PAVER: DUAL-SOURCE PARSING AND PREMISE-AWARE ROUTING FOR RELIABLE LVLM RESPONSES

Anonymous authors
Paper under double-blind review

ABSTRACT

Large vision-language models (LVLMs) can produce unreliable responses for two distinct reasons: a user question may presuppose visual conditions that do not hold, while the model may introduce unsupported content in its answer. These failures require different treatment because a visual proposition may be a required question premise, queried content, or a model assertion. We introduce **PAVER** (**P**remise- and **A**nswer-aware **V**isual **E**vidence **R**outing), a training-free framework that operationalizes this distinction asymmetrically. Required question premises determine whether the question remains valid, whereas entities exposed by an initial draft only guide targeted visual inspection. Grounding DINO and Florence-2 provide shared visual evidence for premise-aware routing and final response generation. The router selects refusal when a required premise is contradicted and otherwise selects evidence-grounded re-answering. Across six benchmarks and two LVLM backbones, PAVER reduces answer-side hallucination and improves premise-sensitive reliability. Factorized ablations further show complementary roles: the draft-guided pathway improves answer-side reliability, while explicit premise modeling improves premise-sensitive behavior. These results support interaction-role-aware evidence routing as a practical principle for reliable LVLM responses.

1 INTRODUCTION

Large vision-language models (LVLMs) are increasingly capable visual assistants (Liu et al., 2023; Li et al., 2023a; Liu et al., 2024), yet often generate objects, attributes, or relations unsupported by the image (Li et al., 2023b; Guan et al., 2024; Wu et al., 2024a; Wang et al., 2024). Training-free mitigation typically modifies decoding (Leng et al., 2024; Huang et al., 2024; He et al., 2025), verifies generated claims (Sahu et al., 2024), or injects auxiliary visual evidence (Li et al., 2025). These approaches are largely *answer-centric*, implicitly assuming that the user question is visually valid. Conversely, false-premise research shows that questions themselves may contain invalid visual conditions (Mahendru et al., 2017; Wu et al., 2024b; Wang et al., 2024), but generally does not address unsupported content subsequently introduced by the answer. Addressing both sources requires interpreting visual evidence according to the role of the proposition being assessed.

We characterize this problem by the *semantic role* of visual propositions. Question content may be presupposed as a required premise or placed under inquiry, while draft content constitutes model assertions. As illustrated in Figure 1, "the man is wearing a helmet" is required by "What color is the helmet the man is wearing?", but queried by "Is the man wearing a helmet?" The same proposition becomes an answer assertion when introduced by a draft. These roles require different actions: a contradicted required premise should invalidate the question as posed, queried content may simply receive a negative answer, and unsupported draft content should be removed through evidence-grounded re-answering. Thus, visual evidence alone is insufficient; its interpretation must preserve the interaction-level role that motivated verification.

We introduce **PAVER**, a training-free framework for asymmetric evidence routing. Required question premises govern question validity, while entities exposed by an initial draft guide targeted inspection. Grounding DINO and Florence-2 provide shared evidence for premise-aware refusal or evidence-grounded re-answering. Only contradicted required premises can trigger refusal; the draft serves as an inspection plan.

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

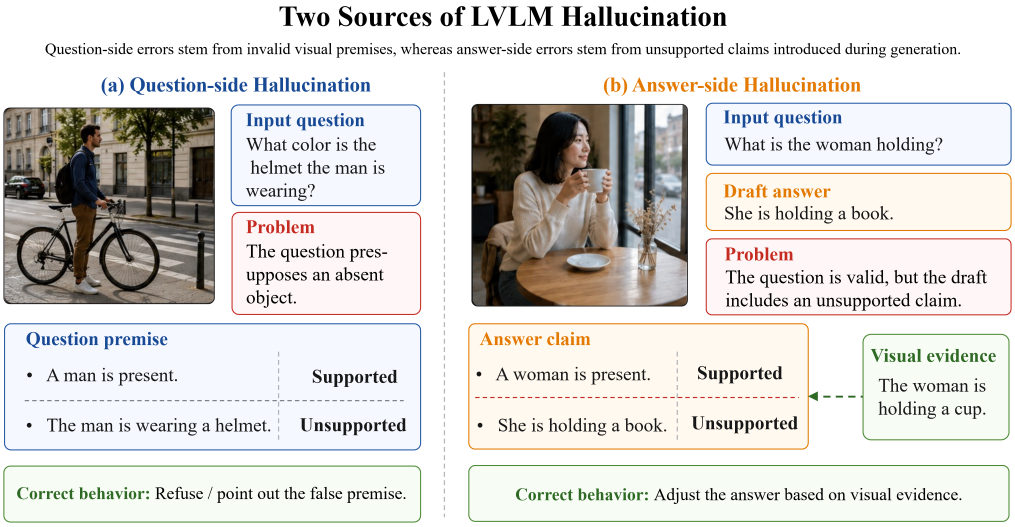

Figure 1: Two interaction-level sources of failure require different response actions. A contradicted required premise should trigger refusal, whereas an unsupported answer assertion should be removed through evidence-grounded re-answering.

Across six benchmarks and two LVLM backbones, PAVER reduces unsupported answer content and improves premise-sensitive reliability. Length-controlled AMBER results show that the hallucination gains persist at comparable response lengths. Factorized ablations further reveal the predicted division of labor: the draft-guided pathway improves answer-side reliability, while explicit premise modeling improves premise-sensitive behavior. A targeted human audit additionally shows that ICL improves fully correct premise extraction from 63% to 83% and reduces queried-content leakage from 19% to 9%, supporting the intended premise–query distinction.

Our contributions are threefold:

(1) Interaction-role formulation of visual reliability. We distinguish required question premises, queried content, and model-introduced assertions by their interaction roles, motivating different response actions under conflicting visual evidence: invalidating the question, determining the answer, or revising unsupported model content.

(2) Asymmetric role-conditioned evidence routing. We introduce PAVER, a training-free framework with ICL-enhanced source-specific parsing, in which required premises govern question validity while draft-exposed entities guide visual evidence acquisition. A premise-aware router then selects refusal or evidence-grounded re-answering.

(3) Empirical validation of the role decomposition. Across six benchmarks and two LVLM backbones, PAVER improves answer-side and premise-sensitive reliability. Ablations show complementary roles of draft-guided inspection and premise modeling, while direct audits support the premise–query distinction and the operational role of draft-guided inspection.

2 Related Work

2.1 LVLM Hallucination and Training-Free Mitigation

LVLM hallucination is commonly studied as a mismatch between generated content and the image, with evaluations extending from object existence (Li et al., 2023b) to relations (Wu et al., 2024a) and entangled visual-language reasoning (Guan et al., 2024). Training-free mitigation improves visual grounding through decoding or representation-level interventions. Visual Contrastive Decoding contrasts clean and distorted visual contexts (Leng et al., 2024), while OPERA penalizes over-trusted

token patterns (Huang et al., 2024). DeCo recovers visual information from preceding layers (Wang et al., 2025), VHR reinforces vision-aware attention heads (He et al., 2025), AGLA combines global and prompt-relevant local features (An et al., 2025), and SHIELD addresses biases in the visual encoder (Huang et al., 2026). These methods primarily suppress unsupported answer content while assuming a visually valid question. PAVER instead combines draft-guided inspection with explicit modeling of the premises required by the question.

2.2 False-Premise Reasoning in Multimodal Models

Question premises have long been recognized as an under-modeled aspect of visual question answering (Mahendru et al., 2017). Recent work explicitly studies false-premise queries: SESAME detects nonexistent referents in multimodal segmentation (Wu et al., 2024b), while HaloQuest evaluates false premises alongside insufficient context and difficult visual reasoning (Wang et al., 2024). Bottom-Up Holistic Reasoning verifies visual and textual inputs through perception- and cognition-level reasoning (Wu et al., 2025a), and JBA evaluates whether MLLMs preserve true-premise questions while rejecting false-premise ones (Li et al., 2026). However, mentioning an absent visual entity does not necessarily presuppose its existence, as in existence questions that admit a negative answer. PAVER therefore extracts only propositions required for question validity, preventing queried content from independently triggering refusal.

2.3 Claim-Level Verification and External Visual Evidence

Fine-grained hallucination methods increasingly decompose outputs into verifiable units. FaithScore extracts atomic image facts for reference-free faithfulness evaluation (Jing et al., 2024), while UNIHD coordinates auxiliary tools to detect multiple hallucination types (Chen et al., 2024). PELICAN decomposes generated claims for tool-based verification and correction (Sahu et al., 2024), while Visual Evidence Prompting translates specialist visual outputs into textual evidence (Li et al., 2025). MARINE uses external object-level guidance to improve visual grounding (Zhao et al., 2024). REVERSE combines hallucination-aware training with retrospective resampling for self-verification and correction (Wu et al., 2025b). PAVER introduces an explicit separation between question validity and visual inspection: required premises govern whether to answer, while draft-exposed entities guide what to inspect. This separation connects shared visual evidence to premise-aware refusal or evidence-grounded re-answering.

3 Method

3.1 Problem Formulation and Overview

Given an image I and question q, PAVER uses a frozen LVLM M_θ to generate a reliable response $a^\star$. Visual propositions have distinct interaction-level roles: required premises or queried information in q, and model assertions in a draft. These roles determine how conflicting evidence affects the response.

As shown in Figure 2, PAVER first generates a draft answer:

$$a^{(0)} = M_\theta(I, q; \pi_{\text{draft}}). \tag{1}$$

Two parsers extract required premises $\mathcal{P}_q$ from the question and grounding targets $\mathcal{U}_a$ from $(q, a^{(0)})$. The targets combine question-derived and draft-exposed entities for Grounding DINO localization. These detections and Florence-2's global observations form shared evidence $\mathcal{E}$. The router selects refusal only when the image and tool evidence clearly contradict a required premise, and the answer branch otherwise.

$\mathcal{P}_q$ is a *decision representation* defining question validity, whereas $\mathcal{U}_a$ is an *evidence-acquisition representation* guiding inspection without assigning validity. Only contradictions of required premises can trigger refusal.

Formally, let $\mathcal{Z}_q$ denote candidate visual propositions evoked by the question. Their question-side roles are characterized by

$$\begin{aligned} &\rho_q : \mathcal{Z}_q \to \{\textsc{premise}, \textsc{query}\}, \\ &\mathcal{P}_q = \{z \in \mathcal{Z}_q : \rho_q(z) = \textsc{premise}\}. \end{aligned} \tag{2}$$

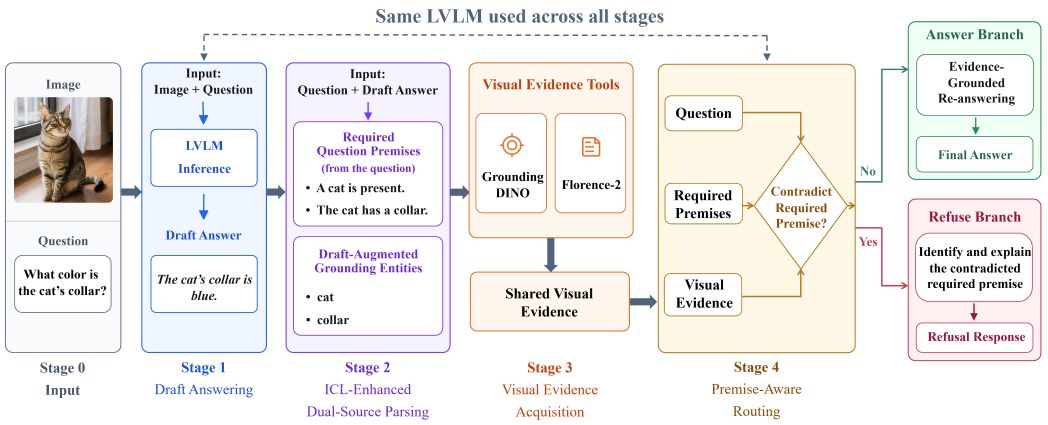

Figure 2: Overview of PAVER's asymmetric evidence flow. Required question premises $\mathcal{P}_q$ define validity conditions, while draft-augmented grounding targets $\mathcal{U}_a$ define what visual content to inspect. Grounding DINO and Florence-2 provide shared evidence $\mathcal{E}$, which the premise-aware router interprets with respect to $\mathcal{P}_q$ to select refusal or evidence-grounded re-answering.

Here ρ_q denotes the selection criterion induced by the premise-extraction prompt rather than an explicitly trained classifier. PAVER materializes only $\mathcal{P}_q$. Query-role content remains in the original question and cannot independently trigger refusal. Each $p_i \in \mathcal{P}_q$ must hold for q to remain answerable as written.

The draft may introduce unsupported visual assertions. PAVER uses entities exposed by the draft to expand $\mathcal{U}_a$ and acquire evidence for a new answer, without explicitly verifying each draft assertion.

3.2 ICL-Enhanced Source-Specific Parsing

PAVER instantiates the two representations using separate ICL-enhanced prompts on the same frozen LVLM:

$$\mathcal{P}_q = P_\theta(q; \mathcal{D}_Q), \qquad \mathcal{U}_a = G_\theta(q, a^{(0)}; \mathcal{D}_A), \tag{3}$$

where $\mathcal{D}_Q$ and $\mathcal{D}_A$ are source-specific demonstration sets fixed before evaluation and shared across datasets. The premise demonstrations are backbone-specific (Appendix F), while all six evaluation benchmarks use the same six grounding demonstrations in $\mathcal{D}_A$ (Appendix G). Neither set contains instances from any of the six evaluation benchmarks. P_θ and G_θ denote prompted extraction operations rather than separately trained models.

The **question-side parser** extracts only propositions presupposed by the question. For example, "What color is the cat's collar?" requires a collar to exist, whereas "Is the cat wearing a collar?" queries that relation and should receive a negative answer rather than refusal when no collar is present. The ICL demonstrations therefore encode the distinction between answerability conditions and queried content rather than relying on surface entity mentions.

The **draft-augmented grounding parser** extracts visual inspection targets from $(q, a^{(0)})$. Question-derived entities preserve the original task target and referential constraints, while draft-exposed entities expand the inspection set. Together, these entities specify the queries used for targeted visual evidence acquisition.

3.3 Visual Evidence Acquisition

PAVER acquires visual evidence using Grounding DINO (Liu et al., 2025) for open-vocabulary localization and Florence-2 (Xiao et al., 2024). Let T_D and T_F denote the frozen visual tools:

$$\mathcal{E}_D = \{T_D(I, u) : u \in \mathcal{U}_a\}, \qquad \mathcal{E}_F = T_F(I), \qquad \mathcal{E} = \mathcal{E}_D \cup \mathcal{E}_F. \tag{4}$$

$\mathcal{U}_a$ determines the targets for Grounding DINO, while Florence-2 provides complementary global visual evidence. Their outputs are aggregated into the shared evidence $\mathcal{E}$. Importantly, the visual tools are role-agnostic: they provide observations but do not decide whether those observations invalidate a premise, answer queried content, or challenge a draft-induced commitment. The router uses these observations to assess question validity, and the selected response branch uses them to generate the final answer.

3.4 Premise-Aware Routing and Evidence-Grounded Answering

The router R_θ uses the same frozen LVLM with a dedicated routing prompt:

$$r = R_\theta(I, q, \mathcal{P}_q, \mathcal{E}; \pi_{\text{route}}) \in \{\text{ANSWER}, \text{REFUSE}\}. \tag{5}$$

The router receives the original image and question, extracted required premises, and shared tool evidence, but not the draft or its grounding targets. It assesses question validity using both the image and tool observations, selecting REFUSE only when they clearly contradict a required premise and ANSWER otherwise. This decision selects the response branch, whose prompt governs final generation. A failed Grounding DINO localization alone is therefore insufficient to establish contradiction, and draft-exposed entities cannot directly invalidate the question.

This role-conditioned decision is essential because the same visual evidence can imply different response actions. Evidence that an entity is absent should support a negative answer when its existence is queried, but may require refusal when its existence is presupposed by the question. The router therefore interprets evidence relative to question validity rather than mapping visual absence directly to refusal.

The selected route invokes one of two branch-specific response prompts:

$$a^\star = \begin{cases} M_\theta(I, q, \mathcal{P}_q, \mathcal{E}; \pi_{\text{ref}}), & r = \text{REFUSE}, \\ M_\theta(I, q, \mathcal{E}; \pi_{\text{ans}}), & r = \text{ANSWER}. \end{cases} \tag{6}$$

In the refusal branch, π_{ref} asks the model to identify the contradicted required premise and explain why the question cannot be answered as posed. In the answer branch, π_{ans} asks the model to answer the original question using the image and tool evidence. Both branches receive the original image and shared observations, while the refusal branch additionally receives $\mathcal{P}_q$ to explain the invalid premise. Each branch generates a new response rather than editing the draft.

PAVER can therefore be summarized as follows: *the question determines what must hold, the question and draft determine what to inspect, the tools provide what is observed, and the router determines how to respond*.

4 Experiments

We evaluate PAVER along three questions: (1) whether draft-guided inspection reduces unsupported answer content, (2) whether explicit premise modeling improves false-premise detection and the reliability of true-premise predictions, and (3) whether the source-specific pathways and premise-aware router exhibit the complementary roles predicted by our formulation. We report results on six benchmarks spanning open-ended generation, object and attribute judgments, false-premise recognition, and selective answering. In all tables, the best and second-best results within each backbone are shown in **bold** and underlined, respectively; ties share the same style. *Ours* denotes PAVER, with shaded rows highlighting our method.

4.1 Experimental Setup

Models and visual evidence tools. We evaluate `llava-hf/llava-1.5-7b-hf` (Liu et al., 2023) and the stronger `Qwen/Qwen3-VL-8B-Instruct` (Bai et al., 2025). PAVER uses `IDEA-Research/grounding-dino-base` for open-vocabulary localization and `microsoft/Florence-2-large-ft`, both with default Hugging Face configurations. All model parameters remain frozen.

Benchmarks and metrics. For answer-side reliability, OpenCHAIR (Ben-Kish et al., 2024) measures open-vocabulary object hallucination with OCH; AMBER Generation (Wang et al., 2023) reports CHAIR, hallucination rate (Hal), cognition error (Cog), and object coverage (Cover); POPE-MSCOCO (Li et al., 2023b) evaluates object existence using accuracy and F1; and MME-Hall (Fu et al., 2025) covers existence, counting, position, and color. For premise-side reliability, HaloQuest (Wang et al., 2024) evaluates insufficient context (IC), false premises (FP), and visual challenge (VC), while JBA (Li et al., 2026) reports false-premise coverage (FPC), F1, true-premise identification rate (TPIR), and false-premise detection precision (FPDP). Lower values are better for OCH, CHAIR, Hal, and Cog. Higher values are better for all remaining metrics, including AMBER Cover.

Baselines and evaluation protocol. We compare PAVER with Vanilla and representative training-free interventions on frozen LVLM backbones: VCD (Leng et al., 2024), DeCo (Wang et al., 2025), and SHIELD (Huang et al., 2026) modify decoding or internal representations, while MARINE (Zhao et al., 2024) incorporates external visual guidance. We use official implementations and adapt their inference procedures to Qwen3-VL without changing the core algorithms. Bottom-Up Holistic Reasoning (Wu et al., 2025a) is omitted because no public implementation was available. Official benchmark evaluation code and prompts are used. Appendix D details decoding settings and evaluation procedures. HaloQuest and JBA use `qwen3.7-flash` with the official judge prompts. Comparisons use each method's native end-to-end inference procedure, with latency and peak memory reported in Appendix A.1. The appendices provide baseline configurations, evaluation procedures, and ICL demonstrations. Complete PAVER stage prompts and generation configurations are available in the anonymous code repository linked in the Reproducibility Statement. In particular, $\mathcal{D}_Q$ and $\mathcal{D}_A$ contain no instances from any of the six evaluation benchmarks, were fixed before evaluation, and are shared across datasets.

Prompt-based length control. To test whether AMBER gains arise from shorter responses, we regenerate Qwen3-VL-8B Vanilla and MARINE with the same length instruction and a 128-token budget, using PAVER's native outputs as the target-length reference. The resulting mean lengths (59.8 and 60.0 words) closely match PAVER's 61.0 words. Baselines are regenerated rather than truncated, and PAVER's native result is repeated in Table 3 only as the fixed reference. The exact length-control instruction is available in the anonymous code repository.

Human evaluation protocol. We audit premise extraction with and without ICL on the same stratified 100-question JBA sample (50 true-premise and 50 false-premise) using LLaVA-1.5-7B. Appendix D.2 details the annotation procedure and reported categories.

4.2 Answer-Side Response Reliability

Open-ended generation. As shown in Table 1, PAVER achieves the lowest hallucination scores on both OpenCHAIR and AMBER across the two backbones. On OpenCHAIR, OCH decreases by 30.7% on LLaVA-1.5-7B and 20.6% on Qwen3-VL-8B relative to the strongest baseline. On AMBER, PAVER achieves the best CHAIR, Hal, and Cog scores, reducing Hal from the strongest baseline's 23.4 to 14.8 on LLaVA and from 48.3 to 20.1 on Qwen. On LLaVA, these gains accompany the highest object coverage of 53.1, showing that fewer hallucinations can coincide with broader coverage of supported objects. On Qwen, PAVER reduces CHAIR from 7.1 to 3.8 relative to MARINE, while native coverage falls from 72.6 to 63.4. The two backbones therefore exhibit different coverage outcomes alongside consistent hallucination reductions. The following length-controlled analysis examines this trade-off on Qwen, whose PAVER responses are substantially shorter.

Category-level hallucination. On MME-Hall (Table 2), PAVER achieves the highest total score on both backbones, exceeding the runner-up by 88.3 points on LLaVA and 18.3 points on Qwen. It ranks first in all four categories on LLaVA, with particularly large gains in counting and position: the scores increase from the strongest baselines' 133.3 and 108.3 to 163.3 and 135.0, respectively. On Qwen, PAVER leads in existence and position, ties for the best counting score, and shares second place in color. Its position score reaches 168.3, exceeding the strongest baseline by 15.0 points, while its existence score reaches 200.0. Position thus improves over the strongest baseline on both backbones, alongside gains or competitive performance in the other categories. These results extend the evidence for PAVER's reliability beyond open-ended object descriptions to counting, spatial, and attribute judgments.

Table 1: Hallucination and object coverage on OpenCHAIR and AMBER Generation.

	OpenCHAIR	AMBER Generation			
Method	OCH↓	CHAIR↓	Cover↑	Hal↓	Cog↓
LLaVA-1.5-7B					
Vanilla	7.9681	9.5	50.3	42.9	4.2
DeCo	6.0389	7.3	48.0	29.8	2.9
VCD	6.9171	9.9	51.3	43.9	4.7
SHIELD	5.8976	11.3	50.8	47.8	6.6
MARINE	5.9672	5.1	49.7	23.4	2.3
Ours	**4.0853**	**4.1**	**53.1**	**14.8**	**0.8**
Qwen3-VL-8B					
Vanilla	4.8856	8.3	72.7	58.9	3.3
DeCo	4.5790	7.6	**73.1**	57.8	3.2
VCD	5.1481	9.1	72.4	62.7	3.2
SHIELD	4.8585	9.6	72.6	58.9	5.0
MARINE	4.7789	7.1	72.6	48.3	2.5
Ours	**3.6340**	**3.8**	63.4	**20.1**	**1.0**

Table 2: Category-level and total scores on MME-Hall.

Method	Exist.	Count	Pos.	Color	Total
LLaVA-1.5-7B					
Vanilla	185.0	133.3	101.6	150.0	570.0
DeCo	185.0	103.3	90.0	135.0	513.3
VCD	170.0	121.6	108.3	130.0	530.0
SHIELD	185.0	121.6	100.0	128.3	535.0
MARINE	190.0	88.3	86.6	110.0	475.0
Ours	**195.0**	**163.3**	**135.0**	**165.0**	**658.3**
Qwen3-VL-8B					
Vanilla	195.0	166.6	148.3	195.0	705.0
DeCo	195.0	**175.0**	153.3	195.0	718.3
VCD	190.0	163.3	143.3	190.0	686.6
SHIELD	195.0	170.0	148.3	195.0	708.3
MARINE	195.0	171.6	153.3	**200.0**	720.0
Ours	**200.0**	**175.0**	**168.3**	195.0	**738.3**

Response-length control. PAVER's native Qwen responses average 61.0 words, compared with 193.0 for Vanilla and 139.9 for MARINE. We regenerate both baselines to comparable mean lengths of 59.8 and 60.0 words (Table 3). At these lengths, PAVER achieves lower CHAIR (3.8) and Hal (20.1) and higher Cover (63.4) than both baselines.

Table 3: Native and length-controlled results on AMBER Generation with Qwen3-VL-8B.

Method	Length (words)	CHAIR↓	Cover↑	Hal↓	Cog↓
Native outputs (before length control)					
Vanilla	193.0	8.3	**72.7**	58.9	3.3
MARINE	139.9	7.1	72.6	48.3	2.5
Ours	61.0	**3.8**	63.4	**20.1**	**1.0**
Length-controlled comparison (target: PAVER native = 61.0 words)					
Vanilla (regen.)	59.8	5.6	60.7	25.7	**1.0**
MARINE (regen.)	60.0	5.9	62.6	28.1	1.9
Ours (native ref.)	61.0	**3.8**	**63.4**	**20.1**	**1.0**

Object-existence judgments. PAVER also achieves the best aggregate POPE accuracy and F1 on both backbones (Table 4), improving F1 over the runner-up by 1.91 points on LLaVA and 1.97 points on Qwen. Together with the MME-Hall results, these gains show that PAVER's improvements extend to constrained visual judgments.

Table 4: Aggregate object-existence accuracy and F1 on POPE-MSCOCO. Subset-level results are provided in Appendix A.2.

	LLaVA-1.5-7B		Qwen3-VL-8B	
Method	Acc.↑	F1↑	Acc.↑	F1↑
Vanilla	78.72	73.58	88.69	88.01
DeCo	84.34	82.21	88.89	88.50
VCD	80.39	77.41	88.59	88.08
SHIELD	85.96	84.89	87.87	86.66
MARINE	79.88	75.22	88.81	88.26
Ours	**87.76**	**86.80**	**90.79**	**90.47**

4.3 Premise-Side Reliability and Selective Answering

False premises and insufficient context. Table 5 shows that PAVER improves premise-sensitive reliability. On LLaVA, IC and FP accuracy increase from the best baselines' 19.17 and 35.55 to

60.00 and 75.08, yielding a 25.83-point Overall gain. PAVER also achieves the best IC, FP, and Overall scores on Qwen. VC accuracy decreases relative to Vanilla on both backbones, revealing a trade-off between premise-sensitive reliability and performance on challenging visual questions. Appendix C presents an over-refusal case in which the router rejects a question despite a correct initial draft.

Premise detection and prediction reliability. JBA results in Table 6 evaluate false-premise detection and the reliability of true-premise predictions. TPIR measures the proportion of genuinely true-premise questions among those predicted as true-premise. On LLaVA, PAVER achieves the best FPC, F1, and TPIR, although FPDP drops to 66.3. On Qwen, it outperforms the runner-up by 10.7, 14.6, and 13.5 points in FPC, F1, and TPIR, respectively, while maintaining a high FPDP of 93.9.

LLaVA false-premise precision–recall trade-off. On LLaVA, PAVER operates at a broader false-premise detection point. Vanilla and SHIELD recover only 22.3% and 30.2% of false-premise cases, whereas PAVER raises recall to 72.5% while predicting 54.6% of examples as false-premise. This broader operating point trades precision for substantially higher recall: FPDP decreases to 66.3%, while F1 increases to 69.3 from 36.0 and 44.7 for Vanilla and SHIELD.

Table 5: Category-level and overall accuracy on HaloQuest.

Method	IC↑	FP↑	VC↑	Overall↑
LLaVA-1.5-7B				
Vanilla	18.33	34.55	**45.25**	34.50
DeCo	16.67	29.90	38.55	29.83
VCD	13.33	27.57	35.75	27.17
SHIELD	10.83	30.56	39.10	29.16
MARINE	19.17	35.55	43.58	34.67
Ours	**60.00**	**75.08**	36.31	**60.50**
Qwen3-VL-8B				
Vanilla	81.67	86.71	70.95	81.00
DeCo	85.00	85.71	68.72	80.50
VCD	81.67	86.05	70.39	80.50
SHIELD	83.33	89.70	70.95	82.83
MARINE	81.67	87.38	**72.07**	81.67
Ours	**85.83**	**92.03**	67.60	**83.50**

Table 6: Premise-sensitive response performance on JBA.

Method	FPC↑	F1↑	TPIR↑	FPDP↑
LLaVA-1.5-7B				
Vanilla	60.4	36.0	56.0	**93.3**
DeCo	59.2	33.9	55.6	89.3
VCD	60.3	37.6	56.1	87.9
SHIELD	62.7	44.7	58.0	86.0
MARINE	59.7	34.9	55.6	90.2
Ours	**67.9**	**69.3**	**69.8**	66.3
Qwen3-VL-8B				
Vanilla	74.4	67.3	67.1	92.9
DeCo	74.3	67.1	66.9	93.0
VCD	74.5	67.6	67.2	92.8
SHIELD	77.7	73.0	70.6	92.3
MARINE	75.7	69.5	68.1	**94.0**
Ours	**88.4**	**87.6**	**84.1**	93.9

Table 7: False-premise prediction rates, precision, and recall on JBA with LLaVA-1.5-7B.

Method	Pred. FP Rate	FP Recall↑	FPDP↑	F1↑
Vanilla	11.9	22.3	**93.3**	36.0
SHIELD	17.5	30.2	86.0	44.7
Ours	54.6	**72.5**	66.3	**69.3**

4.4 Ablation Studies

We conduct controlled ablations on LLaVA-1.5-7B. Table 8(a) evaluates ICL by removing demonstrations from both source-specific parsers while retaining their instructions and the remaining pipeline. Without ICL, HaloQuest IC and FP drop from 60.00 and 75.08 to 36.67 and 45.85, and JBA FPC from 67.9 to 55.1. Because this ablation affects both parsers, we separately audit question-side premise extraction below.

Table 8(b) tests whether the two source-specific pathways and explicit response control exhibit the division of labor predicted by our formulation. *Answer-side Only* omits explicit premises while retaining the same draft, grounding targets, visual evidence, routing and response prompts, and decoding settings as Full PAVER (Appendix E). *Question-side Only* retains required premises and the router but removes draft-guided Grounding DINO evidence. *Full w/o Router* retains the same premises, draft targets, and shared evidence as Full PAVER, but replaces separate routing and branch-specific generation with a single joint prompt.

Table 8: Ablations of ICL, source-specific pathways, and routing on LLaVA-1.5-7B.

(a) Effect of ICL

Method	IC↑	FP↑	FPC↑
Vanilla	18.33	34.55	60.4
Ours w/o ICL	36.67	45.85	55.1
Ours w/ ICL	**60.00**	**75.08**	**67.9**

(b) Factorized components and routing

Method	Premises	Draft	Router	Hal↓	MME↑	FPC↑
Vanilla	–	–	–	42.9	570.0	60.4
Answer-side Only	×	✓	✓	**14.8**	653.3	56.6
Question-side Only	✓	×	✓	28.2	565.0	67.0
Full w/o Router	✓	✓	×	**14.8**	415.3	63.9
Full (Ours)	✓	✓	✓	**14.8**	**658.3**	**67.9**

✓: retained; ×: removed.

With the same visual evidence, adding explicit premises to *Answer-side Only* raises FPC from 56.6 to 67.9 while leaving AMBER Hal unchanged at 14.8 and increasing MME-Hall from 653.3 to 658.3. This comparison supports the contribution of explicit premise representation to premise-sensitive reliability. Answer-side Only also falls below Vanilla on FPC (56.6 vs. 60.4), despite reducing AMBER Hal from 42.9 to 14.8. Thus, answer-side hallucination reduction does not automatically translate into better false-premise handling. Explicit premise inputs help the response-control stages distinguish question validity from the content selected for inspection. Conversely, *Question-side Only* nearly preserves FPC (67.0 vs. 67.9) but increases Hal to 28.2, supporting the complementary role of the draft-guided pathway.

The router ablation isolates response control while holding $\mathcal{P}_q$, $\mathcal{U}_a$, and $\mathcal{E}$ fixed. Removing it reduces MME-Hall from 658.3 to 415.3, supporting the benefit of separating routing from response generation. Visual absence may imply a negative answer for queried content but invalidate the question when it contradicts a required premise.

Direct evaluation of premise extraction. The human audit in Table 9 directly evaluates the question-side representation. ICL improves Fully Correct premise extraction from 63.0% to 83.0%, while reducing Under-extraction from 12.0% to 7.0%, Over-extraction from 25.0% to 10.0%, and Query Leakage from 19.0% to 9.0%. These results support more selective premise extraction and the intended premise–query distinction. A complementary audit on 1,004 AMBER images shows that ICL improves the coverage of unsupported draft entities and captured-entity removal while preserving over 90% of supported draft entities (Appendix B). These audits characterize the intermediate representations and final-response outcomes.

Table 9: Human audit of premise extraction with and without ICL on JBA ($n = 100$, LLaVA-1.5-7B).

Method	Fully Correct↑	Under-extraction↓	Over-extraction↓	Query Leakage↓
Ours w/o ICL	63.0	12.0	25.0	19.0
Ours w/ ICL	**83.0**	**7.0**	**10.0**	**9.0**

Note. Values are percentages on the balanced sample, with Query Leakage included in Over-extraction.

5 Conclusion

We introduced PAVER, a training-free framework for interaction-role-aware LVLM response reliability. PAVER separates required question premises that govern validity from draft-exposed targets that guide visual inspection, and uses shared visual evidence with premise-aware routing to produce role-appropriate responses. Across six benchmarks and two LVLM backbones, PAVER improves both answer-side and premise-sensitive reliability, while factorized ablations support the intended role decomposition and the benefit of explicit response control. The results also reveal trade-offs in descriptive coverage and challenging visual questions, underscoring that response reliability is distinct from intrinsic visual capability. Future work could explore post-training methods that internalize premise–query discrimination and role-conditioned response control, reducing reliance on external inference stages. Overall, reliable multimodal responding requires not only acquiring visual evidence, but interpreting it according to its interaction-level role.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation. We also used `qwen3.7-flash` to evaluate HaloQuest and JBA responses through their official evaluation scripts and judge prompts, and to perform preliminary checks of premise-extraction outputs. These preliminary checks were subsequently reviewed and confirmed by a human annotator who was unaware of the experimental condition. Apart from these uses, we did not use generative AI tools for other tasks requiring disclosure under the ICLR policy. Additionally, we used generative AI tools to edit code, summarize or analyze existing literature, search for information, and edit the manuscript to improve readability. All AI-assisted work was reviewed by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies response-level reliability in large vision-language models and does not involve human subjects or the collection of new personal data. By reducing unsupported visual claims and improving the handling of false-premise questions, the method is intended to support safer model behavior. Nevertheless, errors in the visual tools or routing stage may still produce incorrect answers or refusals, so the framework should not be treated as a substitute for human review in high-stakes settings.

REPRODUCIBILITY STATEMENT

The main method, model and tool configurations, evaluation protocol, factorized ablations, and audit results are described in Sections 3–4. Appendices A–D provide efficiency measurements, complete POPE results, the draft-side audit, qualitative cases, detailed evaluation and human-audit procedures, baseline prompts, and benchmark-specific decoding configurations. Appendix E details the matched ablation control, and Appendices F and G provide the premise-extraction and grounding-query demonstrations. The bibliography identifies the official benchmark and baseline sources used in the experiments. Complete PAVER stage prompts and generation configurations are available in the anonymous code repository. Anonymous code repository: `https://anonymous.4open.science/r/PAVER-9B24`.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 29915–29926, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. doi:10.48550/arXiv.2511.21631.

Assaf Ben-Kish, Moran Yanuka, Morris Alper, Raja Giryes, and Hadar Averbuch-Elor. Mitigating open-vocabulary caption hallucinations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22680–22698. Association for Computational Linguistics, 2024. doi:10.18653/v1/2024.emnlp-main.1263.

Xiang Chen, Chenxi Wang, Yida Xue, Ningyu Zhang, Xiaoyan Yang, Qiang Li, Yue Shen, Lei Liang, Jinjie Gu, and Huajun Chen. Unified hallucination detection for multimodal large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3235–3252. Association for Computational Linguistics, 2024. doi:10.18653/v1/2024.acl-long.178.

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 *Advances in Neural Information Processing Systems*, volume 38, 2025. doi:10.52202/085713-4899. Datasets and Benchmarks Track.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14375–14385, 2024.

Jinghan He, Kuan Zhu, Haiyun Guo, Junfeng Fang, Zhenglin Hua, Yuheng Jia, Ming Tang, Tat-Seng Chua, and Jinqiao Wang. Cracking the code of hallucination in LVLMs with vision-aware head divergence. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3488–3501. Association for Computational Linguistics, 2025. doi:10.18653/v1/2025.acl-long.175.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13418–13427, 2024.

Yiyang Huang, Liang Shi, Yitian Zhang, Yi Xu, and Yun Fu. SHIELD: Suppressing hallucinations in LVLM encoders via bias and vulnerability defense. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=yk7FFLoNcP`.

Liqiang Jing, Ruosen Li, Yunmo Chen, and Xinya Du. FaithScore: Fine-grained evaluations of hallucinations in large vision-language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 5042–5063. Association for Computational Linguistics, 2024. doi:10.18653/v1/2024.findings-emnlp.290.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13872–13882, 2024.

Jidong Li, Lingyong Fang, Haodong Zhao, Sufeng Duan, and Gongshen Liu. Judge before answer: Can MLLM discern the false premise in question? In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2026. doi:10.1109/ICASSP55912.2026.11462162.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19730–19742. PMLR, 2023a.

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. Association for Computational Linguistics, 2025. doi:10.18653/v1/2025.acl-long.205.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne 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. Association for Computational Linguistics, 2023b. doi:10.18653/v1/2023.emnlp-main.20.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34892–34916. Curran Associates, Inc., 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26296–26306, 2024.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *Computer Vision – ECCV 2024*, volume 15105 of *Lecture Notes in Computer Science*, pp. 38–55. Springer, 2025. doi:10.1007/978-3-031-72970-6_3.

Aroma Mahendru, Viraj Prabhu, Akrit Mohapatra, Dhruv Batra, and Stefan Lee. The promise of premise: Harnessing question premises in visual question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 926–935. Association for Computational Linguistics, 2017. doi:10.18653/v1/D17-1097.

Pritish Sahu, Karan Sikka, and Ajay Divakaran. Pelican: Correcting hallucination in vision-LLMs via claim decomposition and program of thought verification. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8228–8248. Association for Computational Linguistics, 2024. doi:10.18653/v1/2024.emnlp-main.470.

Chenxi Wang, Xiang Chen, Ningyu Zhang, Bozhong Tian, Haoming Xu, Shumin Deng, and Huajun Chen. MLLM can see? dynamic correction decoding for hallucination mitigation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Junyang Wang, Yuhang Wang, Guohai Xu, Jing Zhang, Yukai Gu, Haitao Jia, Jiaqi Wang, Haiyang Xu, Ming Yan, Ji Zhang, and Jitao Sang. AMBER: An LLM-free multi-dimensional benchmark for MLLMs hallucination evaluation. *arXiv preprint arXiv:2311.07397*, 2023.

Zhecan Wang, Garrett Bingham, Adams Wei Yu, Quoc V. Le, Thang Luong, and Golnaz Ghiasi. HaloQuest: A visual hallucination dataset for advancing multimodal reasoning. In *Computer Vision – ECCV 2024*, volume 15135 of *Lecture Notes in Computer Science*, pp. 288–304. Springer, 2024. doi:10.1007/978-3-031-72980-5_17.

Mingrui Wu, Jiayi Ji, Oucheng Huang, Jiale Li, Yuhang Wu, Xiaoshuai Sun, and Rongrong Ji. Evaluating and analyzing relationship hallucinations in large vision-language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53553–53570. PMLR, 2024a.

Shengqiong Wu, Hao Fei, Liangming Pan, William Yang Wang, Shuicheng Yan, and Tat-Seng Chua. Combating multimodal LLM hallucination via bottom-up holistic reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 8460–8468, 2025a. doi:10.1609/aaai.v39i8.32913.

Tsung-Han Wu, Giscard Biamby, David Chan, Lisa Dunlap, Ritwik Gupta, Xudong Wang, Joseph E. Gonzalez, and Trevor Darrell. See, say, and segment: Teaching LMMs to overcome false premises. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13459–13469, 2024b.

Tsung-Han (Patrick) Wu, Heekyung Lee, Jiaxin Ge, Joseph E. Gonzalez, Trevor Darrell, and David Chan. Generate, but verify: Reducing hallucination in vision-language models with retrospective resampling. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. doi:10.52202/085713-2206.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4818–4829, 2024.

Linxi Zhao, Yihe Deng, Weitong Zhang, and Quanquan Gu. Mitigating object hallucination in large vision-language models via image-grounded guidance. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 77461–77486. PMLR, 2024. URL `https://proceedings.mlr.press/v267/zhao25j.html`.

A Additional Experiments

A.1 Inference Efficiency

Profiling protocol. We profile the first 50 examples of AMBER Generation (Wang et al., 2023) with LLaVA-1.5-7B, using each method's default released inference configuration. We select this backbone because the public implementations of all comparison methods support LLaVA, allowing every method to be executed through its native code path. We report the arithmetic mean of per-example latency and the population standard deviation computed with `ddof=0`. One-time model and checkpoint loading is completed before timing and is excluded from every per-example latency measurement.

The timing boundary includes all method-specific inference stages. SHIELD latency comprises caption generation followed by SHIELD's final response generation. MARINE latency includes DETR, RAM++, guidance construction, and MARINE's final response generation. PAVER latency covers the complete pipeline: draft generation, source-specific parsing, Grounding DINO and Florence-2 evidence acquisition, premise-aware routing, and branch-specific response generation. Vanilla, DeCo, and VCD are timed over their respective end-to-end response-generation procedures. Peak GPU memory is the maximum aggregate memory observed by sampling `nvidia-smi` over the entire process tree throughout each complete run; it therefore covers resident models and all method stages.

Table 10: End-to-end latency and peak GPU memory on 50 AMBER Generation examples with LLaVA-1.5-7B. Latency excludes one-time model loading. Peak memory covers the complete process tree.

Method	Mean latency (s)↓	Population std. (s)	Peak GPU memory (MB)↓
Vanilla	**4.31**	1.01	**14,844.0**
SHIELD	11.66	1.94	21,054.0
DeCo	<u>5.74</u>	1.57	15,532.0
MARINE	6.66	1.93	15,884.0
VCD	10.63	2.32	<u>15,522.0</u>
Ours	14.92	5.88	19,750.0

Vanilla is the fastest and most memory-efficient configuration, as expected for an unmodified single-model inference path. PAVER has the highest mean latency (14.92 seconds) and variability (5.88 seconds), reflecting its multi-stage evidence acquisition and response-control pipeline. Its peak GPU memory is 19,750.0 MB: higher than Vanilla, DeCo, MARINE, and VCD, but lower than SHIELD's 21,054.0 MB. These measurements characterize the reliability–efficiency trade-off of PAVER's training-free but multi-stage inference procedure.

A.2 COMPLETE POPE RESULTS

Table 11 reports the complete POPE-MSCOCO results for the Random, Popular, and Adversarial subsets, together with the aggregate scores summarized in the main paper.

Table 11: Subset-level and aggregate results on POPE-MSCOCO.

	Random		Popular		Adversarial		All	
Method	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
LLaVA-1.5-7B								
Vanilla	78.90	73.75	79.37	74.18	77.90	72.84	78.72	73.58
DeCo	85.60	83.40	84.63	82.48	82.80	80.79	84.34	82.21
VCD	82.60	79.55	79.73	76.49	78.83	76.26	80.39	77.41
SHIELD	88.37	87.15	86.50	85.42	83.00	82.24	85.96	84.89
MARINE	80.27	75.58	80.00	75.33	79.37	74.75	79.88	75.22
Ours	**88.53**	**87.47**	**86.76**	**85.82**	**88.00**	**87.13**	**87.76**	**86.80**
Qwen3-VL-8B								
Vanilla	90.73	89.96	88.47	87.80	86.87	86.34	88.69	88.01
DeCo	91.73	91.19	88.57	88.21	86.37	86.25	88.89	88.50
VCD	90.97	90.31	88.20	87.70	86.60	86.33	88.59	88.08
SHIELD	89.00	87.75	87.70	86.50	86.90	85.75	87.87	86.66
MARINE	91.13	90.47	88.47	87.93	86.83	86.47	88.81	88.26
Ours	**93.77**	**93.44**	**91.40**	**91.16**	**87.20**	**86.91**	**90.79**	**90.47**

B Draft-Target Coverage and Final-Response Audit

This audit examines how draft-side parsing shapes visual inspection and how draft commitments are represented in the final response. We measure the coverage of supported and unsupported draft entities by grounding targets, together with their subsequent preservation or removal after evidence-grounded re-answering.

B.1 Experimental Setup

We evaluate all 1,004 images in AMBER Generation with Qwen3-VL-8B as the backbone and PAVER as the inference framework. We compare PAVER w/ ICL with PAVER w/o ICL, where the latter removes only the draft-side parser's few-shot examples. The backbone, dataset, draft generation, Florence-2, Grounding DINO, router, final-answer prompt, and length settings are identical. Consequently, the two runs use exactly the same 1,004 drafts and differ only in the parser targets and their downstream effects. The target sets differ on 713 examples, and the final responses differ on 670 examples.

All entity sets are deduplicated within each image and then micro-aggregated across the 1,004 images. Thus, the set notation below is shorthand for sums of per-image cardinalities rather than a single globally deduplicated vocabulary.

B.2 Entity Extraction and Matching

For each draft caption, we extract the set of visually verifiable entities D. We lowercase the text, apply NLTK part-of-speech tagging, retain nouns, lemmatize them with WordNetLemmatizer, filter them through the AMBER object vocabulary, and deduplicate the resulting entities within the image. Using the AMBER annotations, we partition these entities into supported commitments D^+ and unsupported commitments D^-. A draft entity belongs to D^+ if it (1) exactly matches an AMBER truth object, (2) matches an AMBER-associated word for a truth object, (3) belongs to the AMBER global safe-word list, or (4) has spaCy semantic similarity greater than 0.8 to a truth object or one of its associated words. All remaining entities in the AMBER object vocabulary are assigned to D^-.

Subsequent entity matching uses the same three admissible relations: exact match after lemmatization, AMBER-associated-word match, or semantic similarity greater than 0.8. We extract the final-answer entity set F with the same normalization and vocabulary filtering. The parser output T is the collection of *dino_query* targets. Each query contributes one target to the UTR denominator. For commitment matching, however, we use only nouns in the query that belong to the AMBER object vocabulary. A query containing no valid object noun is therefore recorded as unmatched under this automatic protocol. Importantly, *unmatched* does not imply semantically irrelevant: the parser may produce an implied entity, a region-level phrase, a relational argument, or a taxonomically broader or narrower grounding query that is reasonable for visual inspection but not aligned with an AMBER-recognized draft noun. UTR therefore measures target-to-draft match coverage under the AMBER ontology, not target relevance or visual-tool utility.

B.3 Tracing Metrics

The primary parser metric is *Unsupported Commitment Recall* (UCR):

$$\mathrm{UCR} = \frac{|D^- \cap T|}{|D^-|}. \tag{7}$$

It measures how many unsupported object commitments already exposed by the draft are captured as targets for subsequent visual inspection. *Unmatched Target Rate* (UTR) measures the fraction of parser targets for which the automatic matcher finds no corresponding AMBER-recognized draft commitment:

$$\mathrm{UTR} = \frac{|\{t \in T : t \not\sim (D^+ \cup D^-)\}|}{|T|}, \tag{8}$$

where $\sim$ denotes the matching rules above. A lower UTR indicates that fewer targets are unmatched by this evaluation protocol; it must not be interpreted as the percentage of parser targets that are intrinsically irrelevant.

To summarize the final-response outcome for captured unsupported entities, we report *Hallucination Correction Rate* (HCR):

$$\mathrm{HCR} = \frac{|\{h : h \in D^- \cap T,\ h \notin F\}|}{|D^- \cap T|}. \tag{9}$$

HCR measures the fraction of captured unsupported commitments absent from the final response under the entity-matching protocol. Here, correction is operationalized as entity removal; HCR does not require an explicit correction statement or a replacement entity. It is a system-level outcome that depends on visual evidence acquisition, routing, and final response generation, and does not attribute removal to an individual grounding query.

We additionally measure supported-content behavior to test whether hallucination reduction is achieved by indiscriminately deleting valid content:

$$\mathrm{SCR} = \frac{|D^+ \cap T|}{|D^+|}, \qquad \mathrm{OPR} = \frac{|D^+ \cap F|}{|D^+|}, \qquad \mathrm{CPR} = \frac{|D^+ \cap T \cap F|}{|D^+ \cap T|}. \tag{10}$$

Supported Capture Rate (SCR) measures how many supported draft commitments are targeted; Overall Preservation Rate (OPR) measures how many remain in the final answer; and Conditional Preservation Rate (CPR) measures preservation among the supported commitments that were explicitly targeted.

B.4 Results

Because the two conditions share identical drafts, the underlying commitment counts remain fixed at 3,898 supported and 219 unsupported entities. The tracing implementation also exactly reproduces all aggregate counts for the ICL run.

Table 12: Unsupported draft-entity capture, final-response removal, and target matching on AMBER Generation. Results use Qwen3-VL-8B. Delta is Ours w/o ICL minus Ours w/ ICL.

(a) Unsupported commitment coverage and removal

Parser	Unsupported draft	Captured	UCR↑	Removed in final	HCR↑
Ours w/ ICL	219	**137**	**62.56**	**55**	**40.15**
Ours w/o ICL	219	<u>127</u>	<u>57.99</u>	<u>36</u>	<u>28.35</u>
Delta	0	-10	-4.57 pp	-19	-11.80 pp

(b) Automatic target matching

Parser	All parser targets	Unmatched targets	UTR↓
Ours w/ ICL	5,614	2,299	**40.95**
Ours w/o ICL	5,862	2,424	<u>41.35</u>
Delta	+248	+125	+0.40 pp

Note. UCR, HCR, and UTR are percentages. Each *dino_query* contributes one parser target to the UTR denominator, while commitment matching uses only valid AMBER object nouns within the query. An unmatched target is not necessarily irrelevant or unusable for grounding.

ICL captures 137 of 219 unsupported commitments, compared with 127 without ICL, increasing UCR by 4.57 percentage points. Among captured commitments, 55 of 137 are absent from the final response with ICL (40.15% HCR), compared with 36 of 127 without ICL (28.35%). These conditional rates describe final-response outcomes for the target sets produced by each parser. The higher UCR with ICL accompanies fewer targets: removing ICL produces 248 additional targets and 125 additional unmatched targets, increasing UTR from 40.95% to 41.35%. UTR describes automatic target-to-draft match coverage under the AMBER ontology rather than the semantic utility of unmatched queries.

Table 13: Capture and final-response preservation of supported draft entities on AMBER Generation. Results use Qwen3-VL-8B. Delta is Ours w/o ICL minus Ours w/ ICL.

(a) Overall supported-content behavior

Parser	Supported draft	Captured	SCR↑	Retained in final	OPR↑
Ours w/ ICL	3,898	3,318	85.12	**3,525**	**90.43**
Ours w/o ICL	3,898	3,432	**88.05**	3,512	90.10
Delta	0	+114	+2.92 pp	-13	-0.33 pp

(b) Preservation conditional on capture

Parser	Captured supported	Captured and retained	CPR↑
Ours w/ ICL	3,318	3,136	**94.51**
Ours w/o ICL	3,432	3,231	94.14
Delta	+114	+95	-0.37 pp

Without ICL, the parser captures more supported commitments (88.05% versus 85.12% SCR), consistent with a broader extraction tendency. This broader targeting does not improve preservation: PAVER w/ ICL has slightly higher overall and conditional preservation (90.43% OPR and 94.51% CPR) than PAVER w/o ICL (90.10% and 94.14%). The higher HCR with ICL therefore does not coincide with a larger loss of supported draft content.

Table 14: Final-response removal of uncaptured unsupported draft entities on AMBER Generation. Results use Qwen3-VL-8B.

Parser	Uncaptured unsupported	Removed in final	Removal rate
Ours w/ ICL	82	58	70.73
Ours w/o ICL	92	71	77.17
Delta	+10	+13	+6.44 pp

B.5 Overall Removal and Interpretation

Table 15 combines the captured and uncaptured counts already reported above. Both conditions use the same 219 unsupported draft entities, so the overall rate has a common denominator. It records removal of existing draft entities and does not count newly introduced unsupported entities in the final response.

Table 15: Overall final-response removal of unsupported draft entities on AMBER Generation. Results use Qwen3-VL-8B. Counts are aggregated from Tables 12 and 14.

Parser	Captured: removed	Uncaptured: removed	Total removed	Overall rate
Ours w/ ICL	55/137	58/82	113/219	51.60%
Ours w/o ICL	36/127	71/92	107/219	48.86%

With ICL, 113 of 219 unsupported draft entities are absent from the final response, compared with 107 without ICL: an increase of six entities, or 2.74 percentage points. This overall change complements the 11.80-point increase in conditional HCR. Because ICL changes which entities are captured, the conditional rates summarize different subsets, whereas the overall rates summarize the same initial set of unsupported entities.

Captured and uncaptured entities are selected by the parser rather than randomly assigned. In the ICL condition, the respective removal rates of 40.15% and 70.73% therefore do not measure the effect of targeting an otherwise identical entity. Florence-2 supplies global evidence, and the final response is generated anew without receiving the draft; both can contribute to omission even without a matched grounding target. Response length may also affect omission. The audit does not separately quantify these contributions, and identical length settings do not establish identical realized response lengths.

ICL improves the coverage of unsupported draft entities by inspection targets, alongside a modest increase in their overall removal from final responses and similar preservation of supported entities. The captured and uncaptured subsets describe different parts of this outcome rather than a controlled comparison of targeting effects. These observations characterize the draft-side pathway within the full system; they do not establish that target capture is the principal cause of removal. The audit covers object entities recognized by the AMBER ontology, with attribute- and relation-level commitments outside its measurement scope.

C QUALITATIVE CASE STUDIES

This appendix illustrates how PAVER's source-role decomposition affects response behavior. The first three cases trace successful answer-side inspection, queried-content handling, and premise-aware refusal, respectively. The fourth reports a genuine failure caused by an over-conservative routing decision.

C.1 AMBER GENERATION: DRAFT TARGETS AND EVIDENCE-GROUNDED RE-ANSWERING

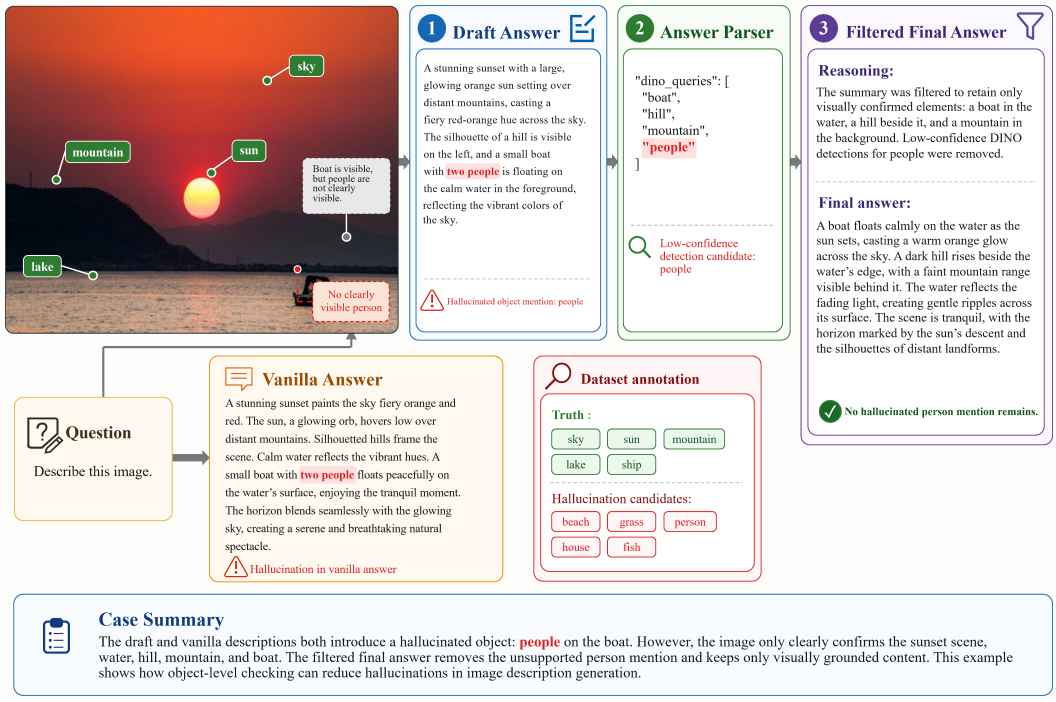

Figure 3: Draft targets and final-response content on AMBER Generation. The draft introduces unsupported people, the grounding parser exposes *people* as an inspection target, and the final caption removes this object while retaining supported scene content.

As shown in Figure 3, the draft describes "two people" in the boat even though no person is clearly visible. The parser includes *people* among the Grounding DINO queries, and the resulting evidence does not provide sufficient support for that commitment. The final caption omits the person mention while preserving the supported boat, water, sunset, hill, and mountain content. This case illustrates the sequence examined in Appendix B: an unsupported draft entity appears among the inspection targets and is absent from the evidence-grounded final response.

C.2 MME-HALLUCINATION: QUERIED PROPOSITION LEADS TO ANSWERING, NOT REFUSAL

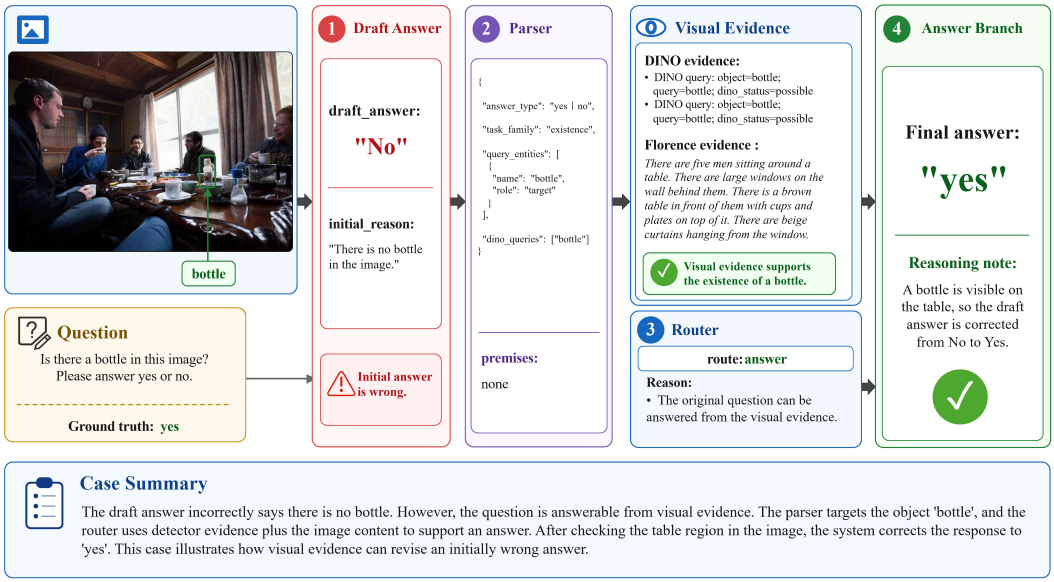

Figure 4: A queried existence proposition should be answered rather than treated as a refusal premise. The draft incorrectly answers *No*; visual evidence supports the queried bottle, and PAVER returns *Yes*.

Figure 4 demonstrates the premise–query distinction in a binary existence question. In "Is there a bottle in this image?", bottle existence is the proposition under inquiry rather than a required premise that must already hold. The draft answer is incorrect, but the parser targets the bottle and the visual evidence supports its presence. The router therefore selects ANSWER, and the final branch corrects the response from *No* to *Yes*. More generally, evidence about a queried proposition determines the answer value; it should not by itself invalidate the question.

C.3 PREMISE-AWARE REFUSAL

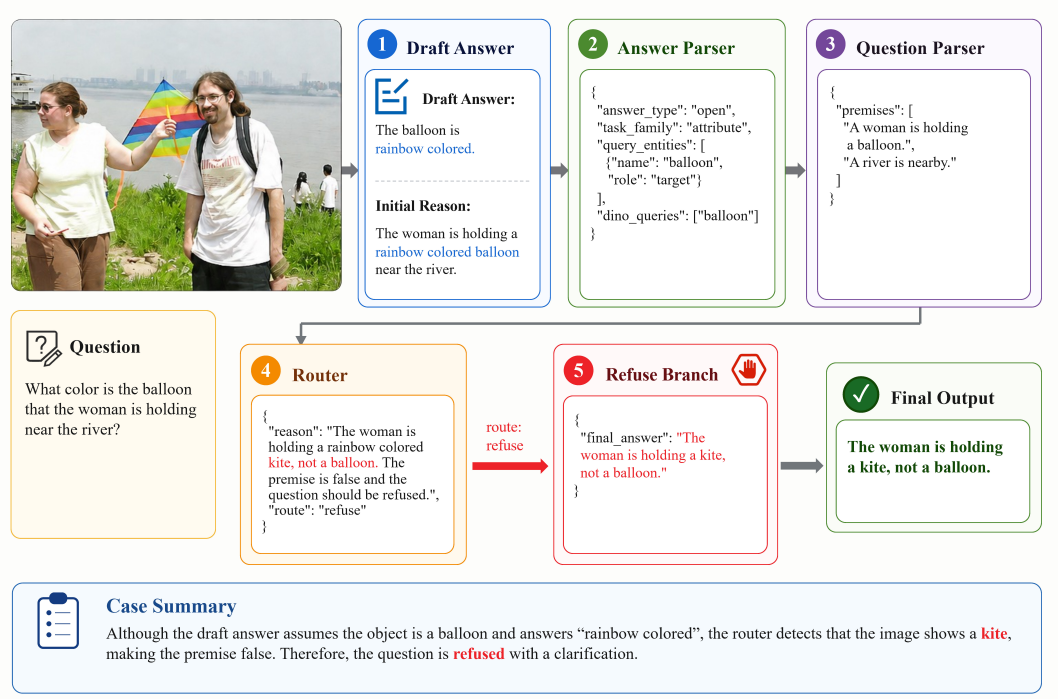

Figure 5: Premise-aware refusal when the referenced object category is contradicted. The question presupposes that the woman is holding a balloon, whereas the image shows a kite.

In Figure 5, the requested color can only be answered if the woman is holding the referenced balloon. Premise parsing makes this required condition explicit, while the visual evidence shows that the object is a kite rather than a balloon. Because the contradiction concerns a required premise rather than merely the draft's proposed answer, the router selects REFUSE. The final response identifies the specific mismatch without supplying a color. Together with Figure 4, this case shows why visually related propositions require different actions according to their interaction-level role.

C.4 HALOQUEST VISUAL CHALLENGE: FAILURE UNDER DIFFICULT VISUAL EVIDENCE

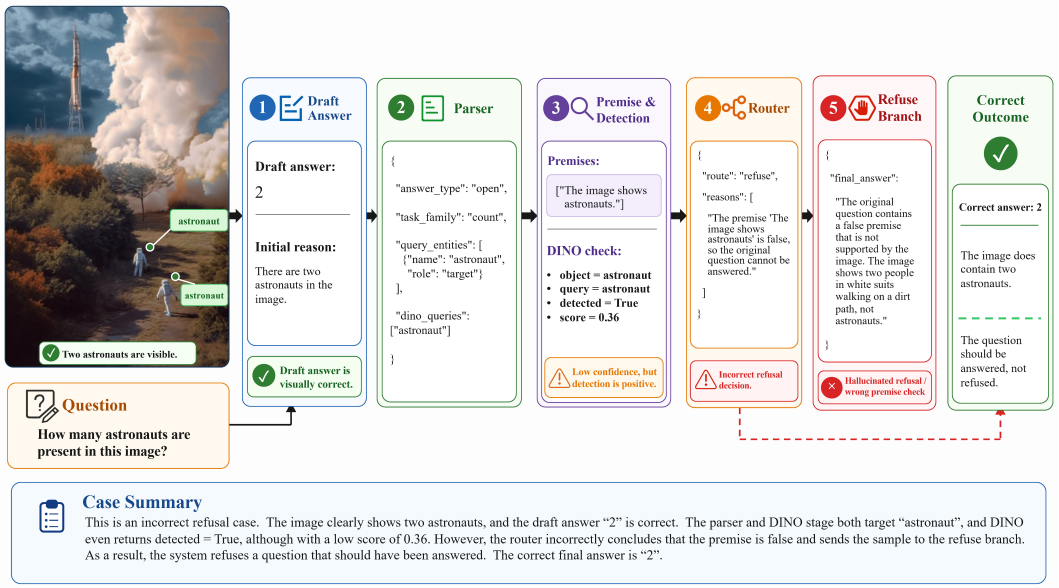

Figure 6: A genuine PAVER failure on a HaloQuest Visual Challenge example. Two astronauts are visible and the draft answer is correct, but the router incorrectly rejects the question despite a positive, low-confidence detection.

Figure 6 exposes an over-refusal failure. The image contains two astronauts, the draft correctly answers "2," and Grounding DINO returns a positive but low-confidence detection. Nevertheless, the router incorrectly judges the extracted premise to be contradicted and sends the example to the refusal branch. This case illustrates a real limitation of the framework: uncertainty in visual evidence and conservative premise assessment can reduce accuracy on questions dominated by difficult intrinsic visual perception. PAVER targets response reliability but does not strengthen the backbone's underlying visual capability, and its routing policy can therefore convert perceptual uncertainty into an incorrect refusal.

D EVALUATION PROTOCOL, BASELINE CONFIGURATIONS, AND PROMPTS

This appendix documents the baseline configurations used with LLaVA-1.5-7B and Qwen3-VL-8B-Instruct for MME-Hallucination, POPE, AMBER Generation, OpenCHAIR, HaloQuest, and JBA. It covers Vanilla, DeCo, SHIELD, VCD, and MARINE.

D.1 EVALUATION PROTOCOL AND INTERPRETATION

Evaluation settings. We evaluate the official benchmark splits using the benchmark-specific decoding configurations reported below. These tables specify sampling parameters, token budgets, and deterministic or stochastic decoding. Unless otherwise stated, the random seed is 42.

Length-controlled comparison. Vanilla and MARINE are regenerated with a shared length instruction and a 128-token budget, while PAVER's native outputs remain fixed. The comparison evaluates hallucination and coverage at similar mean response lengths. It does not match individual response lengths or independently vary length within PAVER.

Automatic judging and result granularity. HaloQuest and JBA responses are evaluated with `qwen3.7-flash` through each benchmark's official evaluation scripts and judge prompts. The reported scores are the outputs of this judge configuration, applied across the compared methods. HaloQuest results are reported by benchmark category as well as overall. The JBA tables report aggregate premise-sensitive performance across the evaluated split; these aggregate scores do not resolve differences among individual premise types.

D.2 QUESTION-SIDE HUMAN AUDIT PROTOCOL

We directly audit question-side premise extraction on a stratified 100-question JBA sample (50 true-premise and 50 false-premise) using LLaVA-1.5-7B. PAVER w/ ICL and w/o ICL are evaluated on the same questions; the latter removes demonstrations from both source-specific parsers while retaining all other components. Because the draft-side parser does not affect the extracted premise set, this comparison directly assesses question-side ICL. Premise sets are first checked by `qwen3.7-flash`, then reviewed by a single human annotator who is unaware of whether each output comes from the ICL or no-ICL condition. Outputs are anonymized and shuffled, and all final labels are human-confirmed. We report *Fully Correct*, *Under-extraction*, and *Over-extraction*, with *Query Leakage* as a diagnostic subset of Over-extraction indicating queried content incorrectly included as a premise.

The audit assesses extracted premise sets, rather than the correctness of HaloQuest or JBA judge decisions. Its percentages summarize the 100-question sample and provide a diagnostic view of premise selection and queried-content leakage.

D.3 BACKBONES AND COMMON SETTINGS

Backbones. We use `llava-hf/llava-1.5-7b-hf` and `Qwen/Qwen3-VL-8B-Instruct`. The LLaVA conversation template or Qwen chat template automatically inserts the image tokens and assistant-generation marker; the prompts below report only the user-visible text.

DeCo. We use `alpha=0.6`, `threshold_top_p=0.9`, `threshold_top_k=20`, `start_layer=20`, and `end_layer=29`.

VCD. We use `noise_step=500`, `cd_alpha=1.0`, and `cd_beta=0.1`.

MARINE. MARINE uses two additional visual models: `facebook/detr-resnet-50` for object detection and RAM++ with the `ram_plus_swin_large_14m.pth` checkpoint for image tagging. The DETR and RAM thresholds are 0.95 and 0.68, respectively. We take the case-insensitive exact intersection of the two output object lists and deduplicate it to construct the grounding guidance. Generation uses `guidance_strength=0.7`. For Qwen, the main prompt and

guidance prompt are decoded in two branches whose log probabilities are interpolated using the guidance strength. If the DETR–RAM++ intersection is empty, the original task prompt is retained without object guidance.

SHIELD. SHIELD additionally uses `openai/clip-vit-large-patch14-336`. CLIP computes caption-text and image features used to construct SHIELD's feature-enhancement and adversarial visual branches. We use three benchmark/backbone configurations:

1. For LLaVA on MME, AMBER, OpenCHAIR, HaloQuest, and JBA, we use the repository's lightweight wrapper with `cd_alpha=2.0` and `cd_beta=0.35`. Parameters not overridden by the launcher retain the wrapper defaults: `the=0.0`, `gamma_gain=1.0`, `gamma_reduce=0.0`, `gain_per=0.0`, `reduce_per=0.0`, `bias_weight=0.0`, and `bias_sample_num=32`. The CW parameters are `epsilon=0.14`, `steps=30`, `c=12`, and `lr=0.02`.
2. For LLaVA on POPE, we use SHIELD's original evaluation runner with `noise_step=999`, `cd_alpha=2.0`, `cd_beta=0.35`, `the=0.011`, `gamma_gain=3.0`, `gamma_reduce=3.0`, `gain_per=0.5`, `reduce_per=0.0`, `bias_weight=0.1`, and `bias_sample_num=32`. The CW parameters are `epsilon=0.14`, `steps=30`, `c=12`, and `lr=0.14`.
3. For Qwen, all tasks use `cd_alpha=2.0` and `cd_beta=0.35`. MME, POPE, HaloQuest, and JBA use `threshold=0.011`, `gamma_gain=3.0`, `gain_per=0.5`, `bias_weight=0.1`, `bias_sample_num=32`, and `bias_batch_size=4`, with CW `epsilon=0.14`, `steps=30`, `c=12`, and `lr=0.14`. AMBER and OpenCHAIR use `threshold=0.002`, `gamma_gain=3.0`, `gain_per=0.55`, `bias_weight=0.01`, `bias_sample_num=32`, and `bias_batch_size=4`, with CW `epsilon=0.14`, `steps=30`, `c=12`, and `lr=0.02`.

Unless otherwise stated, the random seed is 42. Qwen Vanilla uses deterministic greedy decoding and is therefore seed-independent.

D.4 Main-Task Prompts

D.4.1 MME-Hallucination

MME-Hallucination covers existence, count, position, and color, totaling 120 images and 240 yes/no questions. Vanilla, DeCo, VCD, and SHIELD use the following prompt; MARINE augments it with the guidance in Section D.5.

```
{question}
Answer based only on the image.  Respond with exactly one
word:  Yes or No.
```

D.4.2 POPE

We evaluate the random, popular, and adversarial splits. For all five Qwen methods, questions beginning with "There is/are" are normalized to "Is/Are there ... in the image?" before applying:

```
Answer the question based only on the image.
Question:  {normalized_question}
Answer yes or no only.
```

For LLaVA, every method uses the same question normalization followed by this JSON prompt:

```
You are a VQA answer generator.
Basic Principles:
- Answer using only what is visually grounded in the image.
- Do NOT use outside knowledge, category defaults, or
canonical object facts unless they are directly visible in
the image.
- Answer the question about the specific depicted
image/object/scene, not about a standard example of that
category.
- Keep the answer short.
- Provide a brief reason grounded in the image only (1-2 short
sentences).
Core Rules:
- Do not modify the original queried term(s) in the question.
- When the answer refers to the queried target, use the exact
original term from the question whenever possible.
- If your answer depends on identifying a different visible
object than the queried one, preserve the original queried
term where relevant.
- Do not let an observed alternative object replace the
original question object when both matter to the answer.
Question:
{normalized_question}
Return valid JSON only:
{ "reason": "...", "answer": "..." }
```

D.4.3 AMBER GENERATION

All 1,004 examples use the original query below. Vanilla, DeCo, VCD, and SHIELD use it directly; MARINE adds the guidance in Section D.5.

```
Describe this image.
```

D.4.4 OPENCHAIR

Vanilla, DeCo, VCD, and SHIELD use the prompt below; MARINE adds the guidance in Section D.5.

```
Describe this image in detail.
```

D.4.5 HALOQUEST

```
Answer the question based only on the image.
Question: {question}
Answer:
```

D.4.6 JBA

```
Answer the question based only on the image.
Question: {question}
Answer:
```

D.5 SHIELD First-Caption and MARINE Guidance Prompts

D.5.1 SHIELD first-caption prompt

All SHIELD configurations use:

```
Describe the image in as much detail as possible. First,
focus on the main objects, such as people or items in the
foreground. After describing the main objects, provide
details about the background, including natural environments
and other surroundings.
```

D.5.2 MARINE guidance prompts

When Qwen has non-empty DETR/RAM++ grounding objects, all six tasks use:

```
The detected objects in the image are: {object_list}. Use
these detected objects as grounding evidence and answer based
only on the image. {main_task_prompt}
```

LLaVA MARINE uses:

```
Detected visual objects: {object_list}. Treat the image
as final authority and use the detections only as supporting
evidence. {main_task_prompt}
```

D.6 Benchmark-Specific Decoding Configurations

In the following tables, *greedy* denotes `temperature=0` and `do_sample=False`; unreported beam counts are 1. We abbreviate `temperature`, `top_p`, `top_k`, and batch size as T, p, k, and b, respectively. Values of p and k are retained to document the configured arguments even when greedy decoding makes sampling inactive.

D.6.1 MME-Hallucination

All methods use `max_new_tokens=16` and the evaluation seed 42.

Table 16: Baseline decoding configurations for MME-Hallucination.

Backbone	Vanilla	DeCo	VCD	SHIELD	MARINE
LLaVA	greedy; $p = 1$	greedy; $p = 1$	greedy; $p = 1$	greedy; $p = 1$	greedy; $p = 0.9$; $b = 1$
Qwen	greedy	greedy; $p = 0.9, k = 0$	greedy; $p = 0.9, k = 0$	greedy; $p = 0.9, k = 0$	greedy; $p = 0.9, k = 0$

D.6.2 POPE

All methods use `max_new_tokens=128`.

Table 17: Baseline decoding configurations for POPE.

Backbone	Vanilla	DeCo	VCD	SHIELD	MARINE
LLaVA	greedy	greedy	$T = 1, p = 1$	$T = 1, p = 1$	$T = 0.6, p = 0.9$
Qwen	greedy	greedy; $p = 0.9, k = 0$	greedy; $p = 0.9, k = 0$	$T = 1, p = 1, k = 0$	greedy

D.6.3 AMBER GENERATION

Native baseline runs use `max_new_tokens=256`.

Table 18: Baseline decoding configurations for AMBER Generation.

Backbone	Vanilla	DeCo	VCD	SHIELD	MARINE
LLaVA	$T = 1, p = 0.9$	$T = 1, p = 0.9$	$T = 1, p = 1$	$T = 1, p = 1$	$T = 0.6, p = 0.9$; $b = 1$
Qwen	greedy	greedy; $p = 0.9, k = 0$	greedy; $p = 0.9, k = 0$	$T = 1, p = 1, k = 0$	greedy

D.6.4 OPENCHAIR

All methods use `max_new_tokens=256`.

Table 19: Baseline decoding configurations for OpenCHAIR.

Backbone	Vanilla	DeCo	VCD	SHIELD	MARINE
LLaVA	$T = 1, p = 1$	$T = 1, p = 1$	$T = 1, p = 1$	$T = 1, p = 1$	$T = 0.6, p = 0.9$; $b = 1$
Qwen	greedy	greedy; $p = 0.9, k = 0$	greedy; $p = 0.9, k = 0$	$T = 1, p = 1, k = 0$	greedy; $p = 0.9, k = 0$

D.6.5 HALOQUEST

All methods use `max_new_tokens=64`.

Table 20: Baseline decoding configurations for HaloQuest.

Backbone	Vanilla	DeCo	VCD	SHIELD	MARINE
LLaVA	greedy; $p = 1$	$T = 1, p = 0.9$	$T = 1, p = 1$	$T = 1, p = 1$	$T = 0.6, p = 0.9$; $b = 1$
Qwen	greedy	greedy; $p = 0.9, k = 0$	greedy; $p = 0.9, k = 0$	$T = 1, p = 1, k = 0$	greedy

D.6.6 JBA

All methods use `max_new_tokens=64`.

Table 21: Baseline decoding configurations for JBA.

Backbone	Vanilla	DeCo	VCD	SHIELD	MARINE
LLaVA	greedy; $p = 1$	$T = 1, p = 0.9$	$T = 1, p = 1$	$T = 1, p = 1$	$T = 0.6, p = 0.9$; $b = 1$
Qwen	greedy	greedy; $p = 0.9, k = 0$	greedy; $p = 0.9, k = 0$	$T = 1, p = 1, k = 0$	greedy

E Matched Ablation Control for Explicit Premise Representation

Matched inputs and inference stages. Answer-side Only is a matched control without explicit premise representation. It retains the same image I, initial draft $a^{(0)}$, draft-augmented grounding targets $\mathcal{U}_a$, Grounding DINO and Florence-2 evidence $\mathcal{E}$, standalone routing stage, branch-specific response prompts, and final decoding settings as Full PAVER. The extracted premise set $\mathcal{P}_q$ is omitted from the router and premise-conditioned response input. The original question remains available, allowing implicit premise reasoning.

The routing inputs differ as follows:

$$\begin{aligned} r_{\text{Full}} &= R_\theta(I, q, \mathcal{P}_q, \mathcal{E}; \pi_{\text{route}}), \\ r_{\text{AS}} &= R_\theta(I, q, \mathcal{E}; \pi_{\text{route}}). \end{aligned} \tag{11}$$

Both configurations retain the ANSWER and REFUSE branches. In the refusal branch, Answer-side Only uses the same response prompt with the explicit premise input omitted; the answer branch remains unchanged:

$$a^\star_{\text{AS}} = \begin{cases} M_\theta(I, q, \mathcal{E}; \pi_{\text{ref}}), & r_{\text{AS}} = \text{REFUSE}, \\ M_\theta(I, q, \mathcal{E}; \pi_{\text{ans}}), & r_{\text{AS}} = \text{ANSWER}. \end{cases} \tag{12}$$

Thus, the comparison tests the value of supplying extracted required premises to response control under matched visual evidence. It does not remove the question or prevent the model from inferring premises from its wording.

Observed component effects. On LLaVA-1.5-7B (Table 8), Full PAVER raises JBA FPC from 56.6 to 67.9 relative to Answer-side Only, an increase of 11.3 percentage points with the same draft-guided evidence. AMBER Hal remains 14.8, and MME-Hall increases from 653.3 to 658.3. These results support the contribution of explicit premise representation to premise-sensitive response behavior while preserving the observed answer-side hallucination performance. Because explicit premises enter both the router and the refusal response input, the comparison evaluates their combined role in response control.

The Question-side Only variant addresses a different component: it retains required premises and the router but removes draft-guided Grounding DINO evidence. Its FPC of 67.0 and Hal of 28.2 complement the matched-premise comparison by characterizing the contribution of the draft-guided pathway as a whole. The entity-level audit in Appendix B separately examines target coverage and final-response content within that pathway.

F IN-CONTEXT EXAMPLES FOR REQUIRED-PREMISE EXTRACTION

This appendix reproduces the question–output demonstrations for the two backbone-specific premise extractors: 4 for LLaVA and 25 for Qwen. Double braces are retained from the prompt templates. Example numbers are added for reference.

F.1 LLAVA PREMISE EXTRACTOR

Example 1

```
Q: What color are the cat's eyes?
A: {{"premises":["A cat is visible.",
                 "The cat has visible eyes."]}}
```

Example 2

```
Q: What is the town indicated by the vintage clock labeled CHICAGO?
A: {{"premises":["A vintage clock is visible.",
                 "The vintage clock is labeled CHICAGO."]}}
```

Example 3

```
Q: The train supports the track. How does the greenery affect the scene?
A: {{"premises":["A train is visible.","A track is visible.",
                 "The train supports the track.",
                 "Greenery is visible."]}}
```

Example 4

```
Q: Is there a cat?
A: {{"premises":"none"}}
```

F.2 QWEN PREMISE EXTRACTOR

Example 1

```
Question:
"What color are the cat's eyes?"
Output:
{{"premises":["A cat is visible.",
               "The cat has visible eyes."]}}
```

Example 2

```
Question:
"Is there a cat?"
Output:
{{"premises":"none"}}
```

Example 3

```
Question:
"What color is the umbrella?"
Output:
{{"premises":["An umbrella is visible."]}}
```

Example 4

```
Question:
"Is the road wet or dry?"
Output:
{{"premises":["A road is visible."]}}
```

Example 5

```
Question:
"What kind of hat is the monkey wearing?"
Output:
{{"premises":["A monkey is visible.",
               "The monkey is wearing a visible hat."]}}
```

Example 6

```
Question:
"What is written on the yellow placard with green letters?"
Output:
{{"premises":["A yellow placard is visible.",
               "The placard has green letters."]}}
```

Example 7

```
Question:
"Among the four players on the bench, what is the second player from the
right holding?"
Output:
{{"premises":["There are four players on a bench.",
               "A second player from the right is identifiable."]}}
```

Example 8

```
Question:
"What design does the horse's saddle have?"
Output:
{{"premises":["A horse is visible.",
               "The horse has a visible saddle."]}}
```

Example 9

```
Question:
"The blue bus is driving along a road, what kind of area is it passing
through?"
Output:
{{"premises":["A blue bus is visible.",
               "The blue bus is on a road.",
               "The blue bus is driving.",
               "The surrounding area is visible."]}}
```

Example 10

```
Question:
"A man wearing a green jacket is holding a cup. What color is the cup?"
Output:
{{"premises":["A man wearing a green jacket is visible.",
               "The man is holding a visible cup."]}}
```

Example 11

```
Question:
"The dog is sitting beside the woman. What is the woman holding?"
Output:
{{"premises":["A dog is visible.","A woman is visible.",
               "The dog is beside the woman.",
               "The dog is sitting."]}}
```

Example 12

```
Question:
"The train supports the track. How does the greenery affect the scene?"
Output:
{{"premises":["A train is visible.","A track is visible.",
               "The train supports the track.",
               "Greenery is visible."]}}
```

Example 13

```
Question:
"What is the town indicated by the vintage clock labeled CHICAGO?"
Output:
{{"premises":["A vintage clock is visible.",
               "The vintage clock is labeled CHICAGO."]}}
```

Example 14

```
Question:
"The woman is excited. What might she be planning as she applies
lipstick?"
Output:
{{"premises":["A woman is visible.","The woman is excited.",
               "The woman is applying lipstick."]}}
```

Example 15

```
Question:
"Why did the person run after the kite had already flown high?"
Output:
{{"premises":["A person is visible.","A kite is visible.",
               "The person is running.","The kite has flown high.",
               "The person ran after the kite had flown high."]}}
```

Example 16

```
Question:
"Is the man wearing a red shirt?"
Output:
{{"premises":["A man is visible."]}}
```

Example 17

```
Question:
"Is the man holding an umbrella?"
Output:
{{"premises":["A man is visible."]}}
```

Example 18

```
Question:
"Is there a man holding an umbrella?"
Output:
{{"premises":"none"}}
```

Example 19

```
Question:
"Is the man beside the red car holding an umbrella?"
Output:
{{"premises":["A man is visible.","A red car is visible.",
               "The man is beside the red car."]}}
```

Example 20

```
Question:
"Is the man's umbrella open?"
Output:
{{"premises":["A man is visible.",
               "The man has a visible umbrella."]}}
```

Example 21

```
Question:
"What color is the man's umbrella?"
Output:
{{"premises":["A man is visible.",
               "The man has a visible umbrella."]}}
```

Example 22

```
Question:
"Is the dog next to the woman?"
Output:
{{"premises":["A dog is visible.","A woman is visible."]}}
```

Example 23

```
Question:
"What is the dog next to the woman looking at?"
Output:
{{"premises":["A dog is visible.","A woman is visible.",
               "The dog is next to the woman."]}}
```

Example 24

```
Question:
"Does the woman have a hat?"
Output:
{{"premises":["A woman is visible."]}}
```

Example 25

```
Question:
"What color is the woman's hat?"
Output:
{{"premises":["A woman is visible.",
               "The woman has a visible hat."]}}
```

G GROUNDING-QUERY EXTRACTION EXAMPLES

The same six examples below are used across all six evaluation benchmarks for entity-level query extraction from the question and draft. Examples 1–4 cover visual questions, and Examples 5–6 cover caption requests. All six are included together regardless of the evaluation task. Only the `dino_queries` field is shown. Each bad output is a separate alternative.

Example 1

Question: Is there a red car present in the image?
Draft: No.

Good Output:

```
{"dino_queries": ["red car"]}
```

Bad Output:

```
{"dino_queries": ["car"]}
{"dino_queries": ["is car present"]}
{"dino_queries": ["image"]}
```

Example 2

Question: Are there any cars in the image?
Draft: No.

Good Output:

```
{"dino_queries": ["car"]}
```

Bad Output:

```
{"dino_queries": ["are there any car"]}
{"dino_queries": ["are there any cars in image"]}
{"dino_queries": ["image"]}
```

Example 3

Question: What is the man holding a newspaper doing?
Draft: He is sitting on a wooden bench beside a bicycle.

Good Output:

```
{"dino_queries": ["man", "newspaper", "wooden bench", "bicycle"]}
```

Bad Output:

```
{"dino_queries": ["wooden bench", "bicycle"]}
{"dino_queries": ["man", "newspaper", "sitting"]}
{"dino_queries": ["what is the man doing"]}
```

Example 4

Question: What color is the umbrella above the table?
Draft: It is red, and two women are sitting beneath it.

Good Output:

```
{"dino_queries": ["red umbrella", "table", "women"]}
```

Bad Output:

```
{"dino_queries": ["red"]}
{"dino_queries": ["women"]}
{"dino_queries": ["umbrella above the table"]}
```

Example 5

Question: Describe the image.
Draft: A man wearing an orange shirt is riding a bicycle past a wooden bench. A small dog is running beside him.

Good Output:

```
{"dino_queries": ["man", "orange shirt", "bicycle", "wooden bench",
  "dog"]}
```

Bad Output:

```
{"dino_queries": ["orange", "riding", "running"]}
{"dino_queries": ["image"]}
```

Example 6

Question: Describe the image.
Draft: Two women are sitting at a table with pizza and glasses. A red umbrella stands above them.

Good Output:

```
{"dino_queries": ["women", "table", "pizza", "glasses", "red umbrella"]}
```

Bad Output:

```
{"dino_queries": ["sitting", "above"]}
{"dino_queries": ["scene"]}
```