preference data and update the LVLM, but they require expensive data curation and gradient updates through a multi-billion parameter model. *Contrastive decoding* (CD) methods (Leng et al., 2024; Huo et al., 2025; Wang et al., 2024; Favero et al., 2024) contrast the output distribution of the original input against a deliberately weakened one; they avoid retraining, but double the forward passes and rescale the probability of every token indiscriminately. *Representation editing* methods (Jiang et al., 2025; Yang et al., 2025; Liu et al., 2025; Zhang et al., 2024) steer hidden states along a learned direction; they are cheap at inference, but they operate on the *output* of attention—correcting the symptom after the visual evidence has already been discarded.

The left panel of Fig. 1 illustrates head-wise differences in visual sensitivity through an image-replacement experiment. Under teacher-forced decoding, we keep the textual prefix unchanged and replace only the input image, then compare the visual attention maps of individual heads at the same layer and text position. This setup examines how strongly each head's attention responds to changes in visual input. As illustrated, the attention distribution of Head i changes substantially after image replacement, whereas that of Head j remains relatively similar. This contrast suggests that heads within the same layer can differ in their dependence on visual content. We quantify this response

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

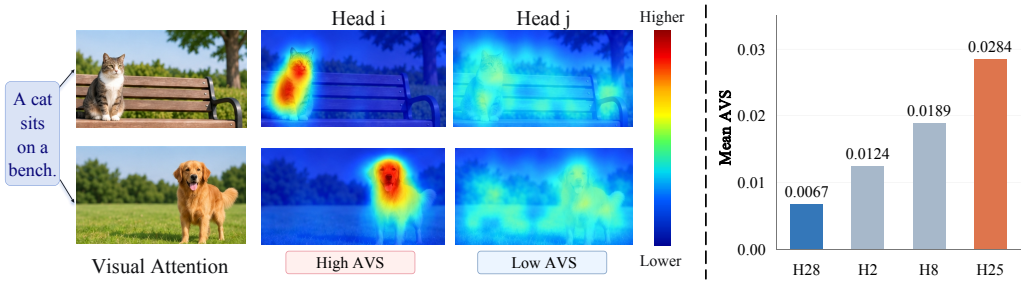

Figure 1: Attention-based Visual Sensitivity (AVS) varies across heads. **Left:** Under teacher-forced decoding with the same text, two heads at the same layer and text position exhibit different responses when the input image is replaced: Head i shifts its visual attention distribution substantially (high AVS), while Head j barely changes (low AVS). **Right:** Mean AVS of four representative heads, averaged over multiple image-replacement samples. The variation indicates that visual sensitivity is a per-head property, motivating head-selective rather than uniform attention editing.

using *Attention-based Visual Sensitivity* (AVS), a per-head measure of the change in visual attention under image replacement.

The right panel of Fig. 1 summarises this difference by reporting the mean AVS of four representative heads over multiple image-replacement samples. The differences among these heads suggest that visual sensitivity should be considered at the head level rather than treated as uniform across the decoder. This observation motivates testing whether visually insensitive heads benefit from stronger attention correction, while already-sensitive heads require less intervention. Building on this hypothesis, we propose **SensiEdit** (**Sensitivity-Guided Spatial Attention Editing**), a framework comprising two components: a lightweight *Visual Sensitivity Predictor* and a learnable *Spatial Attention Bias Editor*. The MLP predictor estimates head-wise AVS at each decoding step, enabling sensitivity-guided editing without repeated image-replacement passes at inference. The editor learns a separate bias strength for each layer, head, and visual position, and adds the resulting bias to visual-key attention logits before the softmax. The two components work together: predicted sensitivity modulates the learned spatial strengths, assigning stronger modulation to lower-sensitivity heads, while a gate produced by the predictor controls whether editing is applied. After training the predictor, we freeze it together with the LVLM and learn only the spatial bias parameters (0.09–0.15M) from 3K images.

Our highlights are summarized as follows:

- We identify differences in visual sensitivity across attention heads and introduce *Attention-based Visual Sensitivity* (AVS), which measures head-wise attention responses to image replacement under identical textual prefixes, providing a diagnostic for sensitivity-guided attention editing.
- We propose SensiEdit, a framework combining a lightweight MLP sensitivity predictor and a learnable spatial attention bias editor. Predicted sensitivity guides head-wise modulation, while spatial biases set editing strength per visual position, with the LVLM backbone frozen.
- We validate SensiEdit across multiple LVLM backbones without updating their parameters and observe consistent reductions in object hallucination.

2 Related Work

2.1 Large Vision-Language Models

Building on the success of Large Language Models (LLMs) (Bai et al., 2023a; Brown et al., 2020; Chiang et al., 2023; Touvron et al., 2023) and cross-modal learning (Radford et al., 2021; Dosovitskiy, 2020), Large Vision-Language Models (LVLMs) achieve breakthrough performance by integrating visual perception and language generation capabilities, excelling in tasks such as image captioning (Hossain et al., 2019), visual question answering (Antol et al., 2015), and multimodal reasoning (Lu et al., 2022). A typical LVLM architecture comprises three core components: a visual encoder for hierarchical image feature extraction like CLIP (Radford et al., 2021); a cross-modal alignment

module implemented via linear projection layers or advanced architectures like Q-Former (Li et al., 2023b); and a large language model fine-tuned through instruction tuning for context-aware text generation. Despite these advancements, LVLMs still suffer from hallucination, where the generated text contains content inconsistent with the input visual information.

2.2 Mitigating Hallucinations in LVLMs

Retraining methods construct hallucination-specific datasets and update the LVLM. HA-DPO (Zhao et al., 2023) fine-tunes with style-consistent hallucination sample pairs, while HDPO (Fu et al., 2025) targets diverse hallucination causes through specialized preference pairs. These methods require substantial data curation and computational resources for model fine-tuning, limiting practical applicability. In contrast, contrastive decoding (CD) methods mitigate hallucinations at inference by contrasting output distributions from original and perturbed inputs. VCD (Leng et al., 2024) introduces noise into visual inputs and contrasts the resulting token distributions, while SID (Huo et al., 2025) retains only the least important visual tokens to amplify hallucinations for contrastive correction. Though effective, CD methods double the forward passes and adjust all token probabilities indiscriminately.

A third line of work edits internal representations directly. ProjectAway (Jiang et al., 2025) removes hallucination components from image features, Nullu (Yang et al., 2025) projects weights onto the null space of hallucination subspaces, and VTI (Liu et al., 2025) steers latent representations at inference. These methods are lightweight but operate on the output of attention, after the visual evidence has already been aggregated. Meanwhile, attention-focused studies (Xiao et al., 2024; Huang et al., 2024; Chen et al., 2024; An et al., 2025) reveal that visual tokens receive disproportionately low attention while sink positions absorb the bulk, yet do not leverage this imbalance for targeted editing. Unlike all of the above, SensiEdit intervenes at the attention logits themselves—the earliest point deciding how much visual evidence enters a decode step—without modifying weights or requiring a second forward pass.

3 Method

As illustrated in Fig. 2, SensiEdit (Sensitivity-Guided Spatial Attention Editing) consists of two components that redistribute attention mass from semantically empty sink positions back to visual tokens during autoregressive decoding. A lightweight *Visual Sensitivity Predictor* (Sec. 3.2) estimates, at each decode step, how strongly each attention head depends on the image, producing per-head sensitivity scores and a thresholded soft gate. A *Spatial Attention Bias Editor* (Sec. 3.3) then injects a per-(layer, head, position) bias into the pre-softmax logits of visual keys, modulated by these predictions. The predictor is trained once from attention statistics and frozen; only the spatial bias table is learned via self-distillation on 3000 captioning examples (Sec. 3.4).

3.1 Target Objective: Decode-Step Attention Reallocation

Let $v = (v_1, \ldots, v_{N_v})$ be the visual tokens and $\mathcal{V} \subset \{1, \ldots, T\}$ their positions in the cached key sequence. At layer l and head h of a decode step, the attention over all T keys is

$$\boldsymbol{a}_{l,h} \;=\; \mathrm{softmax}\!\left(\frac{\boldsymbol{q}_{l,h}\boldsymbol{K}_{l,h}^{\top}}{\sqrt{d}} + \boldsymbol{m}\right) \in \mathbb{R}^{1\times T}. \tag{1}$$

We define

$$M_{l,h}^{\text{vis}} = \sum_{j\in\mathcal{V}} a_{l,h,j}, \qquad M_{l,h}^{\text{sink}} = a_{l,h,0}, \tag{2}$$

where M^{vis} is the attention mass spent on the image and M^{sink} is the mass on position 0—a non-visual prefix token in the evaluated prompts.

Table 1 quantifies the collapse: in LLaVA-1.5-7B layer 4, all 576 visual tokens collectively receive only $M^{\text{vis}} = 0.076$, while the non-visual position 0 alone absorbs $M^{\text{sink}} = 0.818$, with the remaining mass going to text tokens; Qwen2.5-VL-7B assigns $M^{\text{vis}} = 0.042$ to visual tokens and $M^{\text{sink}} = 0.935$ to position 0. Visual evidence is present in the KV cache but receives little attention in these layers.

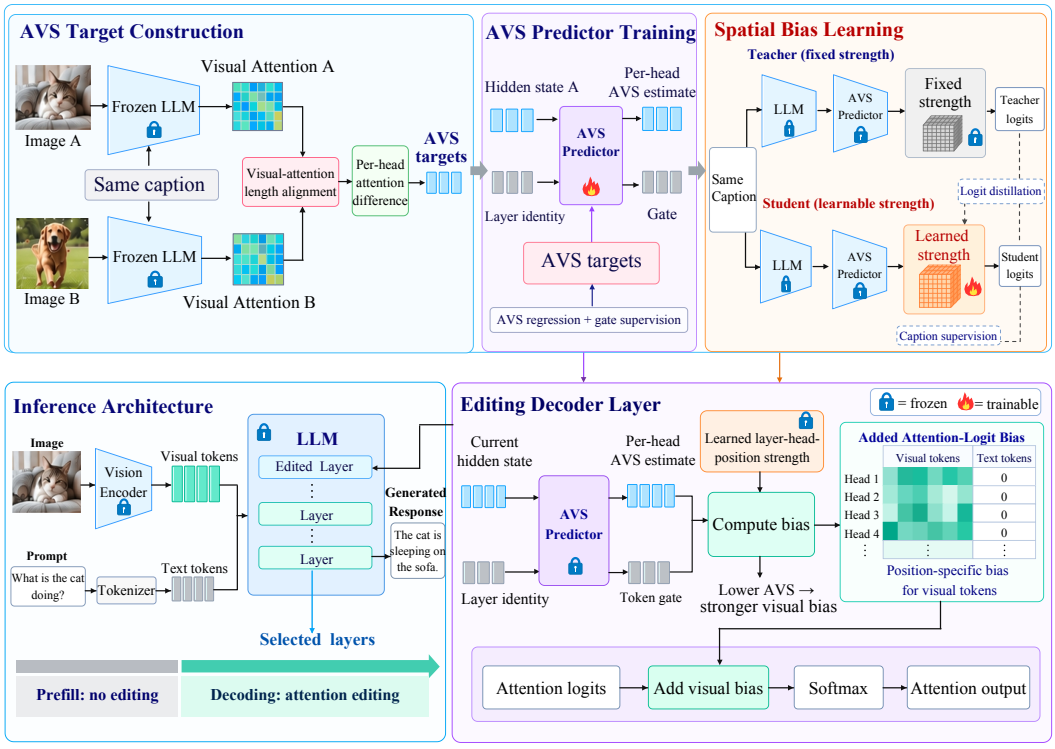

Figure 2: Overview of SensiEdit. **Top (training):** Visual-attention differences between paired images provide AVS targets for the Predictor, which is then frozen. A fixed-strength teacher and a learnable spatial-bias student train the Bias Editor through logit distillation and caption supervision. **Bottom (inference):** At decode steps, the frozen Predictor estimates per-head visual sensitivity and a step gate; the Bias Editor adds a gated, position-specific bias to visual-key logits before the softmax.

Table 1: Attention allocation on COCO validation images, averaged over attention heads at the indicated layer. M^{vis} sums attention over all visual tokens; M^{sink} is attention to position 0. Baseline and edited denote attention without and with visual-logit bias.

Model	**Visual interface**	**Layer**	M^{vis} **(baseline)**	M^{sink} **(baseline)**	M^{vis} **(edited)**
LLaVA-1.5-7B	MLP projection, 576 tokens	4	0.076	0.818	**0.854**
Qwen2.5-VL-7B	MLP projection, dynamic	3	0.042	0.935	**0.962**

This reframes hallucination mitigation as a *reallocation* problem: adding $b > 0$ to every visual logit multiplies the visual/non-visual odds by e^b, drawing mass out of the sink. The last column of Table 1 confirms the effect.

However, removing the step gate increases LLaVA's mean caption length from 75.0 to 120.1 words in the always-on ablation (Table 7). The target is therefore a bias $b_{l,h,v}$ that varies along three dimensions: *which heads* to push, *how strongly at each position*, and *which steps* to edit.

3.2 Attention-based Visual Sensitivity

To identify heads that would benefit from a push, we measure visual sensitivity by holding the textual prefix constant and varying only the image. For a fixed caption, we run teacher-forced decoding with images A and B, obtaining visual attention vectors $\boldsymbol{a}^{(A)}_{l,h,t}$ and $\boldsymbol{a}^{(B)}_{l,h,t}$. If the two images produce different visual-token lengths, their visual-attention vectors are aligned to the same length before comparison, as shown in Figure 2. The *Attention-based Visual Sensitivity* (AVS) is the half-ℓ_1 distance

over the aligned visual positions:

$$\mathrm{AVS}_{l,h,t} \;=\; \frac{1}{2}\sum_{v\in\mathcal{V}}\big|a^{(A)}_{l,h,t,v} - a^{(B)}_{l,h,t,v}\big|. \tag{3}$$

High AVS indicates a stronger attention response to image replacement; low AVS indicates a weaker response (Appendix A.2).

Visual Sensitivity Predictor. Computing AVS at inference would require paired forward passes. We instead train a lightweight MLP to regress AVS from a single hidden state. It takes the pre-layer-norm hidden state $\boldsymbol{h}_l \in \mathbb{R}^{d_{\text{model}}}$ concatenated with a 64-dimensional learned layer embedding and outputs H per-head AVS logits plus one gate logit:

$$\big[\hat{\boldsymbol{s}}_l,\, u_l\big] \;=\; \mathrm{MLP}_\phi\big([\,\boldsymbol{h}_l\,;\,\mathrm{emb}(l)\,]\big), \qquad \hat{s}_{l,h} = \mathrm{MinMaxNorm}\big(\sigma(\hat{\boldsymbol{s}}_l)\big)_h. \tag{4}$$

The gate is $g_l = \sigma(u_l)\mathbb{I}[\sigma(u_l) \geq 0.5]$, retaining its sigmoid weight when active. MLP_ϕ is trained once on AVS targets from Eq. (3) and then **frozen**.

3.3 Spatial Attention Bias Editor

Let $\mathcal{L} = \{l_1, \ldots, l_8\}$ denote the edited decoder layers. At a decode step, for layer $l \in \mathcal{L}$, head h, and visual position $v \in \mathcal{V}$, SensiEdit adds

$$b_{l,h,v} \;=\; g_l \cdot \alpha_{l,h,v} \cdot \big(1 - \hat{s}_{l,h}\big) \tag{5}$$

to the pre-softmax logit, leaving non-visual logits untouched. The edited attention becomes

$$\tilde{\boldsymbol{a}}_{l,h} \;=\; \mathrm{softmax}\left(\frac{\boldsymbol{q}_{l,h}\boldsymbol{K}_{l,h}^\top}{\sqrt{d}} + \boldsymbol{m} + \boldsymbol{b}_{l,h}\right), \qquad b_{l,h,j} = 0 \;\; \forall j \notin \mathcal{V}. \tag{6}$$

The three factors each address one dimension of modulation:

(i) Spatial strength $\alpha_{l,h,v}$. A learnable table of shape $|\mathcal{L}| \times H \times N_v$: 147456 parameters for LLaVA. *Fixed* mode indexes the static grid directly; *grid_interp* mode (Qwen2.5-VL, dynamic resolution) stores α on a 20×20 reference grid and bilinearly resamples per forward pass. Spatial variation reduces CHAIR_S from 30.8 to 22.6 over a per-head scalar (Table 7).

(ii) Visual insensitivity weight $1 - \hat{s}_{l,h}$. The bias on heads with high predicted sensitivity (high $\hat{s}$) is damped toward zero; insensitive heads (low $\hat{s}$) receive nearly the full strength α.

(iii) Step gate g_l. A thresholded soft gate from the predictor; when $g_l = 0$ the layer runs unmodified. The gate fires on $\sim$70% of layer–decode-step events (Table 7), preventing the degenerate over-generation observed without it (Sec. 4.4).

Inference. Bias is injected at autoregressive decode steps ($q_{\text{len}} = 1$). Its effect on general capabilities is assessed empirically (Table 5). The per-model instantiation is in Appendix B (Table 8).

3.4 Training

Training has two stages, each on 3000 COCO train2014 images with the LVLM frozen.

Stage 1: Predictor. For each image pair (A, B), we generate a caption from A, teacher-force it on both images, and extract per-head AVS via Eq. (3). MLP_ϕ is trained with MSE on AVS targets and binary cross-entropy on a gate label indicating low mean AVS, then frozen.

Stage 2: Spatial Bias Table. We train α by self-distillation: a *teacher* pass uses the frozen LVLM with uniform bias α_{init}, a *student* pass uses the current spatial α. The objective is

$$\mathcal{L} \;=\; w_{\text{KD}}\,\mathrm{KL}\big(p_{\text{T}}^\tau \,\|\, p_{\text{S}}^\tau\big) \;+\; w_{\text{CE}}\,\mathrm{CE}\big(p_{\text{S}},\, y^{\text{GT}}\big) \;+\; w_\alpha\,\big\|\alpha - \alpha_{\text{init}}\big\|_2^2, \tag{7}$$

with $w_{\text{KD}} = 1.0$, $w_{\text{CE}} = 0.2$, $w_\alpha = 10^{-4}$, $\tau = 2.0$. The KL term transfers the teacher's grounding behaviour while letting the student redistribute the push spatially; the CE term keeps output fluent; the ℓ_2 term prevents α from diverging. α is initialised to $\alpha_{\text{init}} = 15.0$ to match the teacher strengths at step 0; training then adapts both their magnitude and spatial distribution.

Table 2: CHAIR results (max_new_tokens=512).

Method	**LLaVA-1.5-7B** (Liu et al., 2024)			**Qwen2.5-VL-7B** (Bai et al., 2025)		
	$CHAIR_S \downarrow$	$CHAIR_I \downarrow$	TFLOPs↓	$CHAIR_S \downarrow$	$CHAIR_I \downarrow$	TFLOPs↓
Prior work						
Baseline	51.30	16.80	10.23	46.20	9.60	–
ICD (Wang et al., 2024)	47.40	13.90	20.63	–	–	–
VCD (Leng et al., 2024)	46.80	13.20	20.46	–	–	–
M3ID (Favero et al., 2024)	48.30	13.50	20.71	–	–	–
AVISC (Woo et al., 2025)	45.20	13.40	20.08	–	–	–
Octopus (Suo et al., 2025)	39.20	11.10	21.39	–	–	–
ProjectAway (Jiang et al., 2025)	42.00	12.20	–	–	–	–
VTI (Liu et al., 2025)	35.80	11.10	–	–	–	–
DMAS (Yin et al., 2026)	30.80	11.40	–	43.20	9.50	–
ILVAD (Xie et al., 2026)	32.60	9.42	–	42.90	9.10	–
RUDDER (Zou et al., 2025)	39.50	10.50	–	38.50	8.60	–
HIRE (Suo et al., 2026)	30.20	9.70	11.81	36.60	8.00	–
Ours						
+SensiEdit (uniform $\alpha = 15$)	28.20	8.06	9.82	37.20	10.38	9.89
+SensiEdit (spatial α)	**22.60**	**5.32**	**9.80**	**36.00**	**7.51**	**9.28**

4 Experiments

4.1 Experimental Settings

Models. We evaluate on LLaVA-1.5-7B (Liu et al., 2024) as the primary model and Qwen2.5-VL-7B (Bai et al., 2025) as a cross-architecture generalisation target. Appendix C.3 reports additional results on LLaVA-NeXT-7B, Qwen3-VL-8B, and InternVL3-8B.

Benchmarks. CHAIR (Rohrbach et al., 2018) measures object hallucination in image captions at sentence level ($CHAIR_S$) and instance level ($CHAIR_I$); lower is better. We evaluate on 500 randomly sampled MSCOCO (Lin et al., 2014) images with max_new_tokens=512. POPE (Li et al., 2023c) assesses discriminative object hallucination over 8910 binary questions under random/popular/adversarial sampling. AMBER (Wang et al., 2023) combines 1004 generative and 14216 discriminative instances. We additionally report MME (Fu et al., 2026), SEED-Bench (Li et al., 2023a), and MMMU (Yue et al., 2024) as general-capability probes.

Reporting protocol. Baseline and edited arms use identical image IDs, prompts, decoding parameters, and seeds. For every edited arm, we log the number of attention-bias injections actually applied, so that a null result cannot be confused with a hook that silently failed to fire.

Implementation. We edit 8 decoder layers per model; greedy decoding is used throughout. Full hyperparameters are in Appendix B.

4.2 Main Results

We compare SensiEdit against contrastive-decoding methods (ICD (Wang et al., 2024), VCD (Leng et al., 2024), M3ID (Favero et al., 2024), AVISC (Woo et al., 2025), Octopus (Suo et al., 2025)), representation-editing methods (ProjectAway (Jiang et al., 2025), VTI (Liu et al., 2025)), and recent hallucination mitigation methods (DMAS (Yin et al., 2026), ILVAD (Xie et al., 2026), RUDDER (Zou et al., 2025), HIRE (Suo et al., 2026)).

Table 2 reports captioning results. SensiEdit lowers $CHAIR_S/CHAIR_I$ from 51.30/16.80 to 22.60/5.32 on LLaVA and from 46.20/9.60 to 36.00/7.51 on Qwen. On LLaVA, these scores are below the strongest prior $CHAIR_S$ and $CHAIR_I$ entries (HIRE's 30.20 and ILVAD's 9.42); on Qwen, they are below HIRE's 36.60/8.00.

Table 3: POPE results on MSCOCO. Accuracy and F1 are reported for each sampling strategy; ALL denotes their mean across the three splits.

Method	Random		Popular		Adversarial		ALL		TFLOPs↓
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	
LLaVA-1.5-7B (Liu et al., 2024)									
Baseline	83.77	81.94	82.57	80.86	79.77	78.47	82.04	80.42	8.13
+VTI	89.50	88.89	87.36	86.69	82.57	82.11	86.48	85.90	–
+ICD	87.51	83.28	83.15	83.91	79.13	80.41	83.26	82.53	–
+VCD	85.43	83.99	83.17	81.94	80.27	79.49	82.96	81.81	16.26
+M3ID	86.13	81.85	82.07	80.77	79.50	78.15	82.57	80.26	16.26
+AVISC	84.67	82.21	83.67	81.27	81.83	79.55	83.39	81.01	16.26
+Octopus	87.51	85.40	85.20	84.19	82.22	81.44	85.79	83.44	16.34
+HIRE	**90.37**	**90.25**	87.70	87.86	83.73	83.56	87.27	87.23	10.62
+SensiEdit	89.93	89.94	**87.83**	**87.98**	**84.33**	**84.62**	**87.36**	**87.51**	**8.14**
Qwen2.5-VL-7B (Bai et al., 2025)									
Baseline	88.99	87.65	85.87	84.08	84.43	83.67	86.43	85.13	5.54
+AVISC	92.36	91.50	90.76	**90.16**	87.62	86.93	90.25	89.53	9.91
+SensiEdit	**92.99**	**91.65**	**91.87**	90.08	**88.43**	**87.67**	**91.10**	**89.80**	**5.55**

Table 3 evaluates object-existence judgements. Aggregate accuracy/F1 rises from 82.04/80.42 to 87.36/87.51 on LLaVA and from 86.43/85.13 to 91.10/89.80 on Qwen. Adversarial accuracy improves by 4.56 and 4.00 points, respectively, showing that the gains extend beyond random negatives. Costs are 8.14 and 5.55 TFLOPs, only 0.01 above each unedited backbone.

Table 4 shows the same trend: discriminative accuracy/F1 increases from 67.00/71.10 to 79.40/82.50 on LLaVA and from 84.10/88.60 to 86.40/90.00 on Qwen. Generative CHAIR and Hal decrease on both models, and overall AMBER rises from 81.58 to 89.50 and from 91.80 to 92.85, although Qwen Cover falls from 64.30 to 63.70.

Table 4: AMBER results.

Model	Setting	Generative				Discriminative			AMBER↑
		CHAIR↓	Cover↑	Hal↓	TFLOPs↓	Acc.↑	F1↑	TFLOPs↓	
LLaVA-1.5-7B	baseline	8.00	44.50	31.00	9.89	67.00	71.10	8.07	81.58
	+VCD	6.70	46.50	27.80	19.55	67.30	71.10	16.14	82.20
	+M3ID	6.00	48.90	26.00	19.81	67.25	70.90	16.14	82.25
	+AVISC	6.30	46.60	25.60	19.45	70.70	75.45	16.14	84.60
	+Octopus	4.80	49.20	23.40	20.48	76.70	82.70	16.35	88.95
	+HIRE	4.60	**49.80**	20.40	11.73	79.20	82.83	10.58	89.12
	+SensiEdit	**3.50**	49.60	**17.40**	**9.97**	**79.40**	82.50	**8.08**	**89.50**
Qwen2.5-VL-7B	Baseline	5.00	64.30	26.60	7.98	84.10	88.60	5.50	91.80
	+HIRE	4.70	57.90	23.20	9.85	84.50	87.96	7.76	91.63
	+SensiEdit	**4.30**	**63.70**	**19.50**	**8.47**	**86.40**	**90.00**	**5.51**	**92.85**

Across the two main backbones, SensiEdit reduces object hallucination in open-ended captioning and improves object-existence judgements while maintaining performance on the tested general-capability benchmarks. On LLaVA-NeXT-7B, Qwen3-VL-8B, and InternVL3-8B, it also improves CHAIR, POPE, and MMHal-Bench results relative to each model's baseline (Appendix C.3). The comparable trend across these additional models suggests that the effect is not specific to either main backbone. These results support the effectiveness of selective visual attention editing; the following sections examine its inference cost and the contributions of its individual components.

General capabilities. Table 5 assesses general visual understanding. On LLaVA, SensiEdit raises MME from 1778.9 to 1788.2 and MMMU from 14.86 to 14.96, with SEED-Bench unchanged at 63.68. On Qwen, MME rises from 2367.66 to 2406.98 and MMMU from 51.53 to 52.16, while SEED-Bench remains 76.12. Relative to HIRE on LLaVA and ChainMPQ on Qwen, SensiEdit scores higher on all three benchmarks; relative to the unedited baselines, it improves MME and MMMU while matching SEED-Bench.

Table 5: General-capability results on two backbones.

Setting	MME↑	SEED↑	MMMU↑
LLaVA-1.5-7B			
Baseline	1778.9	63.68	14.86
+HIRE	1778.9	63.48	14.56
+SensiEdit	**1788.2**	**63.68**	**14.96**
Qwen2.5-VL-7B			
Baseline	2367.66	76.12	51.53
+ChainMPQ	2389.90	73.48	51.56
+SensiEdit	**2406.98**	**76.12**	**52.16**

4.3 Inference Cost

Table 6 compares inference costs. SensiEdit uses 9.80 TFLOPs per response versus HIRE's 11.81; these length-dependent totals do not directly measure decoding speed. On the same images with 512 generated tokens each, median per-token decoding latency increases by 28.9% for SensiEdit versus 241% for HIRE. Peak reserved GPU memory increases by 32 versus 188 MiB. Thus, SensiEdit incurs less latency and memory overhead in this setting; its median first-token latency is nearly unchanged.

Table 6: Inference cost on LLaVA-1.5-7B. Baseline latency and memory are absolute; other values are changes.

Method	TFLOPs↓	Latency	Mem
Baseline	10.23	20.63 ms/token	14,202 MiB
+VCD	20.46	+99.93%	+556 MiB
+Octopus	21.39	+98.15%	+553 MiB
+HIRE	11.81	+241%	+188 MiB
+SensiEdit	**9.80**	**+28.9%**	**+32 MiB**

4.4 Ablation Studies

We evaluate module contributions (Table 7) and the choice of edited layers (Figure 3) on 500 images.

Table 7: Module ablation results on LLaVA-1.5-7B and Qwen2.5-VL-7B. Our full method offers the best overall balance between hallucination reduction and object recall.

Model	Variant	$CHAIR_S \downarrow$	$CHAIR_I \downarrow$	Recall↑	Len.	Inj.
LLaVA-1.5-7B	Baseline	51.30	16.80	**76.97**	83.9	–
	fixed-uniform	28.2	8.06	69.38	67.0	70.3%
	learned-per-head	30.8	8.94	69.68	67.3	70.7%
	always-on	**22.4**	6.73	54.25	120.1	100%
	full (SensiEdit)	22.6	**5.32**	**74.07**	**75.0**	70.1%
Qwen2.5-VL-7B	Baseline	46.20	9.60	68.29	159.0	–
	fixed-uniform	37.2	10.38	66.91	135.6	71.1%
	learned-per-head	37.4	8.71	66.85	134.8	71.0%
	always-on	36.6	8.76	66.75	285.6	100%
	full (SensiEdit)	**36.0**	**7.51**	**67.31**	**151.3**	71.96%

Table 7 compares fixed-uniform ($\alpha = 15$), spatially averaged learned-per-head, and ungated always-on editing; Len. and Inj. denote mean caption length and bias application rate. Full editing reaches LLaVA $CHAIR_S = 22.6$ with Recall 74.07, whereas removing the gate lengthens outputs and reduces Recall to 54.25.

Sensitivity guidance. In the single-run ablations of Appendix C.1, reversing head scores worsens both CHAIR metrics on both backbones. On Qwen, shuffling slightly improves $CHAIR_S$ but worsens $CHAIR_I$ and Recall; random gating worsens all three. These results support sensitivity guidance.

Figure 3 shows that LLaVA benefits more from middle-layer editing ($CHAIR_S = 35.0$) than deep-only editing (51.6) in a separate layer-ablation run on 500 images. Editing all eight layers achieves the lowest $CHAIR_I$ within each backbone.

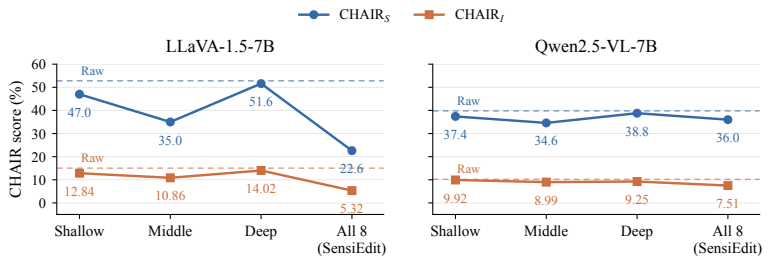

Figure 3: Layer-group ablation.

4.5 Qualitative Examples

Figure 4 pairs attention maps with caption excerpts. Editing increases visual attention mass from approximately 10% to 65% in these examples. Hallucinated mentions of people and a handbag in the train scene, or a dining table and vase in the bedroom, disappear from the edited excerpts.

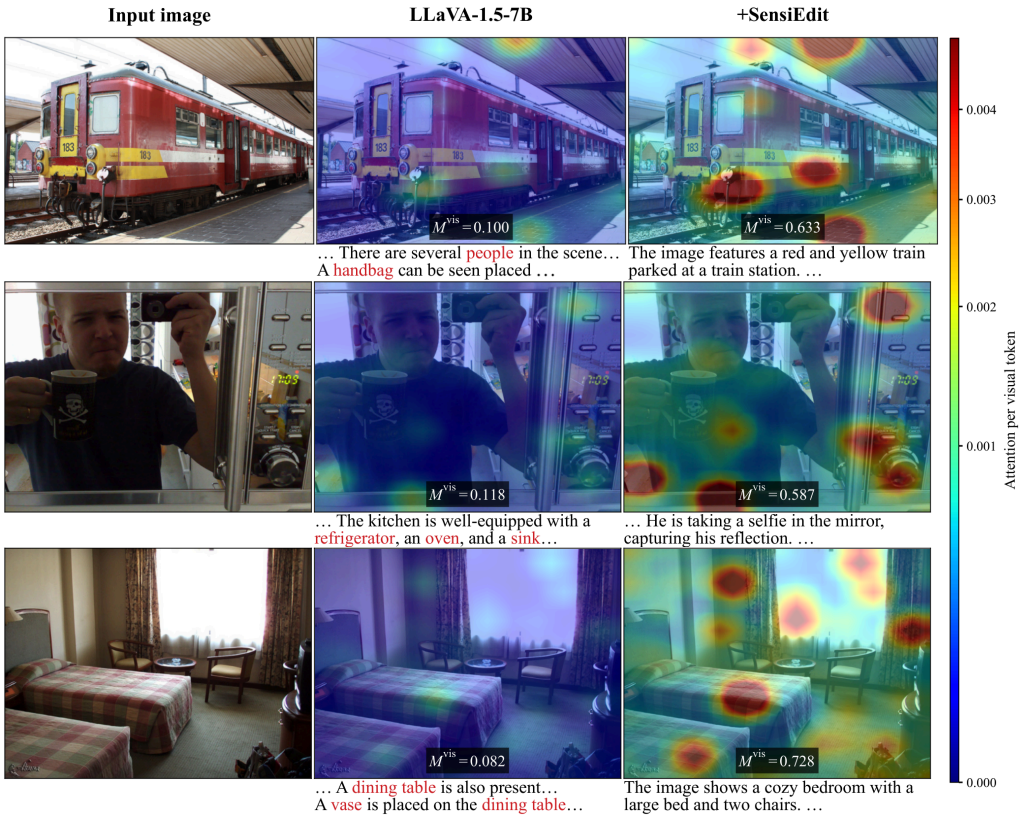

Figure 4: Attention maps and caption excerpts on COCO; M^{vis} denotes visual attention mass.

5 Conclusion and Limitations

We present SensiEdit, an attention editing framework that combines predicted head-wise visual sensitivity with learnable spatial biases to mitigate object hallucination in LVLMs. By selectively modulating visual attention during decoding, SensiEdit keeps the backbone frozen and requires no additional backbone forward pass. Experiments on LLaVA-1.5-7B and Qwen2.5-VL-7B show reduced object hallucination across CHAIR, POPE, and AMBER while maintaining performance on the evaluated general-capability benchmarks. Additional CHAIR, POPE, and MMHal-Bench results on three backbones support transfer beyond the two main models (Appendix C.3). The main limitation is that hallucination reduction can incur coverage and runtime costs.

AI USE STATEMENT

We used a large language model to assist with drafting and polishing the manuscript, checking consistency between text and reported results, and preparing figure-generation scripts and LaTeX formatting. prior-work results are attributed to their sources, and cost estimates are identified separately. The authors are responsible for verifying all AI-assisted text, code and figures against the underlying records and take full responsibility for the final content, claims and artifacts.

ETHICS STATEMENT

This work uses only publicly released models and benchmarks, listed with their licenses in Sec. B.2; training uses public COCO images, and evaluation uses COCO and the listed benchmarks. We introduce no new dataset, collect no data, and involve no human subjects or annotators. SensiEdit reduces object hallucination but does not eliminate it, and we would not advocate deploying an edited model as a sole safeguard in medical, legal, or accessibility settings where an unflagged false object causes harm. Finally, because the backbone remains frozen and only attention is redistributed, any social bias or unsafe association the underlying model carries is preserved: the method changes where the model looks, not what it knows.

REPRODUCIBILITY STATEMENT

Sec. 3 specifies the AVS signal, attention-logit editing, and training objective; Appendix A provides derivations and complexity analysis. Appendix B reports training settings, backbone-specific choices, checkpoint statistics, benchmark splits, metrics, and asset licenses. The evaluation protocol matches image IDs, prompts, seeds, and decoding parameters across baseline and edited runs, records applied bias injections, and interprets CHAIR alongside Recall and caption length. We also document the Qwen decoding issue (Sec. B.6), checkpoint selection (Sec. B.5), and limitations (Sec. E). We will release the implementation and trained predictor and spatial-bias checkpoints upon acceptance.

REFERENCES

Wenbin An, Feng Tian, Sicong Leng, Jiahao Nie, Haonan Lin, QianYing Wang, Ping Chen, Xiaoqin Zhang, and Shijian Lu. Mitigating object hallucinations in large vision-language models with assembly of global and local attention. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29915–29926. IEEE, 2025.

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*, pp. 2425–2433, 2015.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023a.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023b. URL https://arxiv.org/abs/2308.12966.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024.

Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Alessandro Favero, Luca Zancato, Matthew Trager, Siddharth Choudhary, Pramuditha Perera, Alessandro Achille, Ashwin Swaminathan, and Stefano Soatto. Multi-modal hallucination control by visual information grounding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14303–14312. IEEE, 2024.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *Advances in Neural Information Processing Systems*, 38, 2026.

Yuhan Fu, Ruobing Xie, Xingwu Sun, Zhanhui Kang, and Xirong Li. Mitigating hallucination in multimodal large language model via hallucination-targeted direct preference optimization. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 16563–16577, 2025.

MD Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CsUR)*, 51(6):1–36, 2019.

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. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13418–13427. IEEE, 2024.

Fushuo Huo, Wenchao Xu, Zhong Zhang, Haozhao Wang, Zhicheng Chen, and Peilin Zhao. Self-introspective decoding: Alleviating hallucinations for large vision-language models. In *International Conference on Learning Representations*, volume 2025, pp. 24272–24295, 2025.

Chaoya Jiang, Haiyang Xu, Mengfan Dong, Jiaxing Chen, Wei Ye, Ming Yan, Qinghao Ye, Ji Zhang, Fei Huang, and Shikun Zhang. Hallucination augmented contrastive learning for multimodal large language model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27026–27036. IEEE, 2024.

Nick Jiang, Anish Kachinthaya, Suzanne Petryk, and Yossi Gandelsman. Interpreting and editing vision-language representations to mitigate hallucinations. In *International Conference on Learning Representations*, volume 2025, pp. 63582–63605, 2025.

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. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13872–13882. IEEE, 2024.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023a.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PmLR, 2023b.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 292–305, 2023c.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26286–26296. IEEE, 2024.

Sheng Liu, Haotian Ye, and James Y Zou. Reducing hallucinations in large vision-language models via latent space steering. In *International Conference on Learning Representations*, volume 2025, pp. 72402–72419, 2025.

Jiaying Lu, Jinmeng Rao, Kezhen Chen, Xiaoyuan Guo, Yawen Zhang, Baochen Sun, Carl Yang, and Jie Yang. Evaluation and enhancement of semantic grounding in large vision-language models. *arXiv preprint arXiv:2309.04041*, 2023.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35:2507–2521, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 4035–4045, 2018.

Wei Suo, Lijun Zhang, Mengyang Sun, Lin Yuanbo Wu, Peng Wang, and Yanning Zhang. Octopus: Alleviating hallucination via dynamic contrastive decoding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29904–29914. IEEE, 2025.

Wei Suo, Hanzu Zhang, Lijun Zhang, Ji Ma, Peng Wang, and Yanning Zhang. Hallucination-aware intermediate representation edit in large vision-language models. *arXiv preprint arXiv:2603.29405*, 2026.

Ryutaro Tanno, David GT Barrett, Andrew Sellergren, Sumedh Ghaisas, Sumanth Dathathri, Abigail See, Johannes Welbl, Charles Lau, Tao Tu, Shekoofeh Azizi, et al. Collaboration between clinicians and vision–language models in radiology report generation. *Nature Medicine*, 31(2):599–608, 2025.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Junyang Wang, Yuhang Wang, Guohai Xu, Jing Zhang, Yukai Gu, Haitao Jia, Jiaqi Wang, Haiyang Xu, Ming Yan, Ji Zhang, et al. Amber: An llm-free multi-dimensional benchmark for mllms hallucination evaluation. *arXiv preprint arXiv:2311.07397*, 2023.

Xintong Wang, Jingheng Pan, Liang Ding, and Chris Biemann. Mitigating hallucinations in large vision-language models with instruction contrastive decoding. In *Findings of the association for computational linguistics: ACL 2024*, pp. 15840–15853, 2024.

Sangmin Woo, Donguk Kim, Jaehyuk Jang, Yubin Choi, and Changick Kim. Don't miss the forest for the trees: Attentional vision calibration for large vision language models, 2025. URL https://arxiv.org/abs/2405.17820.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, volume 2024, pp. 21875–21895, 2024.

Yutong Xie, Zhenglin Hua, Ran Wang, Wing WY Ng, Xizhao Wang, and Yuheng Jia. Finding the correct visual evidence without forgetting: Mitigating hallucination in lvlms via inter-layer visual attention discrepancy. *arXiv preprint arXiv:2605.20965*, 2026.

Le Yang, Ziwei Zheng, Boxu Chen, Zhengyu Zhao, Chenhao Lin, and Chao Shen. Nullu: Mitigating object hallucinations in large vision-language models via halluspace projection. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14635–14645. IEEE, 2025.

Jianghao Yin, Qin Chen, Kedi Chen, Jie Zhou, Xingjiao Wu, and Liang He. Dynamic multimodal activation steering for hallucination mitigation in large vision-language models. *arXiv preprint arXiv:2602.21704*, 2026.

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

Shaolei Zhang, Tian Yu, and Yang Feng. Truthx: Alleviating hallucinations by editing large language models in truthful space. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8908–8949, 2024.

Zhiyuan Zhao, Bin Wang, Linke Ouyang, Xiaoyi Dong, Jiaqi Wang, and Conghui He. Beyond hallucinations: Enhancing lvlms through hallucination-aware direct preference optimization. *arXiv preprint arXiv:2311.16839*, 2023.

Deyao Zhu, Xiaoqian Shen, Xiang Li, Mohamed Elhoseiny, et al. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *International Conference on Learning Representations*, volume 2024, pp. 18378–18394, 2024.

Zhengtao Zou, Ya Gao, Jiarui Guan, Bin Li, and Pekka Marttinen. Adaptive residual-update steering for low-overhead hallucination mitigation in large vision language models. *arXiv preprint arXiv:2511.10292*, 2025.

CONTENTS OF TECHNICAL APPENDICES

A Additional Method Details **15**

A.1 Odds-Ratio View of Logit Biasing 15

A.2 AVS as a Visual Half-L1 Distance 15

A.3 Complexity and Parameter Accounting 16

B Detailed Experimental Settings **16**

B.1 Benchmarks and Metrics 16

B.2 Existing Assets and Licenses 17

B.3 Backbones and Per-Model Instantiation 17

B.4 Implementation and Training Details 17

B.5 Checkpoint Selection 18

B.6 The Qwen2.5-VL Decoding Path 18

C Additional Ablation Studies **19**

C.1 Ablating the Sensitivity Signal 19

C.2 Output Quality 20

C.3 Cross-Backbone Generalization 20

D Additional Visualization Results **20**

E Overall Summary and Limitations **22**

A Additional Method Details

This section expands three points that Sec. 3 states without derivation: why a uniform additive bias on visual logits is an odds-ratio rescaling (Sec. A.1), what the AVS statistic of Eq. (3) does and does not measure (Sec. A.2), and where the parameter and compute budget actually goes (Sec. A.3).

A.1 Odds-Ratio View of Logit Biasing

Write $z_j = \boldsymbol{q}_{l,h}\boldsymbol{k}_{l,h,j}^{\top}/\sqrt{d} + m_j$ for the pre-softmax logit of key j, so that Eq. (1) reads $a_j = e^{z_j}/\sum_k e^{z_k}$. Define the visual odds

$$\operatorname{odds}\big(M_{l,h}^{\text{vis}}\big) \;=\; \frac{M_{l,h}^{\text{vis}}}{1 - M_{l,h}^{\text{vis}}} \;=\; \frac{\sum_{j\in\mathcal{V}} e^{z_j}}{\sum_{j\notin\mathcal{V}} e^{z_j}}, \tag{8}$$

where the partition function cancels. Applying Eq. (6) with a bias that is constant over visual positions, $b_{l,h,v} \equiv b$ for $v \in \mathcal{V}$ and 0 elsewhere, multiplies every visual numerator term by e^b and leaves the denominator untouched:

$$\operatorname{odds}\big(\tilde{M}_{l,h}^{\text{vis}}\big) \;=\; e^b \cdot \operatorname{odds}\big(M_{l,h}^{\text{vis}}\big), \qquad \text{equivalently} \qquad \operatorname{logit}\big(\tilde{M}_{l,h}^{\text{vis}}\big) \;=\; \operatorname{logit}\big(M_{l,h}^{\text{vis}}\big) + b. \tag{9}$$

This is the statement used in Sec. 3.1. The odds multiplier is e^b regardless of the starting visual mass, but the resulting mass is $\tilde{M}^{\text{vis}} = e^b M^{\text{vis}}/(1 - M^{\text{vis}} + e^b M^{\text{vis}})$. Thus, the absolute change depends on the initial allocation, motivating head-dependent modulation rather than guaranteeing that one constant bias is suitable for every head.

SensiEdit uses a spatially varying $\alpha_{l,h,v}$, so Eq. (9) holds only as a bracket. Writing $b_{\min} = \min_{v\in\mathcal{V}} b_{l,h,v}$ and $b_{\max} = \max_{v\in\mathcal{V}} b_{l,h,v}$, monotonicity of the numerator gives

$$e^{b_{\min}} \cdot \operatorname{odds}\big(M_{l,h}^{\text{vis}}\big) \;\le\; \operatorname{odds}\big(\tilde{M}_{l,h}^{\text{vis}}\big) \;\le\; e^{b_{\max}} \cdot \operatorname{odds}\big(M_{l,h}^{\text{vis}}\big), \tag{10}$$

The exact effective shift is $b_{\text{eff}} = \log \sum_{v\in\mathcal{V}} w_v e^{b_{l,h,v}}$, where $w_v = e^{z_v}/\sum_{j\in\mathcal{V}} e^{z_j}$; it is a log of an attention-weighted average of exponentiated biases.

Inverting Eq. (9) on the measurements of Table 1 recovers the *effective* bias the model actually experienced. For LLaVA-1.5-7B layer 4, M^{vis} moves from 0.076 to 0.854, i.e. the odds move from 0.082 to 5.85, a factor of 71 and hence $b_{\text{eff}} = \ln 71 \approx 4.3$. For Qwen2.5-VL-7B layer 3, $0.042 \to 0.962$ is a factor of 577, i.e. $b_{\text{eff}} \approx 6.4$. Both are far below the initialisation $\alpha_{\text{init}} = 15.0$ of Table 9, which is the expected consequence of the gate g_l and the insensitivity weight $1 - \hat{s}_{l,h}$ in Eq. (5) attenuating the raw table entries.

A.2 AVS as a Visual Half-L1 Distance

Eq. (3) sums $\frac{1}{2}|a^{(A)} - a^{(B)}|$ over visual positions only. Because $\boldsymbol{a}^{(A)}$ and $\boldsymbol{a}^{(B)}$ are normalised over all T keys, their restrictions to $\mathcal{V}$ are sub-probability vectors with masses $M^{\text{vis},A}$ and $M^{\text{vis},B}$, so AVS is a half-ℓ_1 distance between sub-probability vectors and coincides with the total-variation distance only when the two visual masses agree. It inherits the useful properties of TV regardless: it is bounded, $0 \le \text{AVS}_{l,h,t} \le \frac{1}{2}(M^{\text{vis},A} + M^{\text{vis},B}) \le 1$, symmetric, and finite when a position receives zero mass under one image — the case in which a KL-based sensitivity would diverge. Given how concentrated decode-step attention is (Table 1), that robustness matters in practice.

The triangle inequality exposes what the statistic mixes together:

$$\text{AVS}_{l,h,t} \;\ge\; \tfrac{1}{2}\Big|M_{l,h,t}^{\text{vis},A} - M_{l,h,t}^{\text{vis},B}\Big|. \tag{11}$$

A head therefore scores high either because swapping the image changes *how much* total mass it spends on the image, or because it changes *where* within the image that mass lands, and Eq. (3) does not separate the two.

Finally, the target is estimated from a *single* image pair (A, B) per caption. The choice of the contrast image B is a source of variance that the frozen Predictor then regresses over. The ablations in Sec. C.1 probe the downstream consequence of this signal being imperfect.

A.3 Complexity and Parameter Accounting

Trainable parameters are the spatial table α alone: $|\mathcal{L}| \times H \times N_v = 8 \times 32 \times 576 = 147{,}456$ for LLaVA-1.5-7B and $8 \times 28 \times 20 \times 20 = 89{,}600$ for Qwen2.5-VL-7B on the reference grid (Table 8), i.e. 0.002% and 0.001% of the respective backbones. The Visual Sensitivity Predictor adds 1341857 parameters for LLaVA and 1176733 for Qwen. Each backbone has its own predictor, trained in Stage 1, frozen thereafter, and shared across its edited layers through the learned layer embedding described in Sec. 3.2.

For LLaVA, the Predictor runs once per edited layer at each decode step, so 8 forward passes of a 1.34M-parameter MLP, about 21 MFLOPs in total. Against roughly 14 GFLOPs for one decode step of a 7B backbone this is $\approx 0.15\%$; the thresholded soft gate activates the bias on 73.9% of layer–decode-step events in the fixed-length LLaVA timing run, versus 70.1% in the captioning run of Table 7. The bias application itself is one add over $|\mathcal{L}| \times H \times N_v$ entries (147K values for LLaVA) into the attention scores before softmax, through the already-materialised attention mask; for Qwen, *grid_interp* additionally resamples the reference table to the current N_v once per forward pass, whereas LLaVA reads the table directly. Peak *reserved* memory grows by 32 MiB (Table 6); peak *allocated* memory grows by only 5.7 MiB, matching the resident weights of the frozen Predictor and α table (1489313 fp32 parameters =5.68 MiB). The bias tensor itself is a single 147K-entry array (0.56 MiB) and measured peak allocated memory differs mainly by the resident weights in this timing run.

This per-step accounting should not be confused with the totals in Table 6, where SensiEdit reports 9.80 TFLOPs against a 10.23 TFLOPs baseline. That column integrates over the whole generated response, and SensiEdit produces shorter captions; the arithmetic per token is strictly higher, and the measured median decoding latency increases by 28.9% in the current implementation, which uses Python-level forward hooks.

B Detailed Experimental Settings

B.1 Benchmarks and Metrics

CHAIR. On MSCOCO captioning we use the standard sentence-level and instance-level hallucination rates (Rohrbach et al., 2018). CHAIR_S is the fraction of captions containing at least one object absent from the ground-truth annotation; CHAIR_I is the fraction of mentioned object instances that are absent.Final numbers use a 500-image split of COCO val2014.

POPE. The Polling-based Object Probing Evaluation (Li et al., 2023c) converts hallucination into a balanced binary decision: the model is asked whether a given object is present, over *random*, *popular* and *adversarial* samplings of the negative objects, the last drawing negatives from objects that frequently co-occur with the scene. We report Accuracy and F1 over the full 8910-item split.

AMBER. AMBER (Wang et al., 2023) evaluates both generative and discriminative hallucination over 15220 items. We follow the setup of Suo et al. (2025) and report CHAIR, Cover and Hal on the generative task, defined below.

MME. MME (Fu et al., 2026) probes perception and cognition over yes/no questions across fourteen subtasks; we report the overall score, which is the sum of per-subtask accuracy and accuracy-plus terms and is therefore not on a percentage scale.

SEED-Bench. SEED-Bench (Li et al., 2023a) is a multiple-choice benchmark spanning spatial and temporal understanding; we evaluate the full 14232-item split and report accuracy.

MMMU. MMMU (Yue et al., 2024) targets college-level multi-discipline reasoning over figures, diagrams and charts. We report accuracy on the validation split.

For AMBER we reuse the definitions of Wang et al. (2023). Let G_{obj} be the objects mentioned in the generated response and A_{obj} the annotated objects. ***CHAIR*** measures the proportion of mentioned objects absent from the annotation,

$$CHAIR = 1 - \frac{len(G_{obj} \cap A_{obj})}{len(G_{obj})}, \tag{12}$$

Cover the proportion of annotated objects that were mentioned,

$$Cover = \frac{len(G_{obj} \cap A_{obj})}{len(A_{obj})}, \tag{13}$$

and *Hal* the fraction of responses containing at least one hallucination,

$$Hal = \mathbb{I}(CHAIR > 0). \tag{14}$$

CHAIR and Hal capture complementary aspects: CHAIR is an object-level rate aggregated over mentions, whereas Hal is a response-level indicator, and improving one does not in general improve the other. In the Qwen2.5-VL results of Table 4 both improve (CHAIR $5.0 \rightarrow 4.3$, Hal $26.6 \rightarrow 19.5$) while Cover decreases slightly ($64.3 \rightarrow 63.7$); we report the metrics together to keep that factuality/coverage trade-off visible.

B.2 Existing Assets and Licenses

LLaVA-1.5 (Liu et al., 2024) is available under the Apache-2.0 License, and Qwen2.5-VL (Bai et al., 2025) is available under the Apache-2.0 License. CHAIR is under the BSD 2-Clause License, POPE is available under the MIT License, and AMBER is available under the Apache-2.0 License. Training uses public COCO images, and evaluation uses COCO and the listed benchmarks; we introduce no new dataset.

B.3 Backbones and Per-Model Instantiation

Table 8 lists the per-model instantiation. The two backbones differ in three ways that matter for editing: head count and grouping (32 MHA heads for LLaVA versus 4 KV groups under GQA for Qwen), visual token count (a static 24×24 grid versus dynamic resolution), and attention implementation (eager versus SDPA).

Table 8: Per-model instantiation of SensiEdit. Trainable parameters count the spatial table α only; the Visual Sensitivity Predictor (≈ 1.34M) is frozen and shared.

Model	d_{model}	H	**KV heads**	**Visual tokens**	**Spatial mode**	α **shape**	**Trainable**
LLaVA-1.5-7B	4096	32	32 (MHA)	576 (24×24)	fixed	$[8, 32, 576]$	147,456
Qwen2.5-VL-7B	3584	28	4 (GQA)	dynamic ($\sim$358*)	grid_interp	$[8, 28, 20, 20]$	89,600

*Visual token count varies with resolution; each token aggregates 2×2 input patches.

B.4 Implementation and Training Details

Table 9 lists the full training configuration, which is identical for both models except for the α tensor shape and the spatial mode. During Stage 2, only α receives gradients; the Predictor and LVLM are frozen. At inference, computed biases are clipped to $[0, 25]$ before injection.

Table 9: Training configuration. The same recipe is used for both backbones.

Training data	3,000 images from COCO train2014
Epochs	50
Optimiser	AdamW
Learning rate	0.03, cosine annealing to $\eta_{\min} = 0.005$
Gradient accumulation	4
Objective	$1.0 \cdot \mathcal{L}_{\text{KD}} + 0.2 \cdot \mathcal{L}_{\text{CE}} + 10^{-4} \cdot \|\alpha - \alpha_0\|^2$
Distillation temperature	$\tau = 2.0$
α initialisation	15.0 (the best constant bias found by grid search)
Gate threshold	0.5
Hardware	4 GPUs, torchrun
Attention implementation	eager (LLaVA), sdpa (Qwen2.5-VL)
Decoding at evaluation	greedy, seed=42

Table 10: Training statistics of the released checkpoints.

Model	Spatial mode	α shape	Best val. loss	α mean / std
LLaVA-1.5-7B	fixed	$[8, 32, 576]$	–	–
Qwen2.5-VL-7B	grid_interp	$[8, 28, 20, 20]$	0.8901	13.62 / 4.22

Table 10 records the training statistics. The learned α on Qwen has mean 13.62 and standard deviation 4.22 around an initialisation of 15.0: training moves the tensor meaningfully away from the constant it started at, and the spread across layers, heads and grid cells is roughly 30% of the mean. This is the quantity the *learned-per-head* ablation of Table 7 destroys by averaging over positions.

B.5 Checkpoint Selection

The LLaVA training run saves checkpoints every ten epochs, and we evaluated several of them on the 500-image CHAIR split; Figure 5 plots the trend and Table 11 reports the corresponding results.

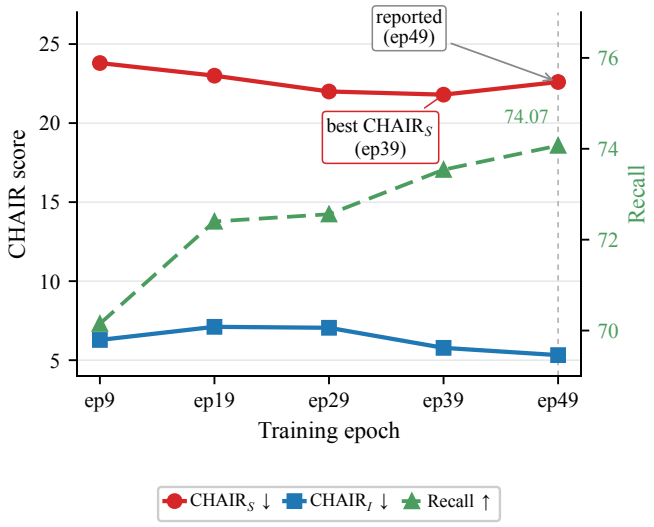

Figure 5: Checkpoint dynamics for LLaVA-1.5-7B on the 500-image CHAIR split.

Table 11: Checkpoint CHAIR evaluation on LLaVA-1.5-7B using 500 images.

Checkpoint	CHAIR$_S$↓	CHAIR$_I$↓
ep9	23.8	6.28
ep19	23.0	7.11
ep29	22.0	7.05
ep39	**21.8**	5.78
ep49 (reported)	22.6	**5.32**

We report **ep49**, the last epoch, to avoid selecting an epoch by its test-set CHAIR score. This checkpoint achieves CHAIR$_S$/CHAIR$_I$ = 22.6/5.32 and Recall 74.07, matching Table 7. Across the last four checkpoints (ep19–ep49), the ranges are 1.2 points in CHAIR$_S$ and 1.79 points in CHAIR$_I$. These describe variation along one training trajectory, not variability across independent runs.

B.6 The Qwen2.5-VL Decoding Path

Qwen2.5-VL uses SDPA attention, and at decode time its attention_mask is frequently None, with the causal mask constructed internally from the cache position. Our first implementation returned early whenever position_ids was present—correct for LLaVA, wrong for Qwen's mRoPE, where

single-token decoding also carries position_ids. The result was $bias_calls = 0$: the editor was loaded, the hooks were registered, and no bias was ever applied.

All Qwen numbers in this paper come from the corrected path, which builds the causal mask from cache positions and permits gated injection at $q_{\text{len}} = 1$ steps; the 71.96% injection rate reported in Table 7 is the direct evidence that the fix took effect. We describe this failure at length because it is the reason for the $bias_calls$ acceptance rule above, and because any work that edits attention through forward hooks is exposed to the same class of bug.

C Additional Ablation Studies

Sec. 4.4 ablates the three *modules* of Eq. (5). Here we examine the *signal* that drives them: whether the per-head ordering produced by the Predictor matters (Sec. C.1), and the output-quality cost (Sec. C.2). We also report a cross-backbone evaluation on three additional models (Sec. C.3).

C.1 Ablating the Sensitivity Signal

Eq. (5) weights the bias by $1 - \hat{s}_{l,h}$, so the claim being tested is that the Predictor's *ordering* of heads carries information, not merely that some modulation is applied. We compare the full model against four interventions that keep the editing machinery intact and corrupt only the signal: *equal-head* replaces $\hat{s}_{l,h}$ with its mean over heads, *shuffle-head* permutes it randomly across heads, *reverse-head* inverts the ordering so that the most visually sensitive heads receive the strongest push, and *random-gate* replaces the learned gate g_l with a randomized gate matched in expected firing rate. The three head interventions rescale their biases to match the full model's summed bias after clipping at each evaluated layer and step. Figure 6 reports all four on both backbones.

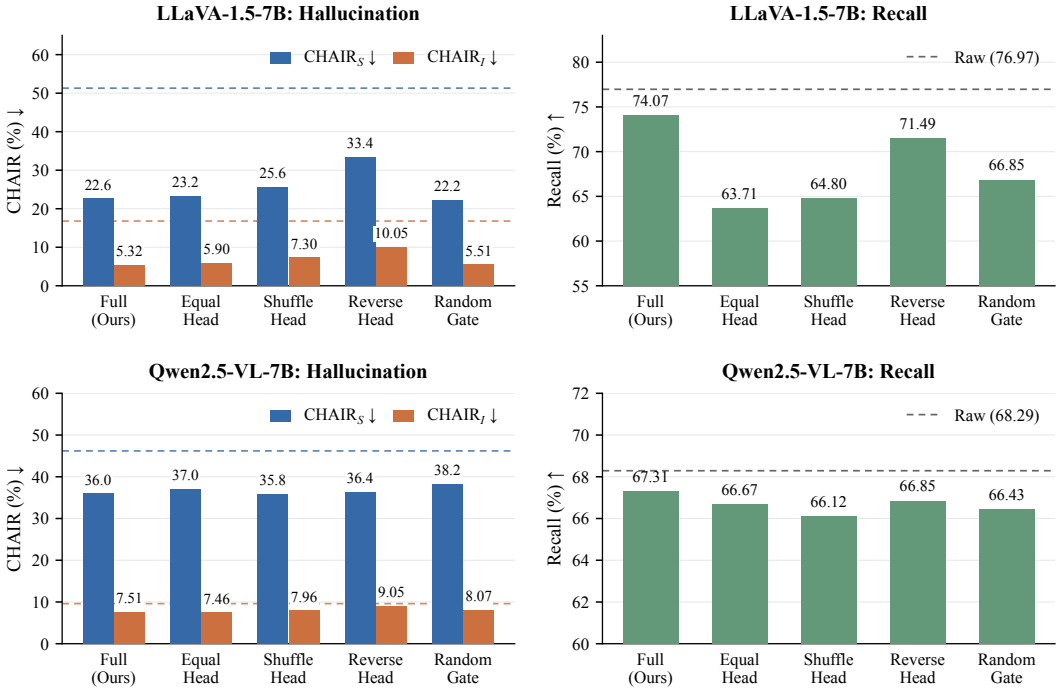

Figure 6: Ablating the sensitivity signal on the 500-image CHAIR split. Dashed lines mark the unedited backbone; baseline and full-method values match Table 7. Head-ordering interventions worsen both LLaVA CHAIR metrics and reduce Recall, whereas Qwen exhibits mixed effects across the two CHAIR metrics.

On LLaVA-1.5-7B, full editing achieves $\text{CHAIR}_S/\text{CHAIR}_I = 22.6/5.32$ and Recall 74.07. Equal-head, shuffle-head, and reverse-head increase CHAIR_S to 23.2, 25.6, and 33.4, respectively; all three

also increase CHAIR_I and reduce Recall. Reverse-head nearly doubles CHAIR_I, from 5.32 to 10.05, while Recall falls from 74.07 to 71.49. The larger degradation under reversed ordering than under equal-head weighting supports the usefulness of the predicted head ordering in this evaluation.

On LLaVA, *random-gate* slightly lowers CHAIR_S relative to full editing (22.2 vs. 22.6), but increases CHAIR_I (5.51 vs. 5.32) and reduces Recall (66.85 vs. 74.07). Thus, the learned gate gives a better instance-level hallucination rate and object Recall, although it does not achieve the lowest sentence-level hallucination rate in this comparison.

The Qwen2.5-VL-7B results show a mixed head-ordering effect. Full editing achieves $\text{CHAIR}_S/\text{CHAIR}_I = 36.0/7.51$. Equal-head increases CHAIR_S to 37.0 but slightly lowers CHAIR_I to 7.46, while shuffle-head slightly lowers CHAIR_S to 35.8 but raises CHAIR_I to 7.96. Reverse-head increases both metrics to $36.4/9.05$. Thus, reversing the sensitivity assignment is detrimental on both CHAIR metrics, but full editing does not dominate every head-ordering control on both metrics. Random-gate increases $\text{CHAIR}_S/\text{CHAIR}_I$ to $38.2/8.07$, with Recall 66.43 versus 67.31 for full editing; this run supports the learned gate on Qwen. All edited variants improve both CHAIR metrics over the unedited Qwen backbone ($46.20/9.60$), with lower Recall.

C.2 Output Quality

Because SensiEdit pushes probability mass towards visual evidence, a natural worry is degenerate repetition. We counted captions containing a phrase repeated five or more times, and captions longer than 100 words, over the 500-image split:

Variant	$\geq 5\times$ phrase repeat	Long output (> 100 words)
Baseline LLaVA-1.5-7B	0	0
Uniform $\alpha = 15$	0	0
SensiEdit (ep9)	23 (4.6%)	30 (6.0%)
SensiEdit (ep39)	29 (5.8%)	30 (6.0%)
SensiEdit (ep49, reported)	27 (5.4%)	28 (5.6%)

Later checkpoints show slightly more phrase-level repetition than Baseline, and none of the captions degenerate at the sentence level. We report this because it is a cost the CHAIR/Recall/length columns do not capture: roughly one caption in twenty repeats a phrase that the unedited model would not have repeated.

C.3 Cross-Backbone Generalization

Table 12 evaluates transfer to LLaVA-NeXT-7B, Qwen3-VL-8B, and InternVL3-8B. Baseline and SensiEdit are results from our evaluation; the ILVAD rows reproduce the published results of Xie et al. (2026). Baseline uses greedy decoding. Relative to the matched Baseline rows, SensiEdit lowers both CHAIR rates on all three backbones: $29.6/7.78 \rightarrow 22.4/4.87$ on LLaVA-NeXT, $52.3/9.67 \rightarrow 43.9/7.76$ on Qwen3-VL, and $21.1/8.79 \rightarrow 14.9/4.71$ on InternVL3. POPE Accuracy and F1 also rise on each model, while MMHal-Bench hallucination rate falls and score rises. Across the 18 backbone-by-metric comparisons with published ILVAD values, SensiEdit is numerically better in 17; the sole exception is InternVL3 POPE F1 (87.61 versus 87.95).

D Additional Visualization Results

Figure 7 extends the attention visualization of Figure 4 to six additional COCO val2014 images. Across these examples, the baseline allocates about 9–11% of attention to visual tokens; the remaining mass is distributed over non-visual positions. After editing, visual attention mass rises consistently to 64–72%, and the spatial heatmaps show that the redistributed attention concentrates on semantically relevant image regions rather than spreading uniformly. Read together with Sec. A.1, the mass figures correspond to an aggregate effective shift of roughly 2.7–3.2 nats on the visual/non-visual odds.

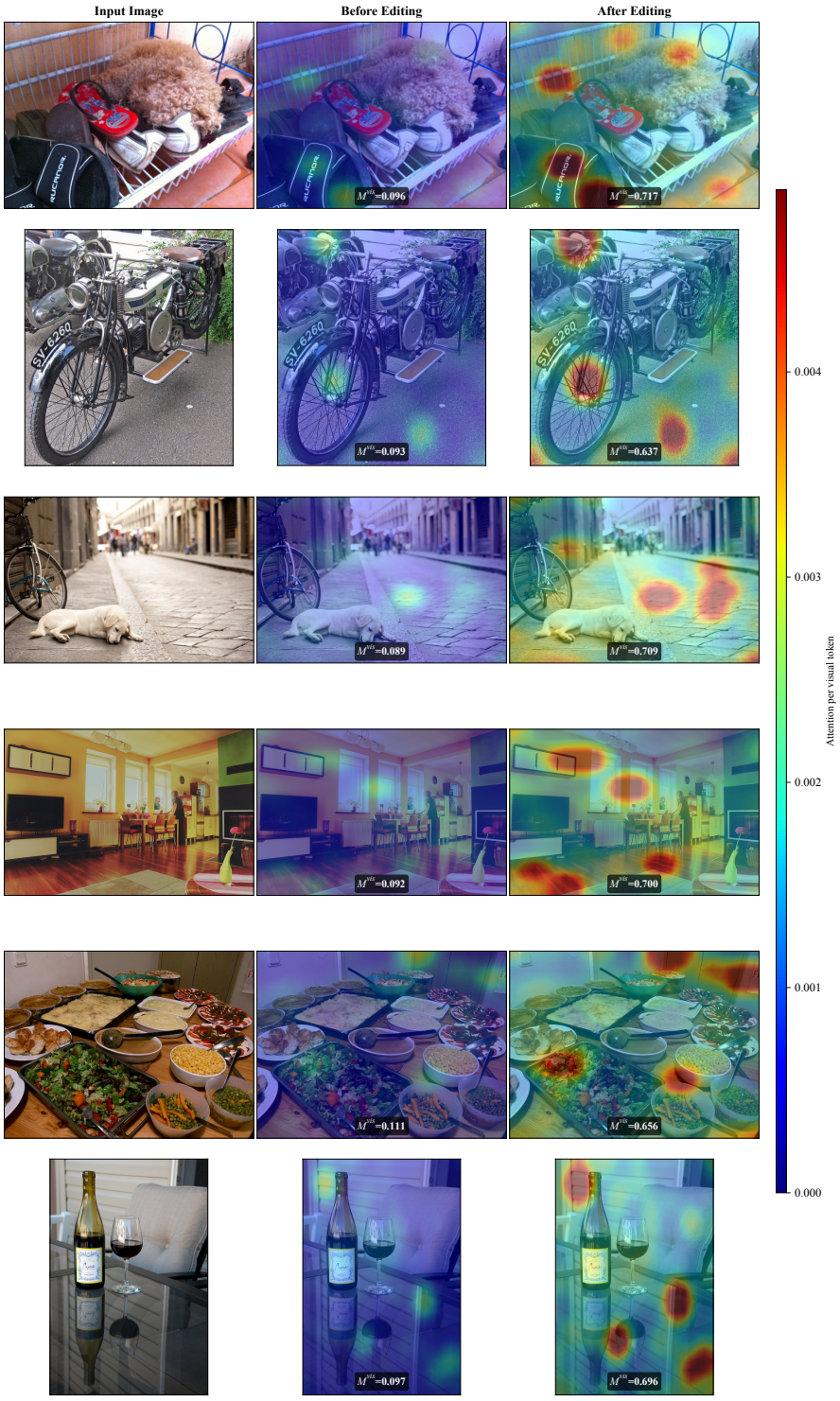

Figure 7: Additional attention visualizations on six COCO val2014 images. Each row shows the input image, the decode-step visual attention heatmap before editing, and the heatmap after editing with SensiEdit. M^{vis} denotes the total attention mass on visual tokens, averaged over target layers and heads.

Table 12: Cross-backbone hallucination evaluation. Baseline (greedy decoding) and SensiEdit are from our evaluation; ILVAD values are from Xie et al. (2026). Len. is mean caption length.

Backbone	Method	CHAIR			POPE		MMHal-Bench	
		$CHAIR_S \downarrow$	$CHAIR_I \downarrow$	Len.	Acc.↑	F1↑	Hal.↓	Score↑
LLaVA-NeXT-7B	Baseline	29.6	7.78	159.20	87.89	87.21	63.3	2.14
	ILVAD	25.8	6.52	152.87	88.92	89.08	56.8	2.57
	SensiEdit	**22.4**	**4.87**	151.65	**89.95**	**90.16**	**54.3**	**2.61**
Qwen3-VL-8B	Baseline	52.3	9.67	318.23	89.20	88.57	35.2	4.28
	ILVAD	45.2	8.52	310.59	89.80	89.62	32.6	4.32
	SensiEdit	**43.9**	**7.76**	311.55	**90.15**	**89.99**	**30.75**	**4.64**
InternVL3-8B	Baseline	21.1	8.79	101.59	87.47	86.15	48.8	2.74
	ILVAD	16.8	4.94	104.04	88.31	**87.95**	47.6	2.76
	SensiEdit	**14.9**	**4.71**	101.85	**88.52**	87.61	**46.6**	**2.80**

E Overall Summary and Limitations

This paper studies object hallucination through decoding-time attention allocation. In selected layers of LLaVA-1.5-7B and Qwen2.5-VL-7B, visual tokens receive little attention relative to an early non-visual sink (Table 1). We introduce SensiEdit, which predicts head-wise sensitivity to image replacement and applies learned spatial biases to visual-key logits at selected decoding steps while keeping the backbone frozen. Matched evaluations on the two primary backbones reduce CHAIR hallucination rates and improve POPE accuracy/F1 and AMBER scores, with no decline on the reported MME, SEED-Bench, and MMMU results. Comparisons with matched baselines on three additional backbones show gains on CHAIR, POPE, and MMHal-Bench (Sec. C.3). Together, these results support selective attention editing in the evaluated settings.

Limitations and future work. First, AVS measures changes in attention after image replacement; its relation to a head's causal reliance on visual evidence remains untested. Second, lower hallucination rates can reduce coverage and affect output quality: LLaVA object Recall falls from 76.97 to 74.07 (Table 7), and 27 of 500 edited captions repeat a phrase at least five times (Sec. C.2). Caption length also changes, so future objectives should assess hallucination, Recall, and output quality jointly.