RETHINKING CONTRASTIVE DECODING: EVIDENCE GUIDED BRANCH ASSIGNMENT FOR MULTIMODAL HALLUCINATION MITIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal large language models (MLLMs) integrate visual perception with the language understanding and generation capabilities of large language models, enabling a wide range of multimodal tasks. However, object hallucination remains a critical challenge that undermines the reliability of MLLMs. Existing contrastive decoding methods mitigate hallucinations by constructing positive and negative branches through visual perturbation, token selection, or attention modulation. However, the roles of these branches are often determined by predefined operations rather than by explicitly verifying whether their contents are supported by the input image. To address this limitation, we propose Visual Evidence guided Role Assignment (VERA), an inference time hallucination mitigation framework that verifies intermediate visual hypotheses before constructing contrastive branches. VERA assigns visually supported components to the positive branch, while uncertain hypotheses are reverified before their unresolved residuals are used for negative suppression. To prevent residual information from being inadvertently converted into positive evidence during contrastive subtraction, VERA further constrains the residual branch to provide suppression only. Extensive experiments on CHAIR, POPE, and five general visual question answering benchmarks demonstrate that VERA effectively reduces object hallucinations while preserving the general multimodal understanding capabilities of MLLMs across LLaVA-1.5, LLaVA-NeXT, and Qwen2.5-VL.

1 INTRODUCTION

In recent years, Large Language Models (LLMs) have developed rapidly, significantly improving work efficiency in fields such as natural language processing and machine translation Brown et al. (2020); Wei et al. (2021); Wang et al. (2023b); Hu et al. (2024). Multimodal Large Language Models (MLLMs) combine the strengths of visual and large language models, with representative examples including LLaVA and Qwen2.5-VL Liu et al. (2023); Bai et al. (2025). These multimodal large language models are capable of handling a range of complex tasks, including text-image understanding, image tracking, and facial recognition Chen et al. (2024); Han et al. (2025). Despite these achievements, a critical issue of object hallucination persists Rohrbach et al. (2018); Li et al. (2023), which refers to inconsistencies between visual inputs and textual outputs that undermine the reliability and credibility of MLLMs.

A key reason why MLLMs produce hallucinations is the conflict between linguistic prior knowledge and visual evidence Li et al. (2023; 2026). When a model relies heavily on statistical co-occurrences in language without making full use of image information, it may generate objects that do not actually exist in the image. To mitigate hallucinations, existing approaches can be divided into two main categories: resource-intensive strategies and lightweight strategies. Resource-intensive strategies use retraining or reinforcement learning to align model outputs but involve additional training resources and complex reward modeling requirements Yu et al. (2024); Sun et al. (2024); Yang et al. (2025). In some scenarios, they are costly and may even be difficult to implement. In contrast, lightweight strategies do not require retraining the model, directly correcting and filtering out erroneous outputs Leng et al. (2024); Wang et al. (2024); Huang et al. (2024); Liu et al. (2024d); Huo et al. (2025); Zou et al. (2025); Wang et al. (2026); Jiang et al. (2026). Based on lightweight

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.

strategies, contrastive decoding methods construct a branch that is more prone to producing hallucinations, and then compare it with the normal visual branch, thereby mitigating the erroneous predictions caused by linguistic prior knowledge.

Such as, Visual Contrastive Decoding (VCD) Leng et al. (2024) via original and distorted visual inputs contrasing output distributions to mitigate hallucinations. SID Huo et al. (2025) further constructs finer-grained contrastive signals by selecting informative visual tokens, while PAI Liu et al. (2024d) and ASCD Wang et al. (2026) directly modulate cross-modal attention to strengthen the model's reliance on visual information. MemVR Zou et al. (2025), from the perspective of visual information degradation, re-injects visual representations into intermediate layers to restore lost visual evidence. Although these methods mitigate hallucinations by intervening at different levels, such as visual inputs, visual tokens, and attention mechanisms, PND Jiang et al. (2026) further points out that relying solely on destructive negative pathways may simultaneously weaken useful visual evidence, thus reducing hallucinations but omitting genuinely present objects. We further quantify this effect in Fig. 2. As shown in Fig. 2(a), existing decoding interventions can suppress predictions that are already correct under vanilla decoding, while their ability to recover missed objects varies substantially across methods. Fig. 2(b) further shows that the recall of visually present objects drops markedly for small targets, where visual evidence is relatively difficult to capture. These observations suggest that weak visual responses should not be directly treated as negative evidence during branch construction. However, PND still determines branch construction based on perceptual importance, without explicitly verifying whether uncertain visual information represents hallucinated content or weakly supported but valid evidence. **Thus, the key challenge in contrastive decoding is not merely how to strengthen positive and negative branches, but how to determine what information should constitute each branch based on actual visual support.**

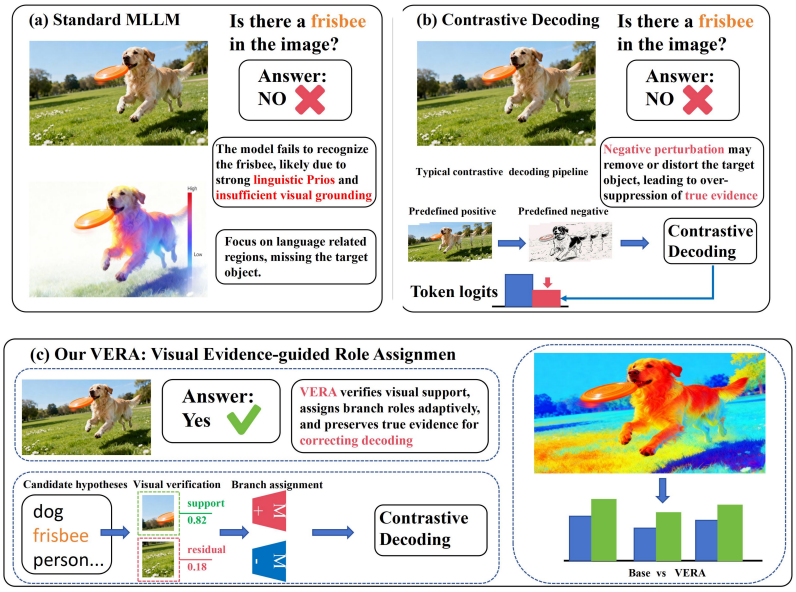

Figure 1: Motivation of the proposed VERA framework. (a) A standard MLLM may fail to recognize visually present objects when linguistic priors dominate weak visual evidence. (b) Existing contrastive decoding methods construct positive and negative branches through predefined visual manipulation. Such negative perturbations may distort or remove useful visual evidence, causing the assigned branch roles to deviate from the actual visual support and leading to over suppression of present objects. (c) VERA addresses this issue by verifying intermediate visual hypotheses against the original image before branch assignment. Visually supported components are assigned to the positive branch, while unresolved residuals are retained for constrained negative suppression, enabling more reliable contrastive decoding and recovery of visually grounded objects.

To address the above issues, we propose Visual Evidence guided Role Assignment (VERA), an inference-time hallucination mitigation framework. Unlike conventional contrastive decoding methods that predefine branch roles through visual enhancement or degradation, our method determines

branch composition according to verified visual support, assigning supported components to the positive branch and re-verified unresolved residuals to the negative branch. By grounding branch construction in verified visual support, VERA enables more reliable contrast between the two branches, suppressing unsupported objects while preserving visually supported ones. Our contributions can be summarized as follows:

- We propose VERA, a visual evidence guided contrastive decoding framework that addresses the unreliable branch construction problem in existing hallucination mitigation methods. By verifying intermediate visual hypotheses before branch assignment, VERA constructs positive and negative branches according to actual visual support, enabling more reliable contrastive decoding.
- We propose an evidence aware branch assignment strategy that distinguishes unsupported information from unresolved visual evidence. By rechecking uncertain hypotheses before suppression, VERA promotes visually supported information into the positive branch while retaining residual uncertainty as negative evidence, reducing the risk of excessive suppression in contrastive decoding.
- Extensive experiments on CHAIR, POPE, and five general visual question answering benchmarks demonstrate that VERA consistently outperforms existing decoding-based hallucination mitigation methods while preserving the general visual understanding capabilities of MLLMs.

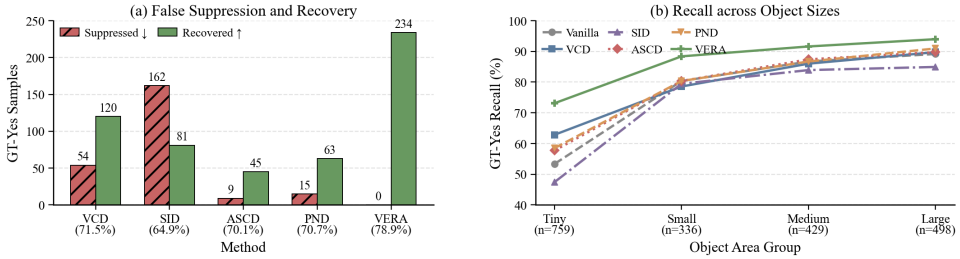

Figure 2: Effects of hallucination mitigation methods on the recognition of visually present objects. (a) Answer-level prediction changes on ground-truth positive (GT-Yes) samples relative to vanilla decoding. Suppressed denotes samples correctly answered by vanilla decoding but incorrectly answered after intervention, whereas Recovered denotes samples incorrectly answered by vanilla decoding but correctly answered after intervention. The GT-Yes recall of each method is shown in parentheses below its name. (b) GT-Yes recall across four object-area groups. The number of valid samples in each group is indicated in parentheses below the group label.

2 Related Work

2.1 Hallucination in Multimodal Large Language Models

Multimodal large language models (MLLMs), such as LLaVA and Qwen2.5-VL Liu et al. (2023); Bai et al. (2025), integrate visual perception with the language understanding and generation capabilities of LLMs, enabling a wide range of multimodal tasks, including image captioning, visual question answering, and multimodal reasoning. However, object hallucination remains a critical issue, where the generated content is inconsistent with the visual input, undermining the reliability and credibility of MLLMs Li et al. (2023); Guan et al. (2024b). Prior studies have identified three major factors that contribute to object hallucinations in MLLMs: over-reliance on language priors Favero et al. (2024); Leng et al. (2024), (2) insufficient visual perception and grounding Jiang et al. (2024); An et al. (2025), and (3) modality imbalance, where linguistic representations dominate visual information during multimodal fusion and decoding Liu et al. (2024d); Wang et al. (2026). Although hallucinated content is often plausible, it may not be visually factual, which undermines the reliability of MLLMs and limits their deployment in practical scenarios such as industrial inspection, remote sensing, and embodied navigation Liu et al. (2024b). Therefore, mitigating object hallucinations is essential for enabling the broader and more reliable deployment of MLLMs in real-world applications.

2.2 Contrastive Decoding for Hallucination Mitigation

Contrastive decoding has become a widely used strategy for mitigating object hallucinations without additional training Leng et al. (2024); Liu et al. (2024b). VCD Leng et al. (2024) constructs contrastive distributions by perturbing visual input, while SID Huo et al. (2025) further refines the contrastive signal through selective visual token manipulation. ASCD Wang et al. (2026) shifts the intervention from visual inputs to attention, directly adjusting the interaction between visual and textual information. In recent work, PND Jiang et al. (2026) has introduced positive and negative pathways, where the positive path enhances salient visual evidence, and the negative path degrades features of the core object to construct counterfactual predictions. Meanwhile, Mirage Yin et al. (2026) revisits the effectiveness of contrastive decoding and demonstrates that performance gains do not necessarily indicate improved visual grounding.

2.3 Hallucination Evaluation Benchmarks

Early evaluations of multimodal hallucinations often relied on human assessment or advanced language models, resulting in high evaluation costs Liu et al. (2024a); Guan et al. (2024b). CHAIR Rohrbach et al. (2018) provides an automatic metric for measuring object hallucinations in image captioning, while POPE Li et al. (2023) formulates object hallucination evaluation as binary visual questioning for more stable and flexible assessment. AMBER Wang et al. (2023a) further extends hallucination evaluation to both generative and discriminative settings, covering object existence, attributes, and relations without relying on external LLM judges. However, hallucination specific benchmarks mainly focus on visual faithfulness and cannot fully reflect whether mitigation methods preserve the general multimodal capabilities of MLLMs. Therefore, general VQA benchmarks Yu et al. (2023); Yue et al. (2024); Hudson & Manning (2019); Singh et al. (2019); Lu et al. (2022); Cheng et al. (2025); Li et al. (2024) are commonly adopted as complementary evaluations to assess the tradeoff between hallucination mitigation and overall task performance.

3 Preliminaries

Given an image I and a textual instruction x, an MLLM generates the output sequence autoregressively. During decoding, visual and textual representations are jointly processed by the Transformer, producing contextual hidden states together with query, key, and value representations. VERA uses these generation-time contextual states and the original visual key-value representations to verify the visual support of intermediate hypotheses.

3.1 Contrastive Decoding

Contrastive decoding mitigates hallucination by constructing positive and negative branches and contrasting their output distributions during generation. Let ℓ_t^+ and ℓ_t^- denote the logits produced by the positive and negative branches at decoding step t, respectively. A general contrastive formulation is

$$\ell_t^{\mathrm{CD}} = (1+\alpha)\ell_t^+ - \alpha\ell_t^-, \tag{1}$$

where α controls the contrastive strength.

Existing methods mainly differ in how the two branches are constructed. Negative branches can be obtained by perturbing visual inputs, while recent approaches further manipulate visual tokens or cross-modal attention. Their branch roles, however, are still largely determined by the predefined intervention itself. VERA instead determines branch composition according to verified visual support.

4 Methodology

In this section, We present Visual Evidence guided Role Assignment (VERA), an inference-time framework that constructs contrastive branches according to verified visual support rather than pre-

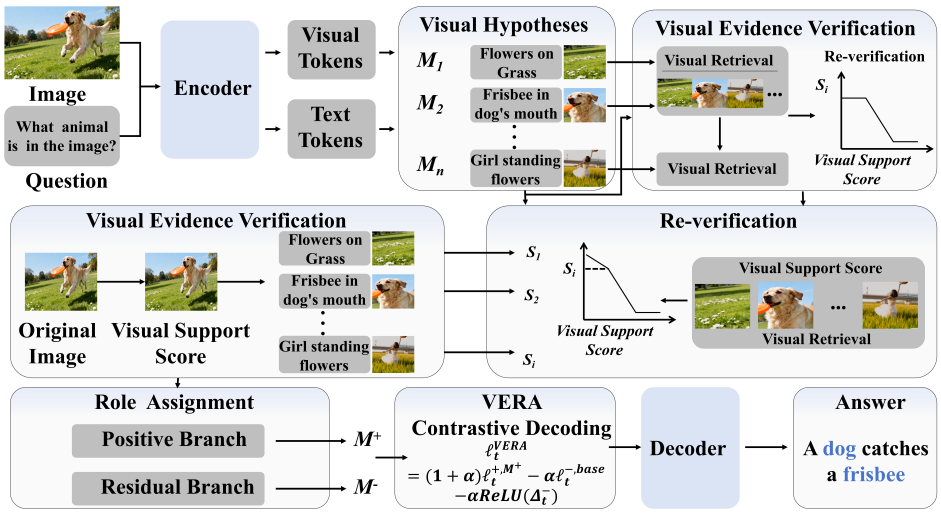

Figure 3: Overview of VERA. Given an image–question pair, the model first generates intermediate visual hypotheses together with their contextual hidden representations. VERA then rebinds each hypothesis to the original visual key–value states and estimates its visual-support score. Hypotheses with insufficient support are re-verified before being decomposed into visually supported memory M^+ and unresolved residual memory M^-. The supported component is assigned to the positive branch, while the residual component is used only for constrained negative suppression. The two branches are finally combined through contrastive decoding to produce the answer.

defined visual manipulation. As illustrated in Fig. 3, VERA first exposes intermediate visual hypotheses together with their generation-time contextual states and verifies them against the original image. Hypotheses with insufficient visual support are re-verified before branch assignment. The verified components are then assigned to the positive branch, while the remaining unresolved components are retained for constrained negative suppression. Algorithm 1 summarizes the inference procedure. The complete mathematical formulation is provided in Appendix C.

4.1 Evidence-Guided Branch Assignment

Given an image I and instruction x, the model first generates a temporary set of intermediate visual hypotheses

$$\mathcal{H} = \{h_1, h_2, \ldots, h_U\}. \tag{2}$$

Each h_i describes an object, attribute, or relation considered relevant to the current image and query.

VERA does not re-encode the textual surface form of h_i. Instead, it retains the contextual representation produced when the hypothesis is generated, including its hidden representation $\mathbf{r}_i$ and attention query $\mathbf{q}_i$. The query is then rebound to the original visual key-value representations $\mathbf{K}_I$ and $\mathbf{V}_I$:

$$\mathbf{a}_i = \mathrm{Softmax}\left(\frac{\mathbf{q}_i \mathbf{K}_I^\top}{\sqrt{d}}\right), \qquad \widetilde{\mathbf{v}}_i = \mathbf{a}_i \mathbf{V}_I. \tag{3}$$

The contextual representation $\mathbf{r}_i$ and visually rebound representation $\widetilde{\mathbf{v}}_i$ are used to estimate an initial visual-support score $s_i^{(0)}$. Here $s_i^{(0)} \in [0,1]$ denotes the estimated confidence that the intermediate hypothesis is supported by the original visual evidence. The complete support estimator is given in Appendix C.2.

A low initial support score is treated as uncertainty rather than immediate negative evidence. VERA therefore performs a bounded re-verification step against the original image and updates the support estimate in log-odds space:

$$s_i = \sigma\left(\mathrm{logit}\left(s_i^{(0)}\right) + \delta_i\right). \tag{4}$$

This second inspection allows weakly supported but valid hypotheses to recover visual support, while initially overestimated hypotheses can also be demoted. The final score s_i, rather than the initial score $s_i^{(0)}$, determines the subsequent supported/residual decomposition. After re-verification, the verified visual components form the supported memory $\mathcal{M}^+$ and the remaining unresolved components form the residual memory $\mathcal{M}^-$. Their complete construction is described in Appendix C.3.

4.2 Constrained Contrastive Decoding

The supported memory $\mathcal{M}^+$ is injected into the positive branch to reinforce visually grounded hypotheses. In parallel, $\mathcal{M}^-$ is introduced into the residual branch to characterize information that remains insufficiently supported after re-verification.

Let $\boldsymbol{\ell}_t^{+,M^+}$ denote the logits of the positive branch with supported memory, $\boldsymbol{\ell}_t^{-,\text{base}}$ the logits of the base residual branch without injecting $\mathcal{M}^-$, and $\boldsymbol{\ell}_t^{-,M^-}$ the logits of the same residual branch after injecting $\mathcal{M}^-$. Thus, $\boldsymbol{\ell}_t^{-,M^-} - \boldsymbol{\ell}_t^{-,\text{base}}$ isolates the token-level logit change introduced by unresolved residual memory.

Directly using $\boldsymbol{\ell}_t^{-,M^-}$ may introduce an undesirable effect. Since the negative branch is subtracted during contrastive decoding, a residual-induced decrease in the negative logits could be converted into an unintended positive contribution. VERA therefore retains only the suppressive residual change:

$$\Delta_t^- = \boldsymbol{\ell}_t^{-,M^-} - \boldsymbol{\ell}_t^{-,\text{base}}, \qquad \mathbf{P}_t^- = \text{ReLU}\left(\Delta_t^-\right). \tag{5}$$

Because the base residual branch is kept fixed when computing Δ_t^-, applying ReLU removes residual-induced decreases in the negative logits that would otherwise become positive contributions after contrastive subtraction.

The final contrastive logits are computed as

$$\boxed{\boldsymbol{\ell}_t^{\text{VERA}} = (1+\alpha)\boldsymbol{\ell}_t^{+,M^+} - \alpha\boldsymbol{\ell}_t^{-,\text{base}} - \alpha\mathbf{P}_t^-}. \tag{6}$$

Because $\mathbf{P}_t^- \geq 0$, the unresolved residual can suppress a candidate or leave its logit unchanged, but cannot directly introduce positive evidence through contrastive subtraction. We additionally retain the standard positive-branch plausibility constraint before selecting the next token. Detailed derivations and the complete decoding formulation are provided in the Appendix C.5.

5 Experiments

We evaluate VERA on object hallucination mitigation, general visual question answering, component effectiveness, branch assignment, and inference efficiency. The experiments examine whether VERA consistently reduces hallucinations across different MLLM backbones, preserves general multimodal capability, and benefits from visual-evidence-guided branch construction without introducing excessive inference overhead.

5.1 Experimental Setup

5.1.1 EXPERIMENTAL SETTINGS

Datasets. We evaluate hallucination mitigation on two representative benchmarks, CHAIR Rohrbach et al. (2018) and POPE Li et al. (2023). CHAIR measures object hallucination in open ended image captioning, while POPE evaluates object existence hallucination through binary visual questions. To further assess whether hallucination mitigation compromises the general multimodal understanding capabilities of MLLMs, we evaluate VERA on five widely used visual question answering benchmarks, including MME Fu et al. (2025), Hallusion Guan et al. (2024a), SEED-Bench Li et al. (2024), SimpleVQA Cheng et al. (2025), GQA Hudson & Manning (2019).

Models and Baselines. We utilize three representative MLLMs: LLaVALiu et al. (2023),Qwen2.5-VL Bai et al. (2025), LLaVA-NeXT Liu et al. (2024c) at the 7B scale. For detailed setting and

additional results, please refer to the Appendix D.1. As VERA constructs contrastive branches according to actual visual evidence, we compare it with four representative contrastive decoding methods, including VCDLeng et al. (2024), PND Jiang et al. (2026), ASCD Wang et al. (2026), and SID Huo et al. (2025). We further include CausalLens Ji et al. (2026) and HulluEdit Lin et al. (2026) for a more comprehensive comparison with recent hallucination mitigation methods.

Implementation Details. Throughout our experiments, we set the contrastive coefficient $\alpha = 1$, $\tau = 0.50$, $\lambda = 4.0$ and the plausibility threshold $\beta = 0.1$ unless explicitly stated otherwise. All baselines are evaluated with greedy decoding. For complete results under other decoding conditions, please refer to the Appendix F.4. The regular baseline directly decodes from the original model distribution, while contrastive baselines decode from a calibrated distribution after applying their corresponding visual or contextual intervention. All backbone models are frozen, and only the lightweight VERA projector is trained. Experiments are conducted on NVIDIA RTX 4090, and RTX Pro 6000 GPUs according to model size and resource availability. The full experimental settings are provided in Appendix D.

Table 1: Main comparison on CHAIR and POPE. Lower CHAIR_s and CHAIR_i indicate better hallucination mitigation. For POPE, we report accuracy (Acc.) and F1 score under the Random, Popular, and Adversarial sampling settings, where higher values indicate better performance. The best results are highlighted in bold.

Backbone	Method	CHAIR		POPE					
		$\text{CHAIR}_s \downarrow$	$\text{CHAIR}_i \downarrow$	Random		Popular		Adversarial	
				Acc. ↑	F1 ↑	Acc. ↑	F1 ↑	Acc. ↑	F1 ↑
Qwen2.5-VL	Vanilla	35.66	10.14	81.86	78.17	81.53	77.85	80.60	76.99
	VCD	25.33	9.02	83.76	80.66	83.23	80.15	82.96	79.90
	SID	22.00	8.63	85.53	83.16	85.00	82.65	84.20	81.89
	ASCD	15.33	8.45	84.86	82.25	84.53	81.93	83.93	81.36
	PND	33.00	8.58	85.26	82.74	84.73	82.23	84.06	81.60
	CausalLens	33.33	9.09	82.40	78.67	82.06	78.35	81.60	77.91
	HulluEdit	29.33	7.96	84.60	81.82	84.06	81.31	83.53	80.80
	VERA	**15.00**	**7.31**	**88.86**	**87.63**	**87.93**	**86.73**	**86.86**	**85.73**
LLaVA-1.5	Vanilla	66.66	26.20	80.40	81.82	77.20	79.47	71.33	75.48
	VCD	55.00	16.57	87.93	87.72	85.20	85.35	79.13	80.52
	SID	54.33	15.28	89.83	89.56	86.26	86.40	80.40	81.65
	ASCD	55.00	14.26	90.00	89.95	86.73	87.10	79.26	81.20
	PND	54.66	15.25	89.93	89.70	**87.33**	87.38	80.13	81.53
	CausalLens	53.32	13.78	90.20	90.24	86.26	86.84	78.80	81.04
	HulluEdit	56.66	15.61	90.06	89.81	87.20	87.25	80.13	81.51
	VERA	**50.00**	**13.75**	**91.00**	**90.52**	87.00	**87.45**	**81.00**	**82.68**
LLaVA-NeXT	Vanilla	38.50	14.27	86.50	86.20	85.83	85.62	80.33	81.09
	VCD	33.50	10.83	90.33	89.68	88.67	88.11	85.50	85.28
	SID	28.50	9.42	91.17	90.59	90.17	89.63	88.00	87.63
	ASCD	29.00	9.71	90.66	90.78	88.16	88.60	81.83	83.50
	PND	31.00	10.12	**91.50**	90.94	89.67	89.42	88.00	87.67
	CausalLens	34.00	**9.05**	90.67	90.54	87.17	87.44	82.67	83.75
	HulluEdit	28.50	9.98	91.17	90.59	90.33	89.79	88.00	87.63
	VERA	**27.50**	9.70	91.33	**90.97**	**90.50**	**89.98**	**88.50**	**87.76**

5.2 Experimental Results

Results on POPE: To verify the ability of VERA to mitigate object hallucinations, we conduct experiments under the Random, Popular, and Adversarial sampling settings of the POPE dataset (as shown in Table 1). Evaluated on Qwen2.5-VL, LLaVA-1.5, and LLaVA-NeXT, our method achieves steady improvements on the POPE benchmark. Compared with the vanilla decoding baseline, it obtains an average Accuracy gain of 7.4% and an F1-score gain of 7.5%. It is worth noting that VERA yields better overall performance than other baselines across various model backbones and sampling configurations, demonstrating its strong robustness against diverse data distributions and model architectures. At the same time, as the difficulty of the POPE sampling setup continues to increase, VERA is still able to maintain excellent experimental performance. This phenomenon

fully demonstrates that, thanks to the reasonable construction of its contrast decoding branches, VERA can effectively suppress hallucination outputs caused by nonexistent objects while also retrieving objects that actually exist in the image—including small objects that are easily overlooked by baseline models.

Results on Chair: To verify the plausibility of VERA's language generation output and its ability to suppress hallucinations, we used the CHAIR metric to quantitatively evaluate the model's output. The experimental results are shown in Table 1. Our method achieved good results on the CHAIR metric, effectively alleviating object hallucinations. These results indicate that VERA can efficiently suppress a large number of false hallucinated objects. Additionally, We present the complete results for POPE and Chair in the Appendix E.

Results on visual question answering benchmarks: To further validate VERA's generalization performance, we conducted experiments based on the Qwen2.5-VL across five general visual question answering benchmarks, as shown in Figure 4. VERA achieved the best average metrics, outperforming the original baseline (Vanilla) as well as comparative decoding methods such as VCD, ASCD, and PND. These results demonstrate that VERA can maintain or even improve performance in general visual question answering while suppressing multimodal hallucinations. It exhibits strong generalization capabilities across various VQA tasks and avoids performance degradation issues observed in some contrastive decoding methods on general tasks. To evaluate VERA under weak visual evidence, we use VisDrone-Dataset Zhu et al. (2021), which contains more than 2.6 million manually annotated bounding boxes for targets such as pedestrians, cars, bicycles, and tricycles to compare VERA with Vanilla, VCD Leng et al. (2024), ASCD Wang et al. (2026), and PND Jiang et al. (2026) on small-target object-existence questions in the Appendix F.3.

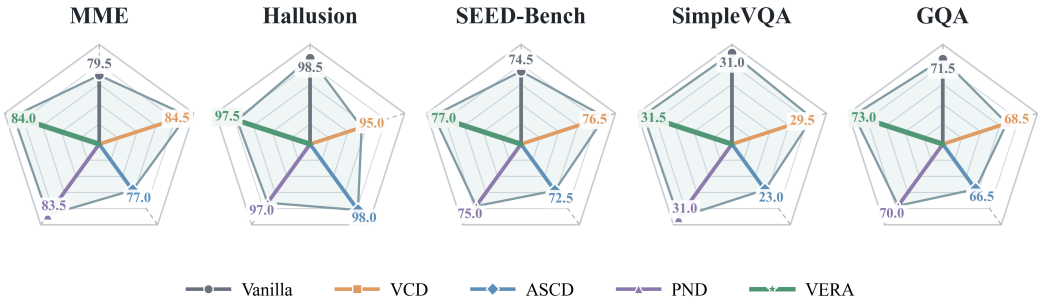

Figure 4: Performance on five general visual question answering benchmarks. Higher values indicate better performance.

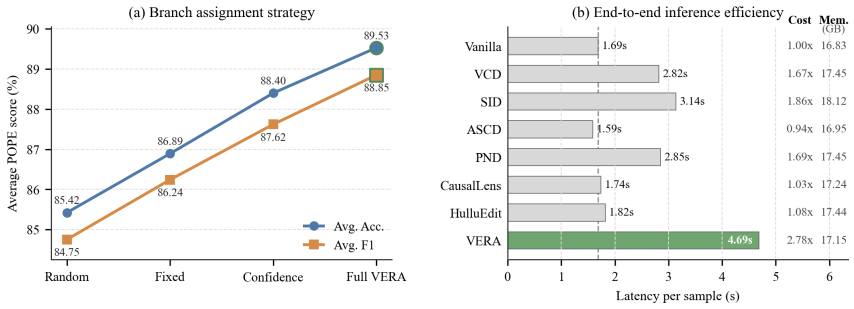

Figure 5: Branch assignment and inference efficiency analysis. (a) Comparison of different branch assignment strategies under the same POPE setting. (b) End-to-end inference latency on Qwen2.5-VL 7B. Cost is normalized to Vanilla, and peak memory is reported in GB.

Ablation Study: To validate the effectiveness of each module in VERA, we conduct ablation experiments on the POPE and CHAIR benchmarks (Table 2). We compare the full VERA model with

several ablated variants by removing individual components. The results demonstrate that all modules yield consistent performance gains, and the support and residual branches deliver distinct yet complementary effects. Specifically, the support branch reinforces visually grounded valid information, while the residual branch suppresses image-unsupported false hypotheses. The unilateral residual constraint alleviates undesired logit shifts during contrastive decoding, and the secondary verification mainly mitigates hallucinations for open-ended descriptions. These findings validate our design principle: each module performs a unique function, and their synergistic collaboration substantially strengthens the overall hallucination mitigation ability across diverse scenarios.

Table 2: Ablation Study of VERA on Qwen2.5-VL, evaluated on CHAIR and POPE.

Method	$CHAIR_s \downarrow$	$CHAIR_i \downarrow$	POPE Acc. ↑	POPE F1 ↑
w/o Intermediate Visual Hypotheses	20.00	7.23	87.22	83.31
w/o Visual Verification	14.00	15.48	87.22	83.79
w/o Supported Branch	16.00	10.61	87.78	84.48
w/o Residual Branch	17.00	9.48	86.67	83.63
w/o One-sided Residual Constraint	14.00	8.13	86.67	83.51
w/o Re-verification	18.00	9.56	87.77	84.71
VERA	**12.00**	**5.99**	**87.78**	**84.72**

Results on Branch assignment strategy: To analyze VERA's core branch allocation strategy, we conduct comparative experiments on the POPE benchmark (Figure 5 (a)). We compare our vision-perception-driven allocation strategy against three alternative schemes: confidence-based, fixed, and random allocation. The results demonstrate that full VERA achieves the best performance across all POPE subsets. This reveals that relying only on token confidence, static rules or random assignment is inadequate. Branch construction needs dynamic guidance from visual evidence to assign hypotheses to either the support branch or the residual branch.

Inference Efficiency We further evaluate the inference latency and GPU memory overhead of different methods on Qwen2.5-VL-7B, as reported in Figure 5 (b). VERA achieves competitive hallucination mitigation at the cost of moderate inference delay, with an average per-sample inference time of 4.69 s, which is 2.78 times slower than the vanilla baseline. The additional computational overhead mainly comes from intermediate visual hypothesis generation, visual evidence verification, and dual-branch contrastive decoding. Nevertheless, VERA only introduces negligible memory overhead. Its peak GPU memory usage is 17.15 GB, merely 0.32 GB higher than the vanilla decoder and substantially lower than VCD, SID, and PND. This indicates that VERA's extra cost primarily lies in computational latency rather than memory occupation. In summary, VERA trades a modest inference delay for prominent hallucination suppression, while maintaining superior memory efficiency. It achieves a favorable balance between performance and deployment cost in practical applications.

6 Conclusion

In this paper, we address object hallucination in MLLMs from the perspective of branch assignment in contrastive decoding. We propose Visual Evidence guided Role Assignment (VERA), an inference time framework that verifies intermediate visual hypotheses against the original image before constructing contrastive branches. Instead of directly treating weak visual support as negative evidence, VERA re-verifies uncertain hypotheses and assigns visually supported components to the positive branch, while retaining only the remaining unresolved residuals for constrained negative suppression. In addition, VERA prevents residual information from being converted into unintended positive contributions during contrastive decoding, enabling a more reliable interaction between the two branches. Extensive experiments on CHAIR, POPE, and five general visual question answering benchmarks demonstrate that VERA effectively mitigates object hallucinations while preserving the general multimodal understanding capabilities of MLLMs across multiple model architectures. Overall, our results demonstrate that reliable contrastive decoding depends not only on strengthening positive and negative branches, but more fundamentally on determining the appropriate role of visual information according to its actual visual support.

AI USE STATEMENT

In this work, we used generative AI tools to assist with research workflow management, evaluation-script preparation, LaTeX table formatting, figure drafting, and language polishing. We also used these tools to help inspect experimental logs, organize qualitative examples, and improve the clarity of the manuscript.

We did not use generative AI tools to autonomously generate scientific claims, fabricate or alter experimental results, determine authorship, or make final decisions about the reported conclusions. Tasks involving human-subject data or privacy-sensitive personal data are not applicable to this work.

All AI-assisted code, tables, figures, and text were reviewed by the authors. Experimental results reported in the paper were checked against saved evaluation logs or rerun scripts where applicable. The authors take full responsibility for the final content of this work, including text, claims, code, tables, figures, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

This paper provides the full algorithm formulation in Appendix C. Full details of hyperparameters, dataset settings and training configurations are summarized in Appendix D. All datasets used in this work are publicly available. Implementation details for baseline methods are also described in Appendix D.

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.

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, 2025. URL `https://arxiv.org/abs/2502.13923`.

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.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24185–24198, 2024.

Xianfu Cheng, Wei Zhang, Shiwei Zhang, Jian Yang, Xiangyuan Guan, Xianjie Wu, Xiang Li, Ge Zhang, Jiaheng Liu, Yuying Mai, Yutao Zeng, Zhoufutu Wen, Ke Jin, Baorui Wang, Weixiao Zhou, Yunhong Lu, Hangyuan Ji, Tongliang Li, Wenhao Huang, and Zhoujun Li. Simplevqa: Multimodal factuality evaluation for multimodal large language models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4637–4646, 2025. doi: 10.1109/ICCV51701.2025.00441.

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, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. Mme: A comprehensive evaluation benchmark for multimodal large language models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi,

and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025. doi: 10.52202/085713-4899. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/d79a27cf2772fe00be7f341efc0eb517-Paper-Datasets_and_Benchmarks_Track.pdf.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. Hallusionbench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14375–14385, 2024a. doi: 10.1109/CVPR52733.2024.01363.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14375–14385. IEEE, 2024b.

Yue Han, Jiangning Zhang, Junwei Zhu, Runze Hou, Xiaozhong Ji, Chuming Lin, Xiaobin Hu, Zhucun Xue, and Yong Liu. Groundingface: Fine-grained face understanding via pixel grounding multimodal large language model. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3942–3951. IEEE, 2025.

Yuchen Hu, Chen Chen, Chao-Han Huck Yang, Ruizhe Li, Dong Zhang, Zhehuai Chen, and Eng Siong Chng. Gentranslate: Large language models are generative multilingual speech and machine translators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 74–90, 2024.

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.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6693–6702. IEEE, 2019.

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.

Junyang Ji, Qifan Liu, Wenming Yang, and Zhihai He. Causallens: Sensitivity-guided multi-head causal intervention for hallucination mitigation in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4199–4209, June 2026.

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.

Yubo Jiang, Yitong An, Xin Yang, Abudukelimu Wuerkaixi, Xuxin Cheng, Fengying Xie, Zhiguo Jiang, Cao Liu, Ke Zeng, and Haopeng Zhang. Breaking the illusion: When positive meets negative in multimodal decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4210–4220, June 2026.

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, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. Seed-bench: Benchmarking multimodal large language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13299–13308, 2024. doi: 10.1109/CVPR52733.2024.01263.

Wei Li, Zhen Huang, Houqiang Li, Le Lu, Yang Lu, Xinmei Tian, Xu Shen, and Jieping Ye. Visual evidence prompting mitigates hallucinations in large vision-language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4048–4080, 2025.

Xinwei Li, Li Lin, Hui Jiao, Li Yao, Tien-Tsin Wong, and Hanqian Wu. Dive: Decoupling intra-layer visual evidence for mitigating hallucinations in large vision-language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 37552–37568, 2026.

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, 2023.

Yangguang Lin, Quan Fang, Yufei Li, Jiachen Sun, Junyu Gao, and Jitao Sang. Hulluedit: Single-pass evidence-consistent subspace editing for mitigating hallucinations in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11086–11095, June 2026.

Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Mitigating hallucination in large multi-modal models via robust instruction tuning. In *International Conference on Learning Representations*, volume 2024, pp. 57689–57733, 2024a.

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

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024c. URL `https://llava-vl.github.io/blog/2024-01-30-llava-next/`.

Shi Liu, Kecheng Zheng, and Wei Chen. Paying more attention to image: A training-free method for alleviating hallucination in lvlms. In *European Conference on Computer Vision*, pp. 125–140. Springer, 2024d.

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.

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.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8309–8318. IEEE, 2019.

Xiaoxi Sun, Jianxin Liang, Yueqian Wang, Huishuai Zhang, and Dongyan Zhao. Understanding visual detail hallucinations of large vision-language models. In *IJCAI*, pp. 1900–1908, 2025.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liangyan Gui, Yu-Xiong Wang, Yiming Yang, et al. Aligning large multimodal models with factually augmented rlhf. In *Findings of the association for computational linguistics: ACL 2024*, pp. 13088–13110, 2024.

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*, 2023a.

Longyue Wang, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. Document-level machine translation with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 16646–16661, 2023b.

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.

Yujun Wang, Jinhe Bi, Soren Pirk, Yunpu Ma, et al. Ascd: Attention-steerable contrastive decoding for reducing hallucination in mllm. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 10306–10314, 2026.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Zhihe Yang, Xufang Luo, Dongqi Han, Yunjian Xu, and Dongsheng Li. Mitigating hallucinations in large vision-language models via dpo: On-policy data hold the key. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10610–10620. IEEE, 2025.

Hao Yin, Guangzong Si, and Zilei Wang. The mirage of performance gains: Why contrastive decoding fails to mitigate object hallucinations in mllms? *Advances in Neural Information Processing Systems*, 38:33056–33079, 2026.

Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, et al. Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13807–13816, 2024.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.

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.

Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Heng Fan, Qinghua Hu, and Haibin Ling. Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7380–7399, 2021.

Xin Zou, Yizhou Wang, Yibo Yan, Yuanhuiyi Lyu, Kening Zheng, Sirui Huang, Junkai Chen, Peijie Jiang, Jia Liu, Chang Tang, and Xuming Hu. Look twice before you answer: Memory-space visual retracing for hallucination mitigation in multimodal large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 80873–80899, 2025.

A APPENDIX

This section provides the theoretical analysis, complete formulation, experimental details, extended quantitative results, additional analyses, and qualitative examples of VERA.

Appendix B analyzes the effect of branch assignment errors in contrastive decoding, the role of re-verification, and the one-sided residual constraint. Appendix C provides the complete formulation of VERA, including visual-support estimation, re-verification, memory construction, residual suppression, and plausibility-constrained decoding. Appendix D describes the model configurations, datasets, baseline methods, and implementation details. Appendix E reports the complete CHAIR and POPE results on Qwen2.5-VL, LLaVA-1.5, and LLaVA-NeXT. Appendix F contains additional analyses of visual support, residual suppression, weak visual evidence, decoding robustness, hyperparameter sensitivity, and inference efficiency. Appendix G presents representative qualitative examples.

Appendix Organization

Section	Page
B Theoretical Analysis of VERA	14
C Complete Formulation of VERA	16
D Detailed Experimental Settings	20
E Complete Experimental Results	23
F Additional Analyses	25
G Case Studies	29

B THEORETICAL ANALYSIS OF VERA

We analyze VERA from three aspects: the effect of branch assignment errors in contrastive decoding, the role of re-verification in reducing assignment risk, and the safety property introduced by one-sided residual suppression. We then discuss the applicability of VERA and its relation to existing contrastive decoding methods.

B.1 WHY BRANCH ASSIGNMENT MATTERS

Let ℓ_t^+ and ℓ_t^- denote the logits produced by the positive and negative branches at decoding step t. Standard contrastive decoding computes

$$\ell_t^{\mathrm{CD}} = (1+\alpha)\ell_t^+ - \alpha\ell_t^-, \tag{7}$$

where $\alpha \geq 0$ controls the contrastive strength.

Suppose $\ell_t^{+,*}$ and $\ell_t^{-,*}$ denote ideal branch logits whose semantic roles are consistent with the underlying visual evidence. In practice, branch construction may introduce role assignment errors, such that

$$\ell_t^+ = \ell_t^{+,*} + \epsilon_t^+, \qquad \ell_t^- = \ell_t^{-,*} + \epsilon_t^-. \tag{8}$$

The deviation from the ideal contrastive logits is therefore

$$\Delta\ell_t = (1+\alpha)\epsilon_t^+ - \alpha\epsilon_t^-. \tag{9}$$

This decomposition shows that contrastive decoding does not automatically eliminate branch construction errors. Such errors are propagated directly into the final decoding distribution. For a hallucinated candidate, unsupported information mistakenly assigned to the positive branch can be amplified by the factor $1+\alpha$. Conversely, visually supported information mistakenly assigned to the

negative branch can be penalized through contrastive subtraction. Since $1 + \alpha > \alpha$, contamination of the positive branch can be particularly harmful under equal error magnitude.

This motivates VERA to determine branch roles according to verified visual support before performing contrastive decoding, rather than assuming that a predefined enhancement or degradation operation necessarily produces a semantically valid positive or negative branch.

B.2 Effect of Re-verification on Role Assignment

For analysis, let $z_i \in \{0, 1\}$ denote whether an intermediate visual hypothesis h_i is supported by the input image, where $z_i = 1$ indicates visual support. Let $\hat{z}_i$ denote its assigned branch role, with $\hat{z}_i = 1$ corresponding to the positive component.

Two types of assignment errors may occur. An unsupported hypothesis may be incorrectly assigned to the positive branch, while a visually supported but weak hypothesis may be prematurely assigned to the residual branch. Motivated by the asymmetric coefficients in contrastive decoding, we define

$$\mathcal{R}_{\text{role}} = (1 + \alpha)\pi_0 \,\text{FPR} + \alpha\pi_1 \,\text{FNR}, \tag{10}$$

where π_0 and π_1 denote the proportions of unsupported and supported hypotheses, respectively, and FPR and FNR denote the corresponding assignment error rates.

Let $(\text{FPR}_0, \text{FNR}_0)$ denote the errors after the initial verification, and $(\text{FPR}_r, \text{FNR}_r)$ denote those after re-verification. The re-verification stage reduces the above risk whenever

$$\alpha\pi_1(\text{FNR}_0 - \text{FNR}_r) > (1 + \alpha)\pi_0(\text{FPR}_r - \text{FPR}_0). \tag{11}$$

This condition describes the trade-off introduced by re-verification. Recovering visually valid but initially weak evidence is beneficial provided that the recovery does not introduce a larger amount of unsupported information into the positive branch. If re-verification reduces the false-negative rate without increasing the false-positive rate, the role assignment risk decreases.

VERA therefore treats insufficient original support as uncertainty rather than immediate negative evidence, allowing weak but valid visual information to be re-examined before negative assignment.

B.3 Safety Property of Residual Suppression

After re-verification, VERA separates visually supported information into M^+ and retains the remaining unresolved component in M^-. Let

$$\Delta_t^- = \ell_t^{-,M^-} - \ell_t^{-,\text{base}}, \tag{12}$$

and define the residual penalty as

$$P_t^- = \text{ReLU}(\Delta_t^-). \tag{13}$$

VERA computes the final logits as

$$\ell_t^{\text{VERA}} = (1 + \alpha)\ell_t^{+,M^+} - \alpha\ell_t^{-,\text{base}} - \alpha P_t^-. \tag{14}$$

Proposition 1. For $\alpha \geq 0$, the direct contribution introduced by the unresolved residual memory M^- is non-positive for every candidate token.

Proof. Since

$$P_t^- = \text{ReLU}(\Delta_t^-) \geq 0, \tag{15}$$

the residual contribution satisfies

$$\Delta \ell_t^{M^-} = -\alpha P_t^- \leq 0. \tag{16}$$

Therefore, M^- can suppress a candidate or leave its logit unchanged, but it cannot directly increase its logit. □

This property is used because M^- represents unresolved information rather than verified positive evidence. Without the one-sided constraint, a decrease in the negative-branch logit caused by residual memory would be subtracted again during contrastive decoding and could be converted into an unintended positive contribution.

B.4 Applicability and Relation to Existing Contrastive Decoding

VERA operates at inference time without updating the parameters of the backbone MLLM. The backbone remains frozen during the training of the VERA-specific components. Its current formulation requires access to contextual hidden states and visual representations because visual hypotheses are rebound to the original visual key-value states before branch assignment. It is therefore directly applicable to open-weight MLLMs whose intermediate representations are accessible, while models that expose only API-level outputs require a different interface.

The effectiveness of VERA is also bounded by the information available to its verification process. Let $\mathcal{G}$ denote the set of visually grounded concepts relevant to the current query and $\mathcal{H}$ the set of generated intermediate hypotheses. The hypothesis-conditioned positive reinforcement of VERA is limited by

$$C_{\mathcal{H}} = \frac{|\mathcal{G} \cap \mathcal{H}|}{|\mathcal{G}|}. \tag{17}$$

VERA can verify or reject a proposed hypothesis but cannot explicitly reinforce a grounded concept that is never exposed in $\mathcal{H}$. Similarly, if the visual encoder fails to preserve the relevant evidence in its visual representation, re-verification cannot recover information that is absent from that representation.

These properties distinguish VERA from previous contrastive decoding methods. VCD Leng et al. (2024) contrasts original and distorted visual inputs, SID Huo et al. (2025) operates on visual tokens, ASCD Wang et al. (2026) intervenes through cross-modal attention, and PND Jiang et al. (2026) strengthens positive visual evidence while constructing a counterfactual negative pathway. VERA instead focuses on whether the information assigned to each branch is grounded in visual evidence.

C Complete Formulation of VERA

This section gives the complete formulation omitted from the main paper for compactness. We describe the generation-time representation of intermediate visual hypotheses, visual-support estimation, re-verification, memory construction, memory injection, one-sided residual suppression, existence-claim calibration, and plausibility-constrained decoding.

C.1 Intermediate Visual Hypotheses and Contextual States

Given the image I and instruction x, the MLLM first produces a temporary set of intermediate visual hypotheses

$$\mathcal{H} = \{h_1, h_2, \ldots, h_U\}, \tag{18}$$

where each h_i describes an object, attribute, or relation considered relevant to the current image and query.

VERA does not reconstruct the representation of h_i by separately encoding its generated text. Instead, it retrieves the contextual state produced during the actual generation process. For each hypothesis, we retain its hidden representation $\mathbf{r}_i$, attention query $\mathbf{q}_i$, key $\mathbf{k}_i$, and value $\mathbf{v}_i$. These representations preserve the image–question context under which the hypothesis was generated.

Algorithm 1 Inference procedure of VERA

Input: Image I, instruction x, MLLM $\mathcal{F}$, re-verification steps R
Output: Next-token distribution ℓ_t^{VERA}
1: Generate intermediate visual hypotheses $\mathcal{H} = \{h_i\}_{i=1}^{U}$
2: Retrieve generation-time contextual states $\{(\mathbf{r}_i, \mathbf{q}_i)\}_{i=1}^{U}$
3: Obtain original visual key-value states $(\mathbf{K}_I, \mathbf{V}_I)$
4: **for** $h_i \in \mathcal{H}$ **do**
5: Rebind $\mathbf{q}_i$ to the original visual states $(\mathbf{K}_I, \mathbf{V}_I)$
6: Estimate the initial visual-support score $s_i^{(0)}$
7: Perform bounded re-verification against the original image
8: Update the visual-support score to s_i
9: **end for**
10: Construct supported memory $\mathcal{M}^+$ and unresolved residual memory $\mathcal{M}^-$
11: Inject $\mathcal{M}^+$ into the positive branch
12: Inject $\mathcal{M}^-$ into the residual branch
13: Retain only the suppressive residual change $\mathbf{P}_t^-$
14: Compute the VERA contrastive logits ℓ_t^{VERA}
15: Apply the plausibility constraint and select the next token

C.2 Visual Support Estimation

To determine whether h_i is supported by the original image, its generation-time query $\mathbf{q}_i$ is used to revisit the original visual key-value representations $\mathbf{K}_I$ and $\mathbf{V}_I$. The visual attention weights are computed as

$$\mathbf{a}_i = \mathrm{Softmax}\left(\frac{\mathbf{q}_i \mathbf{K}_I^\top}{\sqrt{d}}\right). \tag{19}$$

The corresponding visually rebound key, value, and hidden representations are

$$\widetilde{\mathbf{k}}_i = \mathbf{a}_i \mathbf{K}_I, \qquad \widetilde{\mathbf{v}}_i = \mathbf{a}_i \mathbf{V}_I, \qquad \widetilde{\mathbf{h}}_i = \mathrm{FlattenHeads}\left(\widetilde{\mathbf{v}}_i\right). \tag{20}$$

The contextual hypothesis representation and the visually rebound representation are projected by two independent lightweight projectors:

$$\mathbf{z}_i^r = f_r\left(\mathbf{r}_i\right), \qquad \mathbf{z}_i^v = f_v\left(\widetilde{\mathbf{h}}_i\right). \tag{21}$$

In our implementation, both f_r and f_v are independent LayerNorm–Linear–GELU–Linear modules. For a backbone hidden size H, they map $H \to P \to H$, where $P = 512$ is the projector hidden dimension. The two projectors do not share parameters.

The initial visual-support score is estimated from the two representations and their pairwise interactions:

$$s_i^{(0)} = \sigma\left(f_s\left([\mathbf{z}_i^r; \mathbf{z}_i^v; \mathbf{z}_i^r - \mathbf{z}_i^v; \mathbf{z}_i^r \odot \mathbf{z}_i^v]\right)\right). \tag{22}$$

Here f_s is an MLP consisting of LayerNorm–Linear–GELU–Dropout–Linear and maps $4H \to P \to 1$. The score $s_i^{(0)} \in [0, 1]$ denotes the estimated confidence that the intermediate hypothesis is supported by the original visual evidence. It is not interpreted as a physical proportion of the object or region.

The initial unresolved representation is

$$\mathbf{u}_i^{(0)} = \left(1 - s_i^{(0)}\right)\left(\mathbf{z}_i^r - \mathbf{z}_i^v\right). \tag{23}$$

This residual term records the part of the contextual hypothesis that is not explained by the first visual lookup, but it is not immediately treated as negative evidence.

C.3 Re-verification and Memory Construction

A low initial support score does not necessarily indicate that a hypothesis is incorrect. Small objects, occlusion, ambiguous visual regions, or imperfect initial alignment may all lead to weak visual

responses. VERA therefore treats insufficient support as uncertainty rather than direct negative evidence.

For each active hypothesis, VERA performs a bounded re-verification step over the original image. The query used for re-verification is formed from the unresolved hypothesis query, the prompt query $\mathbf{q}_x$, and the initially rebound visual key:

$$\mathbf{q}_i^r = \frac{\left(1 - s_i^{(0)}\right)\mathbf{q}_i + \mathbf{q}_x + s_i^{(0)}\widetilde{\mathbf{k}}_i}{3}. \tag{24}$$

The second visual lookup is then

$$\mathbf{a}_i^r = \mathrm{Softmax}\left(\frac{\mathbf{q}_i^r \mathbf{K}_I^\top}{\sqrt{d}}\right), \qquad \widehat{\mathbf{k}}_i = \mathbf{a}_i^r \mathbf{K}_I, \qquad \widehat{\mathbf{v}}_i = \mathbf{a}_i^r \mathbf{V}_I. \tag{25}$$

Let $\widehat{\mathbf{z}}_i^v = f_v(\mathrm{FlattenHeads}(\widehat{\mathbf{v}}_i))$. The re-verification module predicts a bounded correction in log-odds space:

$$\delta_i = \Delta_{\max} \tanh\left(f_{\mathrm{rv}}\left(\left[\mathbf{u}_i^{(0)}; \widehat{\mathbf{z}}_i^v; \mathbf{u}_i^{(0)} - \widehat{\mathbf{z}}_i^v; \mathbf{u}_i^{(0)} \odot \widehat{\mathbf{z}}_i^v\right]\right)\right), \tag{26}$$

where f_{rv} has the same MLP structure as f_s but uses independent parameters. We use a fixed $\Delta_{\max} = 1.0$ for all backbones. This bound prevents a single re-verification step from arbitrarily overriding the initial evidence estimate.

Instead of replacing the initial support score with an independent second prediction, VERA updates the score in log-odds space:

$$s_i = \sigma\left(\mathrm{logit}\left(s_i^{(0)}\right) + \delta_i\right). \tag{27}$$

The update can revise the first estimate in either direction. A weakly supported hypothesis may recover support when additional visual evidence is found, while an initially overestimated hypothesis may also be demoted.

After re-verification, VERA constructs two complementary memories. For each decoder layer, the supported memory uses the re-verified visual evidence:

$$m_i^+ = \mathrm{LN}\left(\widehat{\mathbf{z}}_i^v\right), \qquad \mathbf{K}_i^+ = \widehat{\mathbf{k}}_i, \qquad \mathbf{V}_i^+ = \widehat{\mathbf{v}}_i, \qquad w_i^+ = s_i. \tag{28}$$

The unresolved residual memory keeps the residual value component that is not explained by the re-verified visual lookup:

$$m_i^- = \mathrm{LN}\left(\mathrm{FlattenHeads}\left(\mathbf{v}_i - \widehat{\mathbf{v}}_i\right)\right), \qquad \mathbf{K}_i^- = \widehat{\mathbf{k}}_i, \qquad \mathbf{V}_i^- = \mathbf{v}_i - \widehat{\mathbf{v}}_i, \qquad w_i^- = 1 - s_i. \tag{29}$$

The memory sets are therefore

$$\mathcal{M}^+ = \left\{\left(\mathbf{K}_i^+, \mathbf{V}_i^+, w_i^+\right)\right\}_{i=1}^U, \qquad \mathcal{M}^- = \left\{\left(\mathbf{K}_i^-, \mathbf{V}_i^-, w_i^-\right)\right\}_{i=1}^U. \tag{30}$$

The decomposition is soft: all active hypotheses can contribute to both branches through complementary confidence weights. It does not treat a low initial visual response as immediate negative evidence. Only information that remains unresolved after re-verification contributes to the residual memory.

C.4 Memory Injection into Contrastive Branches

After verification and re-verification, the two memory components are incorporated into their corresponding contrastive branches. VERA uses the same injection interface for Qwen2.5-VL, LLaVA-1.5, and LLaVA-NeXT: memory is appended to the key-value states of the selected decoder attention layers. In our Qwen2.5-VL implementation, all decoder layers wrapped by the contrastive attention module are used.

For layer ℓ, the positive branch receives the supported memory:

$$\mathbf{K}_\ell^{+,M^+} = \left[\mathbf{K}_\ell^+; \mathbf{K}_\ell(\mathcal{M}^+)\right], \qquad \mathbf{V}_\ell^{+,M^+} = \left[\mathbf{V}_\ell^+; \mathbf{V}_\ell(\mathcal{M}^+)\right]. \tag{31}$$

The residual branch analogously receives the unresolved residual memory:

$$\mathbf{K}_{\ell}^{-,M^-} = \left[\mathbf{K}_{\ell}^{-}; \mathbf{K}_{\ell}(\mathcal{M}^-)\right], \qquad \mathbf{V}_{\ell}^{-,M^-} = \left[\mathbf{V}_{\ell}^{-}; \mathbf{V}_{\ell}(\mathcal{M}^-)\right]. \tag{32}$$

The memory confidence is applied as an additive attention prior. For a memory unit with branch weight $w_i^{\pm}$ and memory scale γ, the attention logit to that memory token is shifted by

$$b_i^{\pm} = \log\left(\mathrm{clip}\left(\gamma w_i^{\pm}, 10^{-6}, 1\right)\right). \tag{33}$$

The formal configuration uses $\gamma = 0.05$. The appended memory tokens are visible to the current decoding frontier. During prefill, earlier prompt positions are prevented from attending to the appended memory, so the original prompt and image states remain unchanged.

The memory keys are produced from contextual visual key states and therefore already live in the backbone's visual coordinate system. We do not assign new positional ids to the appended memory tokens. For the residual memory, the key uses the re-verified visual key $\widehat{\mathbf{k}}_i$, while the value stores the residual $\mathbf{v}_i - \widehat{\mathbf{v}}_i$; this avoids subtracting keys from incompatible rotary-position frames.

Let $\boldsymbol{\ell}_t^{+,M^+}$ denote the logits of the positive branch with supported memory. We define $\boldsymbol{\ell}_t^{-,\mathrm{base}}$ as the logits of the base residual branch under the same image, instruction, backbone, visual perturbation, and contrastive attention configuration, but without injecting $\mathcal{M}^-$. The logits $\boldsymbol{\ell}_t^{-,M^-}$ are obtained by keeping all components of this base residual branch unchanged and injecting only $\mathcal{M}^-$. Thus, $\boldsymbol{\ell}_t^{-,M^-} - \boldsymbol{\ell}_t^{-,\mathrm{base}}$ isolates the logit change attributable to unresolved residual memory.

C.5 One-sided Residual Suppression

Directly replacing the negative branch with $\boldsymbol{\ell}_t^{-,M^-}$ can introduce an undesirable effect. Since the negative branch is subtracted during contrastive decoding, a decrease in its logit caused by residual memory would be subtracted again and could consequently become a positive contribution.

We first compute the residual-induced change

$$\Delta_t^- = \boldsymbol{\ell}_t^{-,M^-} - \boldsymbol{\ell}_t^{-,\mathrm{base}}. \tag{34}$$

VERA retains only the suppressive part:

$$\mathbf{P}_t^- = \mathrm{ReLU}\left(\Delta_t^-\right). \tag{35}$$

The final VERA logits are

$$\boldsymbol{\ell}_t^{\mathrm{VERA}} = (1+\alpha)\boldsymbol{\ell}_t^{+,M^+} - \alpha\boldsymbol{\ell}_t^{-,\mathrm{base}} - \alpha\mathbf{P}_t^-. \tag{36}$$

For $\alpha \geq 0$ and $\mathbf{P}_t^- \geq 0$, the direct logit contribution introduced by the residual memory satisfies

$$\Delta\boldsymbol{\ell}_{M^-} = -\alpha\mathbf{P}_t^- \leq 0. \tag{37}$$

The unresolved residual memory can therefore suppress a candidate or leave its logit unchanged, but it cannot directly increase the candidate logit through contrastive subtraction.

C.6 Existence-Claim Calibration

For binary existence questions, VERA optionally applies a lightweight claim-level calibration after the one-sided residual correction. This step is used only when a target existence claim can be extracted from the question. Let c denote the target claim and let s_c and r_c be the matched support and residual scores from the memory bank. We define the claim evidence score as

$$e_c = s_c - r_c. \tag{38}$$

If the target claim is found, the support score is above the threshold τ, and $e_c > 0$, VERA releases positive evidence to the "yes" tokens:

$$\rho_c = \lambda e_c \cdot \mathbf{1}\left[s_c \geq \tau\right] \cdot \mathbf{1}\left[e_c > 0\right]. \tag{39}$$

Here τ is the existence-claim support threshold and λ is the evidence calibration coefficient. In our default setting, $\tau = 0.50$ and $\lambda = 4.0$ for this claim-calibration path. This calibration does not define the general visual-support estimator; it is only applied to parsed binary existence claims.

C.7 Plausibility-Constrained Decoding

Following standard contrastive decoding, VERA additionally retains a plausibility constraint based on the positive branch. At decoding step t, candidate token j is retained only when its positive-branch logit satisfies

$$\boldsymbol{\ell}_t^{+,M^+}(j) \geq \log \beta + \max_k \boldsymbol{\ell}_t^{+,M^+}(k), \tag{40}$$

where β controls the plausibility threshold.

Tokens outside this candidate set are excluded before sampling or greedy selection. The next token is then selected according to $\boldsymbol{\ell}_t^{\mathrm{VERA}}$ over the remaining candidate set.

D Detailed Experimental Settings

D.1 Model Configurations

To evaluate the generalizability of VERA, we conduct experiments on three representative MLLMs: LLaVA-1.5 Liu et al. (2023), Qwen2.5-VL Bai et al. (2025), and LLaVA-NeXT Liu et al. (2024c). These models cover different model families and visual representation designs.

LLaVA-1.5 adopts a CLIP vision encoder and an MLP projector to connect visual representations with the language model. Qwen2.5-VL represents a more recent MLLM family with enhanced visual recognition, object localization, and multimodal understanding capabilities. LLaVA-NeXT extends the LLaVA architecture with higher-resolution visual inputs and the AnyRes strategy, preserving finer visual details and increasing the amount of visual information available to the language model. All selected models are of comparable parameter scale, covering both LLaVA-style and Qwen-style architectures.

D.2 Datasets

We consider two groups of benchmarks. COCO, POPE, and CHAIR are used for object hallucination evaluation, with POPE focusing on object presence judgment and CHAIR measuring hallucinated objects in open-ended image captioning. We further evaluate general multimodal capabilities on MME, HallusionBench, SEED-Bench, SimpleVQA, and GQA.

COCO. COCO is a large-scale visual dataset collected from real-world scenes and is widely used in object detection, instance segmentation, and image captioning. It contains approximately 330,000 images, more than 200,000 of which are manually annotated. The annotations cover 80 common object categories and include object instances, segmentation masks, and human-written captions.

POPE. POPE Li et al. (2023) evaluates object hallucination by formulating object presence as a binary Yes/No question. It defines three evaluation settings: Random, Popular, and Adversarial. Random negatives are sampled from object categories absent from the current image, while Popular favors frequently occurring categories that are not present. Adversarial selects absent objects with strong co-occurrence relationships with objects appearing in the image. Under the standard COCO setting, 500 images are used and each setting contains 3,000 questions with balanced positive and negative samples, resulting in 9,000 questions in total. Accuracy, Precision, Recall, and F1 are commonly reported, while we mainly use Accuracy and F1 in our experiments.

CHAIR. CHAIR Rohrbach et al. (2018) measures object hallucination in open-ended image captioning by comparing generated captions with the object annotations and human-written captions provided by COCO. Two complementary metrics are reported:

$$\mathrm{CHAIR}_s = \frac{N_{\mathrm{sent}}^{\mathrm{hall}}}{N_{\mathrm{sent}}}, \qquad \mathrm{CHAIR}_i = \frac{N_{\mathrm{obj}}^{\mathrm{hall}}}{N_{\mathrm{obj}}^{\mathrm{gen}}}. \tag{41}$$

Here, $N_{\mathrm{sent}}^{\mathrm{hall}}$ denotes the number of generated captions containing at least one hallucinated object, and N_{sent} is the total number of generated captions. $N_{\mathrm{obj}}^{\mathrm{hall}}$ counts hallucinated object instances, whereas $N_{\mathrm{obj}}^{\mathrm{gen}}$ counts all object instances mentioned in the generated captions. Lower values of both

metrics indicate fewer hallucinated objects. Following the commonly used protocol, we evaluate open-ended caption generation on 500 COCO images and report both CHAIR_s and CHAIR_i.

MME. MME Fu et al. (2025) evaluates the perception and cognition abilities of multimodal large language models across 14 subtasks. The perception tasks cover object existence, counting, position, color, scene recognition, and OCR, while the cognition tasks include commonsense reasoning, numerical calculation, text translation, and code reasoning. Its concise question-answer format allows us to examine whether hallucination mitigation affects the model's ability to recognize visual content and perform basic multimodal reasoning.

HallusionBench. Designed to distinguish language hallucinations from visual illusions, HallusionBench Guan et al. (2024a) contains 346 images and 1,129 questions that require careful interpretation of visual context. The benchmark includes paired questions and controlled visual settings, allowing inconsistencies between model predictions and image evidence to be examined. We use HallusionBench to assess how VERA handles visual ambiguity and whether its predictions remain grounded in the given image.

SEED-Bench. SEED-Bench Li et al. (2024) uses multiple-choice questions to assess the visual understanding capabilities of multimodal models. Its evaluation dimensions cover object recognition, attribute understanding, spatial relations, and visual reasoning, among other perceptual abilities. The multiple-choice format provides a consistent basis for comparing decoding methods without relying on an external language model to judge free-form responses.

SimpleVQA. Unlike benchmarks that primarily test object recognition, SimpleVQA Cheng et al. (2025) focuses on short-answer factual questions grounded in visual inputs. It contains 2,025 question-answer pairs covering topics such as natural science, history, art, technology, and everyday life. Answering these questions often requires combining information from the image with relevant factual knowledge. We include SimpleVQA to examine the effect of VERA on factual visual question answering beyond object-existence judgments.

GQA. Built on images and scene graphs from Visual Genome, GQA Hudson & Manning (2019) targets compositional visual reasoning in real-world scenes. The full dataset contains approximately 113,000 images and 22 million automatically generated questions involving object properties, spatial relations, relation tracking, logical operations, and comparison. Each image is associated with a structured scene graph, while questions are paired with semantic representations and functional programs that describe the reasoning steps required to obtain the answer.

D.3 Baseline Methods

To evaluate VERA, we compare it with several representative state-of-the-art methods for mitigating object hallucination, including VCD Leng et al. (2024), SID Huo et al. (2025), ASCD Wang et al. (2026), PND Jiang et al. (2026), CausalLens Ji et al. (2026), and HulluEdit Lin et al. (2026). These methods cover contrastive decoding, visual token manipulation, cross-modal attention intervention, dual-branch construction, and representation-level correction.

D.4 Implementation Details

All methods are evaluated under a unified inference setting. Unless otherwise specified, greedy decoding is used throughout the experiments. For contrastive decoding, we set the contrastive coefficient to $\alpha = 1.0$ and the plausibility threshold to $\beta = 0.1$. For VERA, the existence-claim support threshold is fixed to $\tau = 0.50$, and the evidence calibration coefficient is fixed to $\lambda = 4.0$ when the claim-calibration path is applicable.

VERA does not update or finetune the backbone MLLM. A lightweight verification/projector module is learned offline, while all hallucination-mitigation operations, including intermediate hypothesis generation, visual verification, re-verification, memory construction, branch injection, and contrastive correction, are performed at inference time. During projector training, the parameters of the language model, vision encoder, and multimodal backbone remain frozen.

Training of VERA-specific modules. Training samples are constructed only from images in the COCO training split and are strictly separated from the evaluation samples used in POPE, CHAIR,

and the general multimodal benchmarks. We use 10,000 COCO training images for projector training and 2,000 held-out COCO training images for validation. For each image, we construct a paired visual verification prompt containing one COCO object category annotated as present and one COCO category annotated as absent. The model first generates temporary visual hypotheses, and training-only anchor hypotheses are appended when necessary to ensure that both the positive and negative object concepts are represented.

Support labels are obtained from COCO instance annotations and captions. Object hypotheses whose category is present in the image receive a positive support label, while object hypotheses whose category is absent receive a negative label. Attribute and relation hypotheses receive a negative label if they contain an absent object argument, and receive a weak positive label only when the same object and attribute/relation cue are corroborated by a COCO caption. otherwise they are left unlabeled. Only labeled hypotheses contribute to the projector loss.

The VERA-specific training objective is

$$\begin{aligned}\mathcal{L}_{\mathrm{VERA}} = \mathcal{L}_{\mathrm{gate}} + 0.5\,\mathcal{L}_{\mathrm{init}} + \mathcal{L}_{\mathrm{align}} + 0.5\,\mathcal{L}_{\mathrm{neg}} \\ + 0.5\,\mathcal{L}_{\mathrm{rank}} + 0.5\,\mathcal{L}_{\mathrm{hard}} + 0.5\,\mathcal{L}_{\text{neg-rv}} + 0.5\,\mathcal{L}_{\text{pos-demote}}.\end{aligned} \tag{42}$$

Here $\mathcal{L}_{\mathrm{gate}}$ is a class-balanced binary cross-entropy loss on the final support score, and $\mathcal{L}_{\mathrm{init}}$ applies the same supervision to the first-pass support score. $\mathcal{L}_{\mathrm{align}}$ encourages positively supported hypotheses to align with rebound visual evidence, while $\mathcal{L}_{\mathrm{neg}}$ separates unsupported hypotheses from visual evidence. $\mathcal{L}_{\mathrm{rank}}$ enforces a margin between positive and negative support scores. $\mathcal{L}_{\mathrm{hard}}$ penalizes residual representations of unsupported hypotheses that remain closer to visual evidence than to their original contextual value. Finally, $\mathcal{L}_{\text{neg-rv}}$ discourages unsupported hypotheses from being promoted during re-verification, and $\mathcal{L}_{\text{pos-demote}}$ discourages supported hypotheses from being demoted.

We optimize only the VERA projector/support module using AdamW with learning rate 1×10^{-4}, weight decay 0.01, batch size 1, and one training epoch. Gradients are clipped to norm 1.0. The projector hidden dimension is 512, and for Qwen2.5-VL 7B the trainable VERA module contains approximately 22.1M parameters. Unless otherwise stated, each backbone uses its corresponding frozen backbone with a VERA projector configured for that backbone.

For each backbone, all compared methods use the same evaluation samples, input prompts, generation-length limits, and evaluation scripts. POPE follows the standard binary-answer evaluation protocol, where responses that cannot be parsed as valid Yes or No answers are counted as incorrect. For CHAIR, CHAIR_s and CHAIR_i are computed using the same caption preprocessing procedure for all methods. For MME, HallusionBench, SEED-Bench, SimpleVQA, and GQA, we use the same evaluation samples and evaluation scripts for all compared methods. All experiments are conducted on NVIDIA RTX 4090 and RTX PRO 6000 GPUs.

Table 3: Default implementation settings of VERA.

Parameter	**Value**
Contrastive coefficient α	1.0
Plausibility threshold β	0.1
Existence-claim threshold τ	0.50
Evidence calibration coefficient λ	4.0
Re-verification bound $\Delta_{\max}$	1.0
Projector hidden dimension	512
Maximum memory units	24
Number of re-verification passes	1
Memory attention prior scale γ	0.05
Maximum temporary hypothesis length	64
Maximum generation length	128
Decoding strategy	Greedy

Table 3 summarizes the default configuration used throughout the experiments. Unless otherwise stated, $\alpha = 1.0$, $\beta = 0.1$, $\tau = 0.50$, $\lambda = 4.0$, $\Delta_{\max} = 1.0$, and $\gamma = 0.05$ are fixed. Temporary

visual hypotheses are limited to 64 tokens, while the maximum final generation length is 128 tokens. Greedy decoding is used as the default decoding strategy for all compared methods.

E Complete Experimental Results

The main paper reports the compact comparison on CHAIR and POPE. This section provides the complete backbone-specific results. For each backbone, we separately report CHAIR and POPE results, including Random, Popular, and Adversarial performance, average Accuracy and F1, Precision, Recall, and average generation length.

E.1 Qwen2.5-VL

Table 4: CHAIR results on Qwen2.5-VL. Δ denotes absolute reduction over Vanilla.

Method	$\text{CHAIR}_s \downarrow$	Δ	$\text{CHAIR}_i \downarrow$	Δ
Vanilla	35.66	–	10.14	–
VCD	25.33	+10.33	9.02	+1.12
SID	22.00	+13.66	8.63	+1.51
ASCD	15.33	+20.33	8.45	+1.69
PND	33.00	+2.66	8.58	+1.56
CausalLens	33.33	+2.33	9.09	+1.05
HulluEdit	29.33	+6.33	7.96	+2.18
VERA	**15.00**	**+20.66**	**7.31**	**+2.83**

Table 5: POPE and other metrics on Qwen2.5-VL.

Metric / Method	Vanilla	VCD	SID	ASCD	PND	CausalLens	HulluEdit	**VERA**
Random Acc	81.86	83.76	85.53	84.86	85.26	82.40	84.60	**88.86**
Random F1	78.17	80.66	83.16	82.25	82.74	78.67	81.82	**87.63**
Popular Acc	81.53	83.23	85.00	84.53	84.73	82.06	84.06	**87.93**
Popular F1	77.85	80.15	82.65	81.93	82.23	78.35	81.31	**86.73**
Adv. Acc	80.60	82.96	84.20	83.93	84.06	81.60	83.53	**86.86**
Adv. F1	76.99	79.90	81.89	81.36	81.60	77.91	80.80	**85.73**
Avg Acc	81.33	83.32	84.91	84.44	84.68	82.02	84.06	**87.88**
Avg F1	77.67	80.24	82.57	81.85	82.19	78.31	81.31	**86.70**
Prec.	96.65	98.42	97.79	98.24	98.21	**98.69**	98.32	96.20
Recall	64.93	67.73	71.45	70.15	70.67	64.91	69.32	**78.91**
Len.	94.20	93.70	94.00	93.50	**94.40**	93.80	93.60	94.10

Tables 4 and 5 report the complete results on Qwen2.5-VL. VERA reduces CHAIR_s from 35.66 under Vanilla to 15.00 and CHAIR_i from 10.14 to 7.31, corresponding to absolute reductions of 20.66 and 2.83.

On POPE, VERA obtains the highest Accuracy and F1 among the compared methods under Random, Popular, and Adversarial sampling. Its average Accuracy and F1 are 87.88 and 86.70, compared with 81.33 and 77.67 for Vanilla. Recall increases from 64.93 to 78.91, while the average generation length remains close to Vanilla (94.10 versus 94.20).

E.2 LLaVA-1.5

Tables 6 and 7 show the complete LLaVA-1.5 results. VERA reduces CHAIR_s from 66.66 to 50.00 and CHAIR_i from 26.20 to 13.75, corresponding to absolute reductions of 16.66 and 12.45.

For POPE, VERA reaches an average Accuracy of 86.33 and an average F1 of 86.88. PND obtains slightly higher Accuracy under the Popular setting, while VERA gives the highest Random Accu-

Table 6: CHAIR results on LLaVA-1.5. Δ denotes absolute reduction over Vanilla.

Method	$\text{CHAIR}_s \downarrow$	Δ	$\text{CHAIR}_i \downarrow$	Δ
Vanilla	66.66	–	26.20	–
VCD	55.00	+11.66	16.57	+9.63
SID	54.33	+12.33	15.28	+10.92
ASCD	55.00	+11.66	14.26	+11.94
PND	54.66	+12.00	15.25	+10.95
CausalLens	53.32	+13.34	13.78	+12.42
HulluEdit	56.66	+10.00	15.61	+10.59
VERA	**50.00**	**+16.66**	**13.75**	**+12.45**

Table 7: POPE and other metrics on LLaVA-1.5.

Metric / Method	Vanilla	VCD	SID	ASCD	PND	CausalLens	HulluEdit	**VERA**
Random Acc	80.40	87.93	89.83	90.00	89.93	90.20	90.06	**91.00**
Random F1	81.82	87.72	89.56	89.95	89.70	90.24	89.81	**90.52**
Popular Acc	77.20	85.20	86.26	86.73	**87.33**	86.26	87.20	87.00
Popular F1	79.47	85.35	86.40	87.10	87.38	86.84	87.25	**87.45**
Adv. Acc	71.33	79.13	80.40	79.26	80.13	78.80	80.13	**81.00**
Adv. F1	75.48	80.52	81.65	81.20	81.53	81.04	81.51	**82.68**
Avg Acc	76.31	84.09	85.50	85.33	85.80	85.09	85.80	**86.33**
Avg F1	78.92	84.53	85.87	86.08	86.20	86.04	86.19	**86.88**
Prec.	71.50	83.09	84.76	83.13	85.00	82.16	85.08	**85.37**
Recall	88.24	86.24	87.25	89.56	87.71	**90.63**	87.60	89.07
Len.	93.85	93.90	**94.78**	93.25	92.79	88.68	92.87	93.74

racy/F1, Adversarial Accuracy/F1, and average Accuracy and F1 in the table. Its average generation length is 93.74, close to the Vanilla value of 93.85.

E.3 LLaVA-NeXT

Table 8: CHAIR results on LLaVA-NeXT. Δ denotes absolute reduction over Vanilla.

Method	$\text{CHAIR}_s \downarrow$	Δ	$\text{CHAIR}_i \downarrow$	Δ
Vanilla	38.50	–	14.27	–
VCD	33.50	+5.00	10.83	+3.44
SID	28.50	+10.00	**9.42**	+4.85
ASCD	29.00	+9.50	9.71	+4.56
PND	31.00	+7.50	10.12	+4.15
CausalLens	34.00	+4.50	9.05	**+5.22**
HulluEdit	28.50	+10.00	9.98	+4.29
VERA	**27.50**	**+11.00**	9.70	+4.57

Tables 8 and 9 provide the complete LLaVA-NeXT results. VERA achieves $\text{CHAIR}_s = 27.50$, reducing it by 11.00 points relative to Vanilla. For CHAIR_i, other compared methods obtain lower values, while VERA records 9.70 compared with 14.27 for Vanilla.

On POPE, VERA achieves the highest average Accuracy and F1, reaching 90.11 and 89.57, respectively. It also obtains the highest Precision of 94.63. The highest Recall is obtained by ASCD. The average generation length of VERA is 92.50, compared with 92.60 for Vanilla.

Table 9: POPE and other metrics on LLaVA-NeXT.

Metric / Method	Vanilla	VCD	SID	ASCD	PND	CausalLens	HulluEdit	**VERA**
Random Acc	86.50	90.33	91.17	90.66	**91.50**	90.67	91.17	91.33
Random F1	86.20	89.68	90.59	90.78	90.94	90.54	90.59	**90.97**
Popular Acc	85.83	88.67	90.17	88.16	89.67	87.17	90.33	**90.50**
Popular F1	85.62	88.11	89.63	88.60	89.42	87.44	89.79	**89.98**
Adv. Acc	80.33	85.50	88.00	81.83	88.00	82.67	88.00	**88.50**
Adv. F1	81.09	85.28	87.63	83.50	87.67	83.75	87.63	**87.76**
Avg Acc	84.22	88.17	89.78	86.88	89.72	86.84	89.83	**90.11**
Avg F1	84.30	87.69	89.28	87.63	89.34	87.24	89.34	**89.57**
Prec.	84.38	91.81	94.07	83.84	93.05	85.43	94.16	**94.63**
Recall	84.35	84.00	84.99	**91.98**	85.98	89.31	85.02	85.04
Len.	**92.60**	92.10	92.40	91.90	92.30	91.70	92.00	92.50

F Additional Analyses

This section provides analyses complementary to the main experiments. We first examine how the visual-support score differs between GT Yes and GT No samples and how evidence is distributed between the two branches. We then analyze one-sided residual suppression and evaluate VERA under weak visual evidence. Additional experiments consider different decoding configurations, hyperparameter settings, and stage-wise inference latency.

F.1 Visual Support and Branch Assignment Analysis

To examine how verified visual evidence is distributed between the two branches, we analyze the statistics of M^+ and M^- separately. Unlike traditional contrastive decoding Leng et al. (2024); Wang et al. (2026); Jiang et al. (2026), which assigns branch roles according to predefined visual enhancement or degradation operations, VERA reassigns intermediate visual hypotheses according to the visual verification results. Information with sufficient visual support is retained in M^+, while the remaining unsupported component after re-verification is retained in M^-.

Table 10: Analysis of visual evidence guided branch assignment in VERA.

Metric	**GT Yes**	**GT No**
Number of samples	900	900
Visual support score	0.870	0.468
Residual score	0.130	0.532
High confidence positive support (%)	88.0	10.2
Support AUROC	0.929	
One-sided violations	0 / 1800	

Table 10 reports the visual-support statistics for 900 GT Yes and 900 GT No samples. The mean visual-support score is 0.870 for GT Yes and 0.468 for GT No, whereas the corresponding residual scores are 0.130 and 0.532. High-confidence positive support is observed in 88.0% of GT Yes samples and 10.2% of GT No samples. The visual-support score yields an AUROC of 0.929 on this evaluation set. Under the one-sided residual formulation, no direct positive residual contribution is recorded among the 1,800 samples.

F.2 One-sided Residual Suppression Analysis

The residual branch represents unresolved information rather than verified positive evidence. We therefore examine the behavior that would arise if the signed residual change were directly used in contrastive subtraction.

Table 11: Analysis of potential positive contributions induced by signed residual subtraction. The one-sided formulation eliminates direct positive contributions from the residual term in the recorded samples.

GT label	Samples	Promote Yes (%)	Promote No (%)
Yes	900	47.00	45.00
No	945	46.46	43.17
Overall	1,845	46.72	44.07

"Promote Yes/No" denotes the sample-level potential positive-contribution flags recorded under signed residual subtraction. The two flags are not mutually exclusive. Violations count direct positive residual contributions under the one-sided formulation, not incorrect final answers.

As shown in Table 11, signed residual subtraction produces potential positive-contribution flags for both Yes and No candidates. Across all 1,845 samples, the corresponding proportions are 46.72% and 44.07%. The two statistics are not mutually exclusive and therefore should not be added together. The statistics describe the direct effect of the residual term and should not be interpreted as final-answer errors.

F.3 Hallucination Mitigation under Weak Visual Evidence

Existing research Sun et al. (2025); Li et al. (2025); Jiang et al. (2026) indicates that the visual salience and scale of a target influence how MLLMs use visual evidence. When a target is small, has weak local visual features, or is easily obscured by prominent surrounding areas, the model may overlook visual information and produce an incorrect judgment. For hallucination mitigation methods, such samples are difficult because a low visual response does not necessarily imply that the object is absent.

To evaluate VERA under weak visual evidence, we use VisDrone-Dataset Zhu et al. (2021), which contains more than 2.6 million manually annotated bounding boxes for targets such as pedestrians, cars, bicycles, and tricycles. We compare VERA with Vanilla, VCD Leng et al. (2024), ASCD Wang et al. (2026), and PND Jiang et al. (2026) on small-target object-existence questions.

Table 12: Results on the VisDrone small-object existence benchmark.

Method	Acc. ↑	Prec. ↑	Recall ↑	F1 ↑
Vanilla	64.50	60.14	86.00	70.78
VCD	63.00	59.42	82.00	68.91
ASCD	58.50	55.28	**89.00**	68.20
PND	63.00	59.15	84.00	69.42
VERA	**69.50**	**67.26**	76.00	**71.36**

Table 12 shows that VERA obtains an Accuracy of 69.50, a Precision of 67.26, and an F1 score of 71.36. Its Recall is 76.00, lower than Vanilla, VCD, ASCD, and PND in this experiment. The results therefore reflect a different precision-recall balance under small-object conditions rather than a uniform improvement across all four metrics.

F.4 Robustness across Decoding Settings

We further evaluate VERA under different decoding configurations, including top-p, top-k, and temperature sampling. Experiments are conducted on Qwen2.5-VL and LLaVA-1.5. For each setting, VERA is compared with the corresponding Origin result obtained under the same decoding configuration. We report the change in POPE average F1 and CHAIR_i, followed by the win rate over all nine decoding settings.

Figure 6 reports the POPE average F1 change under different top-p, top-k, and temperature settings. For Qwen2.5-VL, VERA improves F1 in eight of the nine settings. For LLaVA-1.5, positive gains

are also observed in eight settings, while the magnitude of the improvement varies with the decoding configuration.

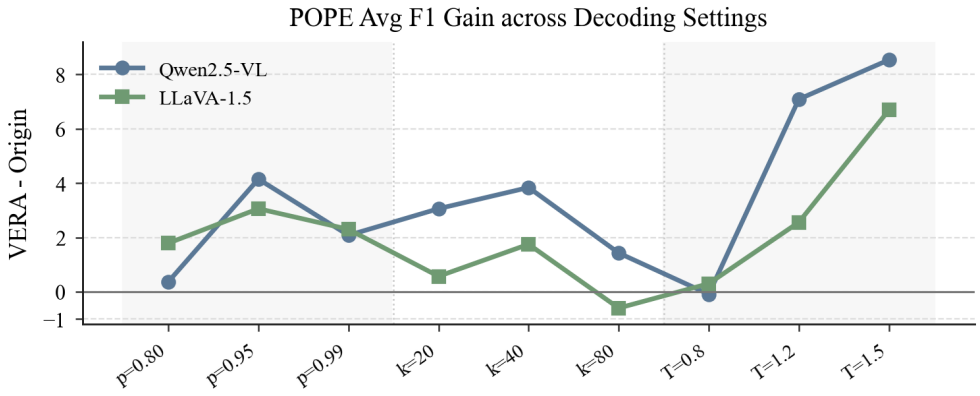

Figure 6: POPE average F1 gain of VERA over Origin under different decoding settings. The vertical axis reports $\mathrm{F1}_{\mathrm{VERA}} - \mathrm{F1}_{\mathrm{Origin}}$. Positive values indicate that VERA achieves a higher average F1 under the corresponding decoding configuration.

Figure 7 presents the corresponding change in CHAIR_i. VERA reduces CHAIR_i in eight of the nine settings on Qwen2.5-VL and in seven settings on LLaVA-1.5. The magnitude of the reduction varies across decoding strategies.

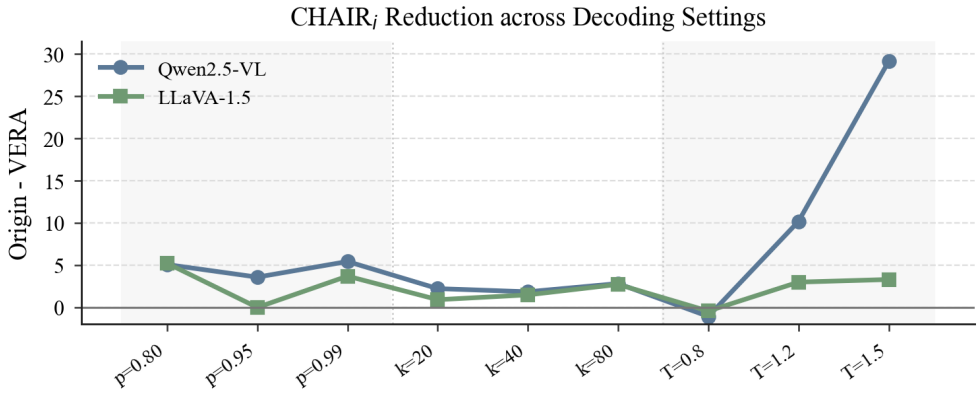

Figure 7: Reduction in CHAIR_i achieved by VERA relative to Origin under different decoding settings. The vertical axis reports $\mathrm{CHAIR}_{i,\mathrm{Origin}} - \mathrm{CHAIR}_{i,\mathrm{VERA}}$. Positive values indicate lower object-level hallucination after applying VERA.

Figure 8 summarizes the nine decoding configurations. On Qwen2.5-VL, VERA records wins in $8/9$ settings for POPE F1, $9/9$ for CHAIR_s, and $8/9$ for CHAIR_i; both F1 improvement and CHAIR_i reduction occur in $8/9$ settings. On LLaVA-1.5, the corresponding counts are $8/9$, $7/9$, $7/9$, and $6/9$.

These results show that the effect of VERA is not restricted to the default greedy decoding configuration. The magnitude of the change depends on the decoding setting and model backbone, while improvements are observed in most of the tested configurations.

F.5 Hyperparameter Sensitivity

To evaluate VERA's sensitivity to its main hyperparameters, we vary the visual-support threshold τ and evidence calibration coefficient λ on POPE. Table 13 reports the corresponding Random, Popular, Adversarial, and average F1 scores.

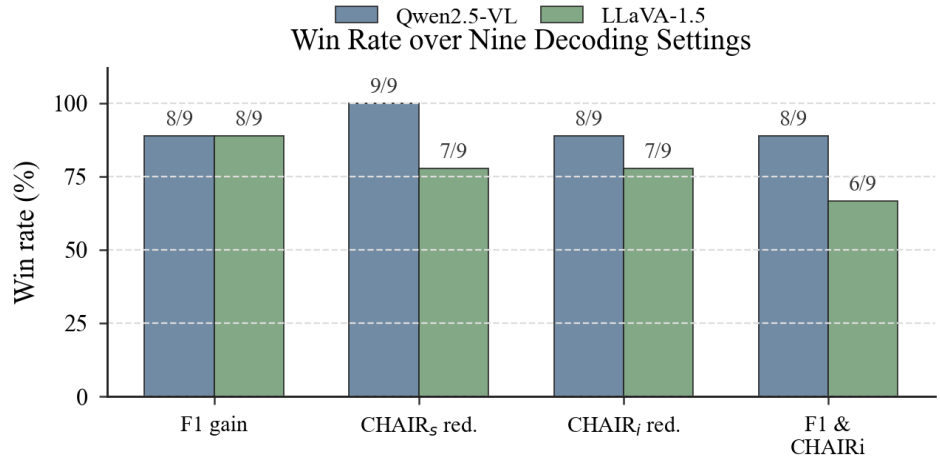

Figure 8: Win rate of VERA over Origin across nine decoding settings. "F1 gain" denotes settings in which VERA achieves a higher POPE average F1, while "CHAIR$_s$ red." and "CHAIR$_i$ red." denote settings in which VERA obtains lower CHAIR$_s$ and CHAIR$_i$, respectively. "F1 & CHAIR$_i$" counts settings where both a positive F1 gain and a reduction in CHAIR$_i$ are obtained.

Table 13: Hyperparameter robustness of VERA on POPE.

Setting	Random F1 ↑	Popular F1 ↑	Adv. F1 ↑	Avg. F1 ↑
$\tau \in [0.40, 0.60], \lambda = 4$	90.20	89.36	86.32	88.62
$\tau = 0.70, \lambda = 4$	89.11	88.17	86.32	87.87
$\lambda = 1, \tau = 0.50$	89.11	82.76	87.23	86.37
$\lambda = 3, \tau = 0.50$	90.20	85.71	86.32	87.41
$\lambda = 4, \tau = 0.50$	90.20	89.36	86.32	88.62
$\lambda = 5, \tau = 0.50$	90.20	**91.67**	**86.87**	**89.58**
$\lambda = 6, \tau = 0.50$	90.20	90.72	**86.87**	89.26

With $\lambda = 4.0$, the results remain unchanged for $\tau \in [0.40, 0.60]$, with an average F1 of 88.62, while increasing τ to 0.70 reduces the average F1 to 87.87. With $\tau = 0.50$, the average F1 increases from 86.37 at $\lambda = 1$ to 89.58 at $\lambda = 5$, followed by a small decrease to 89.26 at $\lambda = 6$. The variation is more pronounced on the Popular split than on Random and Adversarial.

F.6 Inference Efficiency

The end-to-end efficiency comparison is reported in the main paper. Here we further separate the latency of VERA into its main inference stages.

Table 14: Stage-wise latency breakdown of VERA on Qwen2.5-VL 7B.

Stage	Latency (ms/sample) ↓
Initial Hypothesis Generation	1388.30
Visual Evidence Verification	100.22
Re-verification & Memory Construction	4.68
Contrastive Decoding	2977.17
Other Overhead	224.33
Total End-to-End	4694.70

Table 14 shows that Contrastive Decoding accounts for 2977.17 ms per sample, followed by Initial Hypothesis Generation at 1388.30 ms. Visual Evidence Verification requires 100.22 ms, while Re-verification & Memory Construction accounts for 4.68 ms in the measured setting. Other overhead contributes 224.33 ms, giving a total end-to-end latency of 4694.70 ms per sample.

G CASE STUDIES

We provide qualitative examples comparing the predictions produced by VERA and the competing hallucination mitigation methods. The figures report the original image, question, ground-truth answer, and responses generated by the different decoding methods. These cases are included to illustrate the prediction behavior behind the quantitative results.

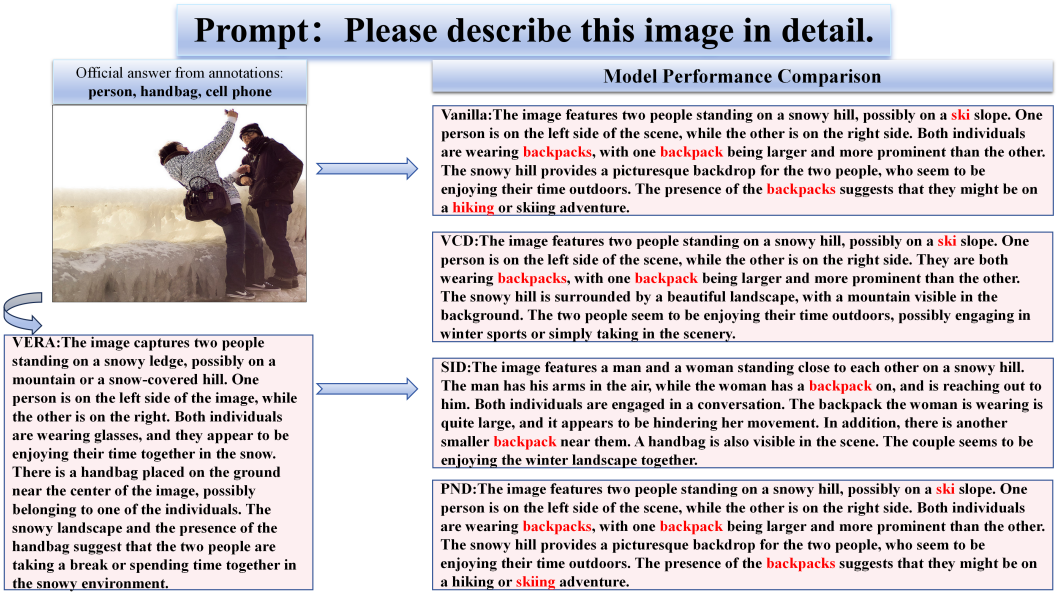

Figure 9: Case study on CHAIR. VERA reduces object hallucinations in long-form image descriptions while preserving comparable caption length. Hallucinated objects identified by the CHAIR evaluator are marked in red.

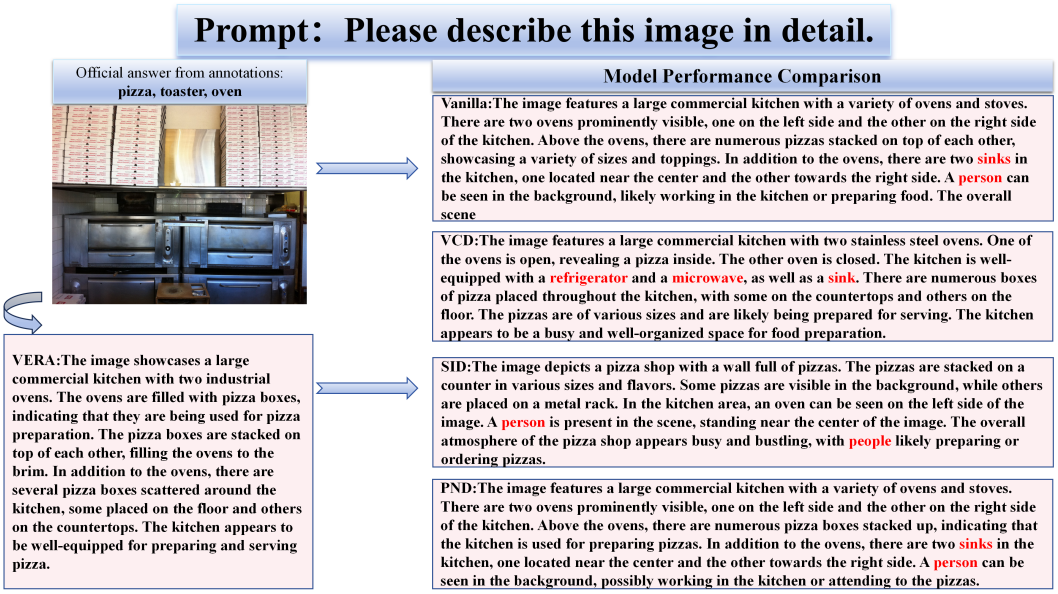

Figure 10: Case study on CHAIR. VERA reduces object hallucinations in long-form image descriptions while preserving comparable caption length. Hallucinated objects identified by the CHAIR evaluator are marked in red.

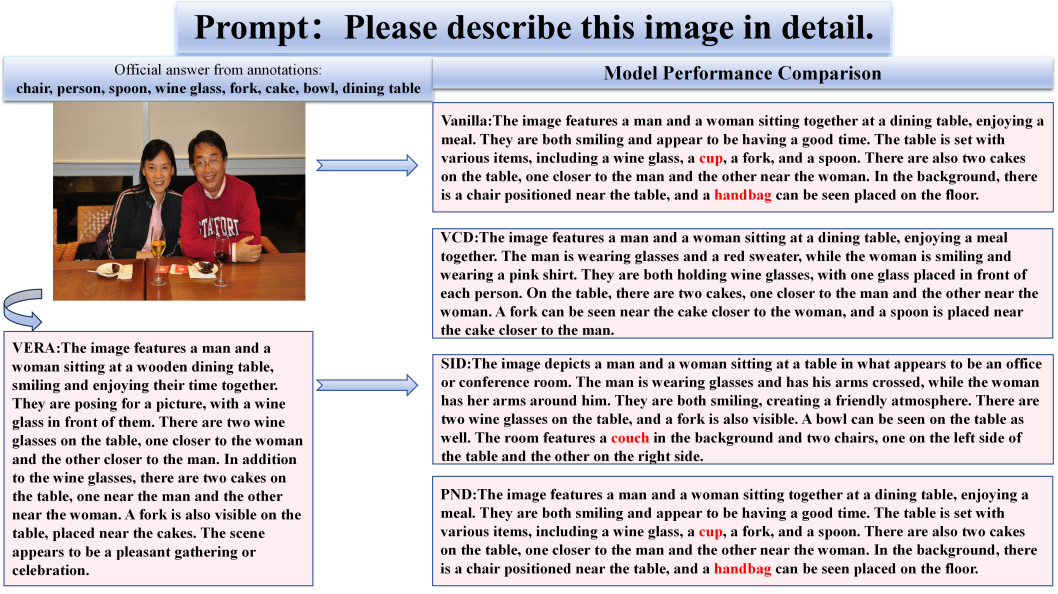

Figure 11: Case study on CHAIR. VERA reduces object hallucinations in long-form image descriptions while preserving comparable caption length. Hallucinated objects identified by the CHAIR evaluator are marked in red.

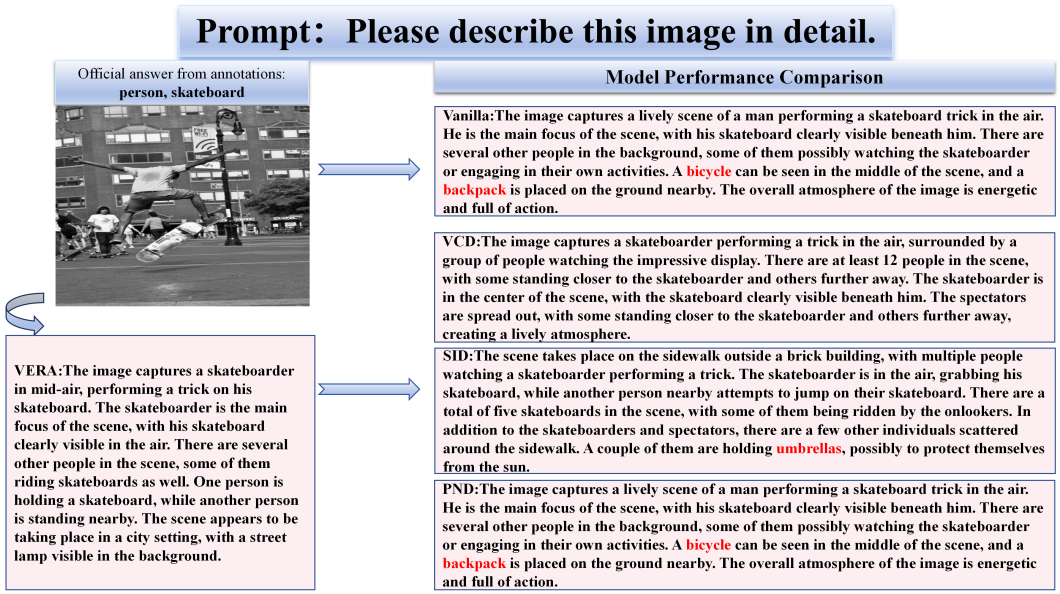

Figure 12: Case study on CHAIR. VERA reduces object hallucinations in long-form image descriptions while preserving comparable caption length. Hallucinated objects identified by the CHAIR evaluator are marked in red.

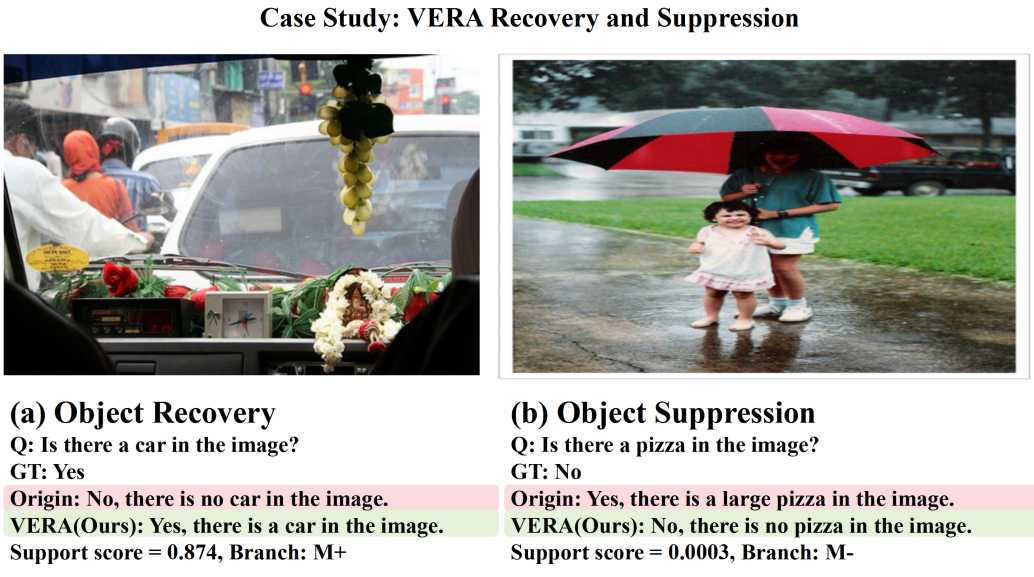

Figure 13: Qualitative examples on POPE. The left case shows present object recovery, where VERA restores a visually supported object claim. The right case shows unsupported-object suppression, where VERA rejects an object claim lacking visual evidence.

Case Study: VERA Recovery and Suppression



(a) Object Recovery
Q: Is there a chair in the image?
GT: Yes
Origin: No, there is no chair in the image.
VERA(Ours): Yes, there is a chair in the image.
Support score = 0.843, Branch: M+



(b) Object Suppression
Q: Is there a broccoli in the image?
GT: No
Origin: Yes, there is a sign on the back of the bus.
VERA(Ours): No, there is no broccoli in the image.
Support score = 0.0003, Branch: M-

Figure 14: Qualitative examples on POPE. The left case shows present object recovery, where VERA restores a visually supported object claim. The right case shows unsupported-object suppression, where VERA rejects an object claim lacking visual evidence.

Case Study: VERA Recovery and Suppression



(a) Object Recovery
Q: Is there a snowboard in the image?
GT: Yes
Origin: No, there is no snowboard in the image.
VERA(Ours): Yes, there is a snowboard in the image.
Support score = 0.843, Branch: M+



(b) Object Suppression
Q: Is there a keyboard in the image?
GT: No
Origin: Yes, there is a keyboard in the image.
VERA(Ours): No, there is no keyboard visible in the image.
Support score = 0.0003, Branch: M-

Figure 15: Qualitative examples on POPE. The left case shows present object recovery, where VERA restores a visually supported object claim. The right case shows unsupported-object suppression, where VERA rejects an object claim lacking visual evidence.

Case Study: VERA Recovery and Suppression



(a) Object Recovery

Q: Is there a backpack in the image?
GT: Yes
Origin: No, there is no backpack in the image.
VERA(Ours): Yes, there is a backpack in the image.
Support score = 0.843, Branch: M+



(b) Object Suppression

Q: Is there a tie in the image?
GT: No
Origin: Yes, there is a tie in the image.
VERA(Ours): No, there is no tie in the image.
Support score = 0.0003, Branch: M-

Figure 16: Qualitative examples on POPE. The left case shows present object recovery, where VERA restores a visually supported object claim. The right case shows unsupported-object suppression, where VERA rejects an object claim lacking visual evidence.