CARE-FLOW: MITIGATING INTER-IMAGE HALLUCINATION IN MULTI-IMAGE MLLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Hallucination remains a major reliability challenge for multimodal large language models (MLLMs), and becomes more complex when reasoning across multiple images. Even when visual content remains grounded, models may select irrelevant evidence, omit required evidence, or incorrectly combine supported evidence. We call such failures in using grounded evidence across images inter-image hallucinations. Standard multimodal inference entangles determining what evidence a question requires, acquiring that evidence from the images, and forming the response, making it difficult to establish which visual information actually drives the inference. Because observed association cannot establish such dependence, the use of evidence must be tested through intervention. We propose CARE-Flow, a causal framework that decouples evidence specification and acquisition from response formation through an explicit evidence mediator. This mediator makes image-specific contributions independently testable before response generation and constrains downstream inference to the established evidence, yielding a constructed conditional front-door structure for tracing visual influence. Existing multi-image benchmarks mainly score final answers and provide limited diagnosis of how evidence use fails. We therefore introduce CARE-Bench, a 12,000-question benchmark that diagnoses failures in evidence formation and combination across multiple visual domains. Across three MLLMs and seven multi-image benchmarks, CARE-Flow consistently improves accuracy, outperforming the strongest baselines by 4.28–7.53 percentage points on CARE-Bench.

1 INTRODUCTION

Hallucination has remained a persistent reliability challenge for large language models (LLMs), which may generate fluent yet fabricated or unsupported content. Extending LLMs with visual perception gives multimodal large language models (MLLMs) access to visual evidence, but also introduces the requirement that generated content be grounded in that evidence, leading to failures such as nonexistent objects or incorrect attributes Li et al. (2023); Wang et al. (2024); Guan et al. (2024). Extending single-image MLLMs to multiple visual inputs enables tasks such as cross-view comparison and event-sequence understanding, but introduces a further requirement: models must compare, relate, and integrate information across images. Multi-image reasoning, therefore, requires more than individually grounded visual evidence, motivating a closer examination of how errors arise during cross-image inference.

Multi-image inference depends on two aspects: forming the visual evidence required by the question and combining evidence to reach the answer. Evidence formation can fail in three ways: the model may retrieve visual evidence present in the input but not required by the question; it may introduce visual evidence unsupported by any input image, resulting in conventional visual hallucination; or it may retrieve required evidence incompletely. Evidence combination can also fail when the required visual evidence is retrieved correctly and completely, but is related inconsistently to the question. Together, these four cases cover the primitive failures in the use of visual evidence, with compound errors arising from their combination. Existing grounding-based approaches primarily mitigate conventional visual hallucinations, in which generated content is unsupported by input visual evidence Leng et al. (2024). But such grounding checks cannot distinguish the other three failures, where the relevant evidence remains grounded but its cross-image use does not faithfully support the inference required by the question; we refer to these as *inter-image hallucinations*.

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.

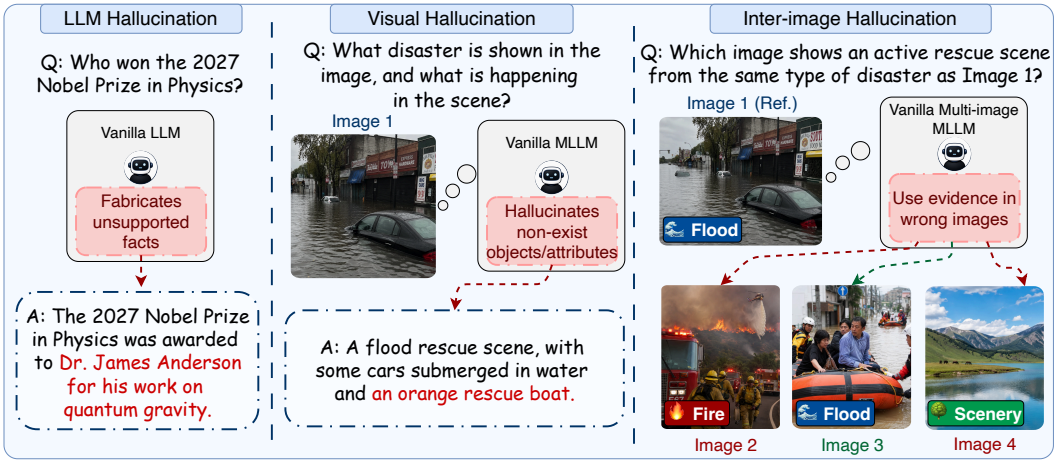


Figure 1: Inter-image hallucination. A text-only LLM may fabricate unsupported content, while a single-image MLLM may hallucinate an absent object such as an "orange rescue boat." In multi-image inference, the evidence itself may be grounded yet incorrectly used: the model correctly recognizes the fire-rescue scene in Image 2, but selects it for a question requiring a rescue scene of the same disaster type as the flood in Image 1.

Inter-image hallucinations shift the problem from whether visual evidence is supported to whether it actually drives the inference. Observed association alone cannot establish such dependence; determining whether it is faithful requires examining whether the inference changes when a visual contribution is altered. Such intervention is informative only when two stages that determine visual evidence use are kept distinct: (1) during question interpretation, all required visual evidence and its intended combination must be specified, providing a fixed reference for detecting when required visual evidence is retrieved incompletely or combined inconsistently with the question; and (2) during visual evidence acquisition, each required evidence must be linked to the image that actually contributes to it, preventing the use of visual evidence that is present in the input but not required by the question or unsupported by any input image. Standard multimodal inference entangles question interpretation and visual evidence acquisition with response generation, preventing the required evidence structure and image-specific contributions from being isolated and independently tested.

To address this, we propose CARE-Flow, a causal framework that separates evidence specification and acquisition from downstream response formation. By introducing an explicit evidence mediator, CARE-Flow turns internal evidence use into an explicit intermediate representation that can be fixed and independently tested before response formation. CARE-Flow first specifies the evidence required by the question and its intended combination, thereby providing a fixed reference for subsequent reasoning. It then acquires source-indexed atomic evidence for these requirements and uses image-specific interventions to test whether each evidence item actually depends on its declared source. Finally, the original visual states are removed, preventing downstream generation or verification from bypassing the mediator and introducing alternative visual evidence. Together, these steps form a constructed conditional front-door structure, allowing visual influence on the response to be traced through evidence whose source dependence has been explicitly established. Existing multi-image benchmarks mainly report final-answer correctness and provide limited diagnosis of where evidence use fails. We therefore introduce CARE-Bench, a 12,000-question diagnostic benchmark that covers evidence-formation and combination failures across multiple visual domains, along with an automated construction pipeline.

Our contributions are threefold. First, we characterize multi-image hallucination through failures in visual evidence formation and combination, providing a systematic account of how grounded evidence can still lead to incorrect inference and identifying the entanglement of question interpretation and visual evidence acquisition with response generation as a key source of these failures. Second, we propose CARE-Flow, which introduces an explicit evidence mediator that turns internal evidence use into a fixed and testable intermediate representation, allowing image-specific contributions to be causally examined before response formation and ensuring that downstream inference proceeds only

through the established evidence. Third, we introduce CARE-Bench, a 12,000-question diagnostic benchmark covering inter-image hallucinations across multiple visual domains. We further describe its automatic construction pipeline. Extensive experiments across three MLLMs demonstrate consistent improvements with CARE-Flow, while comprehensive ablation, diagnostic, and qualitative analyses further validate the effectiveness of its key design choices.

2 Related Work

Multi-Image Hallucination. Recent work has begun to identify hallucination patterns specific to multiple images. Hallucination is commonly defined as source-unfaithful generation, including both contradictions and unsupported content, while MLLM evaluations also cover attribute and relation errors Ji et al. (2023); Maynez et al. (2020); Wang et al. (2024). CcDPO Li et al. (2025b) characterizes context omission, context conflation, and detail misinterpretation in using image-specific information. MIHBench studies object existence, counting, and identity consistency, including effects of image number and ordering Li et al. (2025a). MIOH Min et al. (2026) further covers existence, counting, attributes, and positions under different reasoning patterns. These studies show that multi-image hallucination extends beyond errors on individual images, but mainly describe failures through tasks or observable behaviors. Our CARE-Flow instead characterizes grounded failures by how visual evidence is used, distinguishing incorrect evidence selection, missing required evidence, and incorrect evidence combination.

Mitigating Multi-Image Hallucination. Existing mitigation methods mainly improve how models attend to or integrate information across images. CcDPO uses hierarchical preference optimization to reduce omission, conflation, and misinterpretation Li et al. (2025b). MIHBench proposes Dynamic Attention Balancing to redistribute attention across images Li et al. (2025a), while CAPL combines cross-image token interaction with preference learning to strengthen information exchange and visual grounding Yang et al. (2026). These approaches improve the use of cross-image information through training or attention adjustments, but do not explicitly test whether the evidence used in an inference actually depends on its claimed visual source. CARE-Flow instead introduces an explicit evidence mediator and uses image-specific interventions to test source dependence before response formation, while removing direct visual access afterwards so that downstream inference proceeds only through the established evidence.

Evaluating Multi-Image Hallucination. General multi-image benchmarks such as MuirBench, MIRB, MMIU, and MIBench evaluate perception, comparison, multi-hop reasoning, and multimodal in-context learning, but mainly report final-answer correctness Wang et al. (2025); Zhao et al. (2024); Meng et al. (2025); Liu et al. (2024). Hallucination-specific benchmarks provide finer diagnosis: MIHBench targets object existence, counting, and identity consistency Li et al. (2025a), while MIOH combines multiple object-level tasks with different reasoning patterns and adversarial pressures Min et al. (2026). However, they still organize errors mainly by task or response behavior rather than by how visual evidence use fails. CARE-Bench instead directly evaluates incorrect evidence selection, missing required evidence, and incorrect evidence combination across real and controlled visual domains.

3 Problem Definition

Multi-image evidence structure. Let $V = \{I_1, \ldots, I_K\}$ denote the input images and Q a question or instruction. A frozen MLLM generates $Y = (y_1, \ldots, y_T)$ autoregressively as $p_\theta(Y \mid V, Q) = \prod_t p_\theta(y_t \mid V, Q, y_{<t})$. Each image I_i supports a set of atomic visual evidence $\mathcal{A}_i$, giving the visual evidence universe $\mathcal{A} = \bigcup_i \mathcal{A}_i \subseteq \mathcal{U}$, where $\mathcal{U}$ contains all expressible atomic propositions.

The evidence-use failures introduced in Section 1 can be formalized by separating *what evidence a question requires* from *how that evidence must be combined*. We represent the former by a tuple of question-relevant evidence operands $A_Q^* = (a_1^*, \ldots, a_m^*)$, where each position corresponds to a required evidence role, and the latter by a question-specified operator r_Q^*. Correct inference is therefore $f_Q^* = r_Q^*(A_Q^*)$. Incorrect evidence selection or missing required evidence changes the operands available to the inference, whereas incorrect evidence combination changes the operator applied to them. The operator may be a single relation or a composite inference represented by a Directed Acyclic Graph (DAG). When the atomic fact set and answer space are finite, any deter-

ministic question–answer mapping can, in principle, be represented by such a DAG; Appendix A.3 provides the proof. The model analogously induces $\widehat{f}_Q = \widehat{r}_Q(\widehat{A}_Q)$, with $\widehat{A}_Q = (\widehat{a}_1, \ldots, \widehat{a}_m)$ and $\widehat{a}_j \in \mathcal{U} \cup \{\perp\}$.

Unified hallucination formulation. An *evidence-operand error* occurs when $\widehat{A}_Q \not\equiv A^*_Q$, while an *evidence-operator error* occurs when $\widehat{r}_Q \not\equiv r^*_Q$; both may occur simultaneously. For each required evidence role, operand errors decompose as

$$\widehat{a}_j \begin{cases} \in \mathcal{A},\ \widehat{a}_j \neq a^*_j, & \text{grounded substitution,} \\ \in \mathcal{U} \setminus \mathcal{A}, & \text{ungrounded substitution,} \\ = \perp, & \text{omission.} \end{cases}$$

Grounded substitution formalizes incorrect evidence selection: the evidence is visually supported but does not satisfy the question-required role. Ungrounded substitution introduces evidence that is unsupported by any input image, thereby recovering conventional visual hallucination. Omission leaves a required role unfilled, while an operator error combines evidence inconsistently with Q. We use *inter-image hallucination* for grounded operand substitution, omission, and operator error, all of which may occur even when the evidence remains visually grounded.

Constructed front-door formulation. The formulation above specifies what evidence and composition are required, but not whether the evidence actually depends on its assigned visual source. Let X denote image-indexed visual information paths, M an explicit evidence mediator, and Y the response. Grounding alone cannot determine which image actually contributes to M. CARE-Flow, therefore, constructs the question-conditioned path $X \to M \to Y$: M is fixed before response formation, image-specific interventions test its dependence on X, and a hard reset removes direct visual access before downstream inference. Consequently, $Y \perp X \mid M, Q$, so visual influence must pass through the established mediator. This constructed conditional front-door structure makes evidence-source dependence testable while preventing downstream inference from bypassing the established evidence.

4 CARE-FLOW

CARE-Flow operationalizes the operand–operator structure in Section 3 through three stages. SPECIFY fixes the required evidence roles and operator, CAUSALLY CERTIFY tests the source-specific visual dependence of proposed operands, and COMPOSE applies the fixed operator to certified evidence before generation or verification. CARE-Bridge mediates the resulting evidence flow.

4.1 PRE-RESPONSE EVIDENCE STRUCTURE SPECIFICATION

CARE-Flow first derives from Q an answer-free evidence specification $\mathcal{S}_Q = (R_Q, r_Q)$, where $R_Q = (\rho_1, \ldots, \rho_J)$ contains the required evidence roles and r_Q specifies how those roles must be composed. Composite operators follow the DAG representation introduced in Section 3. Fixing $\mathcal{S}_Q$ before response formation provides an explicit reference for detecting missing operands and question-inconsistent composition. Conditioned on V, Q, and $\mathcal{S}_Q$, the frozen MLLM fills the required roles with a provisional CARE-Bridge

$$\widetilde{M} = \widetilde{\mathcal{E}} = \{\widetilde{e}_j = (\rho_j, S_j, c_j)\}_{j=1}^{J}, \qquad F_j(\mathbf{1}) = \frac{1}{T_j} \sum_{t=1}^{T_j} \log p_\theta(c_{j,t} \mid V, Q, c_{j,<t}; \mathbf{G} = \mathbf{1}),$$

where c_j is an atomic evidence operand and $S_j \subseteq \{1, \ldots, K\}$ is its declared source set. Each operand is scored in an independent teacher-forced branch so that previously generated evidence cannot replace visual support for later operands.

Cross-image questions are decomposed into source-specific operands whenever possible. A temporal question may therefore yield ($\{1\}$, "the cup is on the table") and ($\{2\}$, "the cup is on the shelf"), while r_Q specifies that their positional change must later be computed. Cross-image facts that cannot be decomposed without losing their visual meaning, such as cross-view identity correspondence, may retain a multi-image source set. CARE-Flow fixes $\mathcal{S}_Q$ and $\widetilde{M} = \mathrm{Propose}_\theta(V, Q, \mathcal{S}_Q)$ before any response-specific content is introduced. Response generation, therefore, cannot alter the required evidence roles or operator.

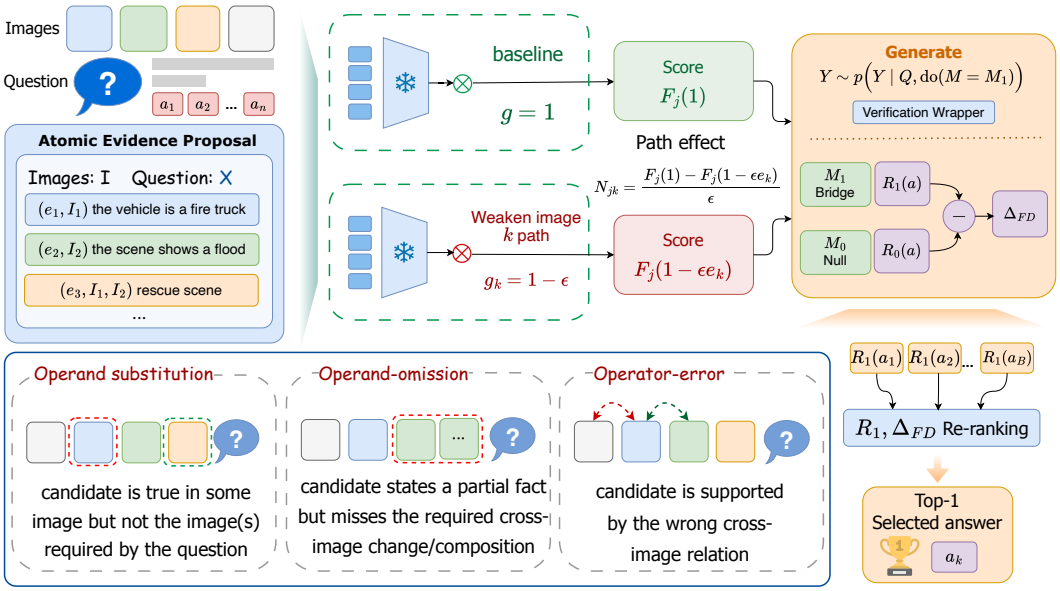


Figure 2: Overview of CARE-Flow. CARE-Flow first specifies the required evidence structure and proposes source-indexed atomic operands, then causally certifies each operand through image-path interventions. Certified evidence forms M_1, after which the fixed question-specified operator is applied before generation or verification. A matched null mediator M_0 controls linguistic compatibility.

4.2 Source-Grouped Intervention and Operand Certification

A declared source set S_j in $\widetilde{M}$ is only a model-generated claim. CARE-Flow tests whether Image k actually contributes to operand c_j by intervening on its outgoing attention value-output paths.

Let $\mathcal{V}_k$ contain the token positions of Image k. At layer ℓ, define the contribution from Image k to receiving position t as

$$c^{(\ell)}_{t \leftarrow I_k} = \sum_h W_O^{(\ell,h)} \sum_{s \in \mathcal{V}_k} \alpha_{ts}^{(\ell,h)} v_s^{(\ell,h)}.$$

Proposition 1: Exact source-grouped attention decomposition. Let $c^{(\ell)}_{t \leftarrow \text{other}}$ contain the bias and all non-visual contributions. Then

$$o_t^{(\ell)} = c^{(\ell)}_{t \leftarrow \text{other}} + \sum_{k=1}^{K} c^{(\ell)}_{t \leftarrow I_k},$$

$$\widetilde{o}_t^{(\ell)}(\mathbf{g}) = c^{(\ell)}_{t \leftarrow \text{other}} + \sum_{k=1}^{K} g_k c^{(\ell)}_{t \leftarrow I_k}, \qquad \frac{\partial \widetilde{o}_t^{(\ell)}}{\partial g_k} = c^{(\ell)}_{t \leftarrow I_k}.$$

Exactness follows because the gate is applied after softmax and value aggregation, leaving the attention weights at the intervened layer unchanged. Proof appears in Appendix A.1.

A local intervention sets $g_k = 1 - \epsilon$ while keeping all other gates at one. CARE-Flow applies the gate at every supported full-attention block and every receiving position outside $\mathcal{V}_k$, thereby weakening the outgoing path set $\mathcal{P}_k = \{I_k \to t : t \notin \mathcal{V}_k\}$. Post-aggregation gating avoids the normalization and attention-competition effects introduced by directly scaling visual embeddings Vaswani et al. (2017); Zhang & Sennrich (2019). A single intervention strength ϵ is fixed across models and datasets and is not tuned on downstream performance.

CARE-Flow measures the necessity of Image k for operand j by

$$N_{jk}^{(\epsilon)} = \frac{F_j(\mathbf{1}) - F_j(\mathbf{1} - \epsilon \mathbf{e}_k)}{\epsilon}.$$

Certification compares the weakest declared-source contribution with the strongest competing contribution:

$$D_j = \min_{k \in S_j} N_{jk}^{(\epsilon)} - \max\left(0, \max_{r \notin S_j} N_{jr}^{(\epsilon)}\right), \qquad \mathrm{Cert}(\widetilde{e}_j) = \mathbb{I}[D_j > 0],$$

where the competing-source term is defined as zero when S_j contains all images. A positive D_j requires every declared source to contribute positively and more strongly than any undeclared source with positive influence. Failed operands remain in their required roles with value `unknown`; for example, if the operator compares two operands and one cannot be certified, that role is kept as `unknown`, preserving the intended operator–role structure instead of changing the computation by silently dropping missing evidence. Certification yields the CARE-Bridge $M_1 = \mathcal{E}^\star = \mathrm{Certify}(\widetilde{\mathcal{E}}, N)$, where $N = \{N_{jk}^{(\epsilon)}\}_{j,k}$ denotes the full set of necessity scores.

Operand certification establishes which visual evidence enters CARE-Bridge and from which sources. Operator control uses the independently fixed r_Q, which is applied only after certification and only to the certified operand roles in M_1.

4.3 Constructed Front-Door Composition and Response Inference

Certification controls what enters CARE-Bridge, but the original multimodal states may still contain uncaptured visual information. CARE-Flow therefore discards the images, visual tokens, multimodal hidden states, intervention states, and all key/value caches. A new pure-text computation receives only Q, the fixed operator r_Q, and the certified CARE-Bridge M_1. Original visual content can therefore no longer introduce alternative operands or bypass the fixed operator during subsequent inference.

CARE-Flow next explicitly executes the fixed operator over the certified operands,

$$z_1 = \mathrm{Compose}_\theta(r_Q, M_1),$$

where z_1 is the resulting composition. Simple operators are executed directly, while composite operators follow their specified DAG. Missing or `unknown` operands propagate to an unresolved composition instead of being replaced by alternative visual evidence. Generation and verification, therefore, operate on an inference whose operands and operator have both been fixed before response formation.

Theorem 1: Constructed front-door mediation. Fix θ and $Q = q$, and let $M = f_M(X, q, U_M)$, $Z = f_C(M, r_q, U_C)$, and $Y = f_Y(Z, q, U_Y)$, where X is the image-indexed visual path state. Assume that M is completed before response-specific content is introduced; the response stage receives no image-derived state outside M; and U_M is independent of (U_C, U_Y), or all stages use deterministic decoding. Then

$$Y \perp X \mid M, Q, \qquad p_\theta(y \mid do(X = x), q) = \sum_m p_\theta(m \mid do(X = x), q) p_\theta(y \mid do(M = m), q),$$

$$p_\theta(y_t \mid do(X = x), M = m, q, y_{<t}) = p_\theta(y_t \mid do(M = m), q, y_{<t}).$$

Proof appears in Appendix A.2.

Complete mediation follows from the constructed computation, since the composition result Z lies downstream of M and Q. CARE-Flow does not claim that arbitrary natural-language rationales satisfy the observational front-door criterion. Instead, CARE-Flow creates an interventional factorization whose two legs are executable: image-path gates probe $X \to M$, and mediator replacement propagates through $M \to Z \to Y$. Accordingly, Theorem 1 establishes causal fidelity of the constructed computation.

After composition, CARE-Flow supports both open-ended generation and complete-response verification. For open-ended inference, $\widehat{Y} \sim p_\theta(Y \mid Q, M_1, z_1)$. Response prefixes may influence subsequent language generation, but cannot redirect visual retrieval or replace the fixed operator because no original visual computation remains.

For verification, a complete response a is evaluated against the certified operands and their composition result using

$$R_1(a) = \log p_\theta(\texttt{Yes} \mid Q, M_1, z_1, a) - \log p_\theta(\texttt{No} \mid Q, M_1, z_1, a).$$

Verification asks whether a is fully supported by the specified evidence structure; substituted operands, omitted evidence, incorrect composition, and uncertified evidence favor `No`.

To isolate the contribution of certified evidence, CARE-Flow constructs a matched null mediator M_0 with the same roles, sources, ordering, and structure as M_1, while replacing factual values with `unavailable`. Applying the same fixed operator gives $z_0 = \mathrm{Compose}_\theta(r_Q, M_0)$. Define

$$R_0(a) = \log p_\theta(\texttt{Yes} \mid Q, M_0, z_0, a) - \log p_\theta(\texttt{No} \mid Q, M_0, z_0, a),$$
$$\Delta_{\mathrm{FD}}(a) = R_1(a) - R_0(a), \qquad \mathcal{H}_{\mathrm{valid}} = \{a \in \mathcal{H} : R_1(a) > 0,\ \Delta_{\mathrm{FD}}(a) > 0\}.$$

A positive $R_1(a)$ requires the specified evidence structure to favor `Yes` over `No`, while a positive $\Delta_{\mathrm{FD}}(a)$ requires certified factual evidence to increase support relative to the matched null mediator. CARE-Flow returns the valid response with maximum R_1; an empty valid set triggers abstention or regeneration from the CARE-Bridge. Algorithm 1 in the appendix summarizes the procedure.

5 CARE-BENCH

CARE-Bench targets inter-image hallucination by evaluating whether an MLLM follows the evidence structure required by each multi-image question, in addition to producing the correct answer. Each instance is constructed from verified atomic evidence and a specified composition, which determines the question, reference answer, and controlled incorrect choices. Instances contain three to five related images with a unique evidence-supported answer. Captions and metadata support retrieval and organisation only, not ground-truth visual evidence.

Three complementary domains balance visual diversity and evidence control. COCO-VQA covers common objects, activities, attributes, and everyday scenarios, while New Zealand museum collections extend coverage to less familiar objects, species, artefacts, and photography. Because real images lack a complete inventory of atomic evidence, a third controlled domain generates images from predefined facts, enabling direct evaluation of fact extraction and separation of evidence-acquisition from reasoning errors. For the two real-image domains, Gemma 3 27B, an open-source multimodal model excluded from evaluation, extracts visually verifiable facts, which an independent closed-source LLM-as-a-Judge validates. Controlled error injection further tests judge reliability using known operand, operator, source, and answer-validity corruptions.

Three diagnostic errors instantiate the operand–operator formulation. *Operand Substitution* replaces a required role with another grounded fact; *Operand Omission* leaves a role unfilled; and *Operator Error* uses a composition that is inconsistent with the question. Required operands can therefore be replaced or omitted, while remaining mismatches after operand alignment lie in the operator. Final quality control checks answer uniqueness, evidence sufficiency, source attribution, error validity, and shortcuts. Further details appear in Appendix C.

6 EXPERIMENTS

We evaluate CARE-Flow from four aspects. We first describe the setup and compare it with existing hallucination-mitigation methods across multiple backbones and multi-image benchmarks. We then analyze its effects on operand substitution, operand omission, and operator errors, along with diagnostic analyses, selective reliability, and ablation studies.

6.1 SETUP

Models and baselines. We evaluate CARE-Flow on LLaVA-NeXT-7B, Qwen3.5-9B, and InternVL3-8B, keeping all backbones frozen. We compare against vanilla inference, a token-level multi-image enhancement method, DTS Lee et al. (2026), and four hallucination-mitigation baselines: FOCUS Park et al. (2025), VCD Leng et al. (2024), OPERA Huang et al. (2024), and DAB Li et al. (2025a). All methods use the same backbone and benchmark-specific input format.

Benchmarks. We evaluate on six public multi-image benchmarks, MMIU Meng et al. (2025), MuirBench Wang et al. (2025), MIBench Liu et al. (2024), Mantis Jiang et al. (2024), MIHBench Li et al. (2025a), and MIRB Zhao et al. (2024), together with our CARE-Bench. We report accuracy as mean $\pm$ standard deviation over five runs.

Table 1: Main results on multi-image benchmarks. All numbers are accuracy (%).

Model	Method	MMIU	MuirBench	MIBench	Mantis	MIHBench	MIRB	CARE-Bench
LLaVA-NeXT-7B	Vanilla	31.42±.49	35.04±.33	32.67±.23	62.21±.69	66.85±.47	47.88±.59	27.72±.66
	DTS	32.49±.77	37.81±.76	33.19±.58	64.40±.55	68.91±.43	48.57±.32	31.87±.02
	FOCUS	32.71±.82	35.35±.77	33.78±.68	64.06±.33	66.95±.51	48.19±.35	28.56±.21
	VCD	31.88±.24	35.98±.44	32.81±.42	63.58±.63	67.23±.49	48.12±.86	29.82±.60
	OPERA	32.95±.71	35.27±.83	33.15±.81	62.47±.45	67.17±.61	47.91±.53	29.47±.42
	DAB	32.13±.28	35.61±.52	33.42±.30	63.91±.76	67.80±.57	48.17±.41	29.31±.73
	CARE-Flow	**33.00±.87**	**38.12±.35**	**34.89±.62**	**65.01±.54**	**70.04±.85**	**49.30±.84**	**39.40±.33**
Qwen3.5-9B	Vanilla	40.59±.63	45.23±.61	41.42±.85	68.66±.79	70.12±.78	63.57±.39	41.78±.51
	DTS	42.09±.27	46.32±.03	43.76±.42	69.35±.19	71.40±.11	64.17±.49	44.09±.53
	FOCUS	42.31±.22	47.15±.28	43.18±.52	69.12±.43	70.18±.32	63.05±.66	42.46±.83
	VCD	40.88±.54	47.68±.67	41.67±.75	69.05±.88	70.04±.33	62.94±.29	42.03±.34
	OPERA	42.76±.79	46.11±.50	42.95±.36	68.74±.31	70.32±.40	62.61±.75	41.83±.68
	DAB	41.53±.40	45.54±.89	42.11±.65	68.89±.61	70.86±.58	62.73±.58	42.76±.28
	CARE-Flow	**43.07±.30**	**47.99±.44**	**45.65±.32**	**71.01±.38**	**72.38±.76**	**64.86±.24**	**50.54±.79**
InternVL3-8B	Vanilla	29.78±.46	36.28±.72	31.08±.47	67.28±.27	69.30±.75	45.38±.82	29.20±.57
	DTS	31.86±.29	37.51±.83	32.54±.54	68.31±.34	70.68±.58	46.86±.81	32.47±.14
	FOCUS	31.98±.84	36.92±.38	31.91±.80	69.12±.72	69.87±.71	46.27±.48	29.98±.70
	VCD	30.84±.58	36.30±.77	31.28±.21	68.74±.49	69.46±.53	45.93±.70	29.41±.81
	OPERA	30.26±.26	36.67±.26	31.37±.56	68.73±.78	69.42±.40	45.38±.37	29.97±.24
	DAB	30.49±.74	36.45±.56	31.16±.86	68.96±.25	70.25±.80	45.46±.89	30.76±.54
	CARE-Flow	**33.02±.64**	**37.85±.82**	**33.02±.40**	**69.78±.68**	**72.98±.33**	**47.29±.47**	**36.75±.42**

Table 2: CARE-Bench errors (%). Sub., Omit., and Op. denote operand substitution, operand omission, and operator errors, respectively.

Model	Method	Sub.↓	Omit.↓	Op.↓	Other↓
LLaVA	Vanilla	33.18	16.07	22.33	0.70
	DTS	31.91	15.56	19.96	0.70
	CARE-Flow	29.36	15.52	15.08	0.65
Qwen	Vanilla	18.49	30.33	8.79	0.61
	DTS	18.25	28.60	8.46	0.60
	CARE-Flow	16.24	26.44	6.18	0.59
InternVL	Vanilla	22.34	33.92	13.78	0.76
	DTS	22.03	31.89	12.85	0.76
	CARE-Flow	21.40	31.40	9.70	0.75

Table 3: CARE-Flow ablation (%). Macro-average over three backbones. Avg. is the unweighted benchmark mean.

Variant	Muir	MIB	CARE	Avg.
CARE-Flow	41.32	37.85	42.23	40.47
w/o Cert.	31.57	28.43	35.26	31.75
Attn. Cert.	36.84	33.62	40.53	37.00
w/o Contrast	39.26	35.74	40.18	38.39
w/o Reset	34.69	32.18	38.74	35.20
w/o Δ_{FD}	39.08	35.91	39.87	38.29

Implementation details. The same frozen MLLM performs the whole CARE-Flow process. Failed operands retain their required fields with value `unknown`. Verification compares the certified CARE-Bridge M_1 with a matched null mediator M_0 sharing the same schema and source roles. A single intervention strength $\epsilon = 0.9$ is fixed across models and datasets, while certification and verification use their natural zero-valued decision boundaries.

6.2 Main Results

Table 1 summarizes results across seven benchmarks, where CARE-Flow achieves the highest accuracy for every backbone–benchmark combination. On the six public benchmarks, it improves over vanilla inference by an average of 2.38, 2.56, and 2.47 percentage points for three backbones, respectively. The largest gains occur on CARE-Bench, outperforming the strongest baselines by 7.53, 6.45, and 4.28 percentage points, respectively. These larger gains on CARE-Bench are consistent with its emphasis on source and interaction errors, while improvements across all external benchmarks indicate that certified evidence routing also benefits general multi-image reasoning.

6.3 Analysis

Evidence-structure correction and source routing. Table 2 shows that CARE-Flow lowers all three reported inter-image hallucination error rates relative to vanilla inference for each backbone. The largest category-wise reductions occur in operator errors for LLaVA-NeXT and InternVL3, at 7.25 and 4.08 percentage points, respectively, and in operand omission for Qwen3.5, at 3.89 points. These backbone-dependent reductions are consistent with improvements in both evidence formation

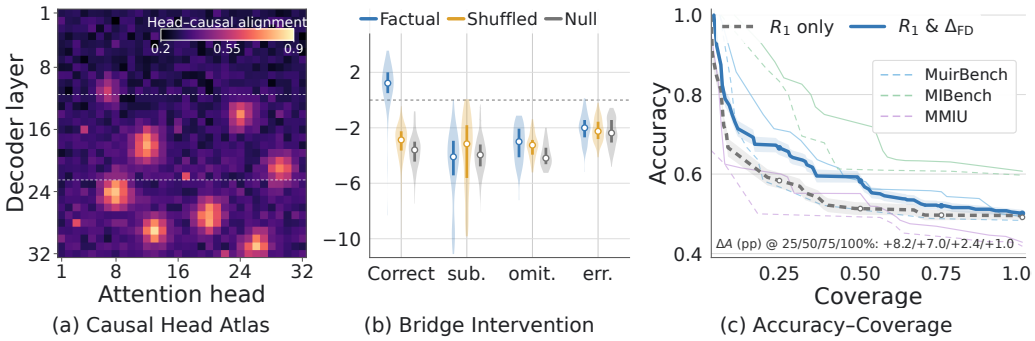


Figure 3: Intervention-based validation of internal evidence routing and reliability. (a) Layer–head agreement between source attention and intervention-derived causal support, revealing where causal source alignment emerges within the decoder. (b) Verifier margins for correct cases, shuffled/null controls, operand substitution (sub.), operand omission (omit.), and operator error (err.), showing their separation under intervention. (c) Comparison of R_1 and $R_1\&\Delta_{\mathrm{FD}}$ across datasets and coverage levels, reflecting the contribution of causal evidence to selective reliability.

and question-consistent composition. The intervention achieves 0.72 AUROC and 0.58 AUPRC for source identification, outperforming attention (0.61/0.42) and masking (0.52/0.38), thereby supporting path intervention for operand certification, as weakening an image path measures its causal contribution rather than only its attention allocation. Figure 3(a) further shows that attention–intervention alignment is sparse and concentrated mainly in middle and later decoder layers, suggesting that source-sensitive routing is localized to a subset of heads.

Evidence-sensitive verification and selective reliability. Figure 3(b) shows that positive verifier margins occur mainly for correct responses with factual evidence. Shuffled/null controls and factual operand-substitution, omission, and operator-error cases remain mostly negative, showing sensitivity to both required operands and question-consistent composition. Figure 3(c) further shows that adding Δ_{FD} to R_1 improves selective prediction across datasets, with the largest gains at low coverage (8.2 points at 25% and 7.0 at 50%). The matched null comparison thus more effectively filters responses that are supported mainly by linguistic compatibility. Appendix C.7 further analyzes the remaining accuracy gap, including imperfect extraction of visual-to-atomic evidence and broader reasoning errors beyond the evidence-flow failures targeted by CARE-Flow.

Ablation Study. Table 3 reports ablations macro-averaged over the three backbones. Removing causal certification causes the largest drop, from 40.47 to 31.75, showing that plausible evidence proposals are insufficient unless their source dependence is explicitly tested. Attention-based certification recovers performance to 37.00 but remains 3.47 points below CARE-Flow, indicating that attention is an imperfect proxy for evidence dependence. Removing the hard reset causes another substantial drop to 35.20, showing that certification alone is insufficient if downstream inference can still bypass the established evidence. Smaller losses without source contrast or Δ_{FD} suggest complementary benefits from rejecting competing sources and evidence-independent response support.

7 Conclusion

We propose CARE-Flow, a causal framework designed to reduce inter-image hallucinations by explicitly controlling evidence operands and their composition as specified by the query. CARE-Flow specifies the required evidence structure before response formation, causally certifies source-indexed operands, and executes the fixed operator only over certified evidence under a constructed conditional front-door computation. CARE-Bench complements the method with diagnostic operand-substitution, operand-omission, and operator-error cases across real and controlled visual domains. Experiments across three MLLMs and seven benchmarks show consistent accuracy gains, improved source identification, and stronger selective reliability. Future work may target the remaining benchmark errors beyond hallucination, including perception and reasoning.

AI USE STATEMENT

In this work, generative AI tools were used to support several stages of the research and writing workflow. During research ideation and execution, LLMs were used to explore and refine theoretical frameworks, provide feedback on methodological and experimental design, and assist with technical reasoning. They were also used to support the formulation and verification of mathematical claims and proofs, and to generate and filter synthetic data samples used in our benchmark construction. In addition, generative AI tools were used to edit and polish author-written text for clarity, readability, and technical presentation.

All AI-assisted outputs were reviewed and validated by the authors. Mathematical claims and proof steps were independently checked and verified for correctness. Synthetic data were subjected to automated quality-control procedures and manual inspection before inclusion in the benchmark. AI-assisted revisions to the manuscript were reviewed and edited by the authors to ensure that the final text accurately reflects the intended technical content. We take full responsibility for all content, claims, and artifacts presented in this work.

ETHICS STATEMENT

This work examines hallucination and evidence attribution in multi-image multimodal language models, aiming to improve the reliability and interpretability of their outputs. Our experiments are based on publicly available datasets, which are used both for evaluation and for constructing synthetic evaluation data. We exclude content involving culturally sensitive material and identifiable portraits of real people from the constructed benchmark. The study does not involve human subjects or the collection of private personal information. As with other methods for improving model reliability, the proposed approach does not guarantee factual correctness and should not be treated as a substitute for appropriate validation in safety-critical applications. We comply with the applicable licenses and terms of use for the datasets and models used in this work.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the CARE-Flow methodology and experimental protocol in the main paper, with additional implementation details, hyperparameters, and evaluation procedures provided in the appendix. The construction and composition of CARE-Bench, including its domains, question generation and filtering procedures, and evaluation criteria, are also documented in the appendix. Complete proofs and assumptions for the theoretical results are provided in the supplementary material. The code and data necessary to reproduce the experiments will be released via the link: https://anonymous.4open.science/r/care-flow-hallucination-DDD5.

REFERENCES

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

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12), 2023. doi: 10.1145/3571730.

Dongfu Jiang, Xuan He, Huaye Zeng, Cong Wei, Max Ku, Qian Liu, and Wenhu Chen. MANTIS: Interleaved multi-image instruction tuning. *Transactions on Machine Learning Research*, 2024.

Minyoung Lee, Yeji Park, Dongjun Hwang, Yejin Kim, Seong Joon Oh, and Junsuk Choe. Enhancing multi-image understanding through delimiter token scaling. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=7QFf05KrOm.

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.

Jiale Li, Mingrui Wu, Zixiang Jin, Hao Chen, Jiayi Ji, Xiaoshuai Sun, Liujuan Cao, and Rongrong Ji. Mihbench: Benchmarking and mitigating multi-image hallucinations in multimodal large language models. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, pp. 3143–3152, New York, NY, USA, 2025a. Association for Computing Machinery. ISBN 9798400720352. doi: 10.1145/3746027.3754993. URL https://doi.org/10.1145/3746027.3754993.

Xudong Li, Mengdan Zhang, Peixian Chen, Xiawu Zheng, Yan Zhang, Jingyuan Zheng, Yunhang Shen, Ke Li, Chaoyou Fu, Xing Sun, and Rongrong Ji. Zooming from context to cue: Hierarchical preference optimization for multi-image MLLMs. In *Advances in Neural Information Processing Systems*, volume 38, 2025b.

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. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.20.

Haowei Liu, Xi Zhang, Haiyang Xu, Yaya Shi, Chaoya Jiang, Ming Yan, Ji Zhang, Fei Huang, Chunfeng Yuan, Bing Li, and Weiming Hu. MIBench: Evaluating multimodal large language models over multiple images. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1906–1919, 2020. doi: 10.18653/v1/2020.acl-main.173.

Fanqing Meng, Jin Wang, Chuanhao Li, Quanfeng Lu, Hao Tian, Jiaqi Liao, Xizhou Zhu, Jifeng Dai, Yu Qiao, Ping Luo, Kaipeng Zhang, and Wenqi Shao. MMIU: Multimodal multi-image understanding for evaluating large vision-language models. In *Proceedings of the Thirteenth International Conference on Learning Representations*, 2025.

Joonki Min, Chaeyun Kim, Hyungwook Choi, Yejin Kim, Kihyun Kim, Yohan Jo, and Joonseok Lee. Fine-grained multi image object hallucination benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18295–18305, 2026.

Yeji Park, Minyoung Lee, Sanghyuk Chun, and Junsuk Choe. Mitigating cross-image information leakage in multi-image understanding with large vision-language models. *arXiv preprint arXiv:2508.13744*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, 2017.

Fei Wang, Xingyu Fu, James Y. Huang, Zekun Li, Qin Liu, Xiaogeng Liu, Mingyu Derek Ma, Nan Xu, Wenxuan Zhou, Kai Zhang, Tianyi Lorena Yan, Wenjie Jacky Mo, Hsiang-Hui Liu, Pan Lu, Chunyuan Li, Chaowei Xiao, Kai-Wei Chang, Dan Roth, Sheng Zhang, Hoifung Poon, and Muhao Chen. Muirbench: A comprehensive benchmark for robust multi-image understanding. In *Proceedings of 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*, 2024.

Xiaochen Yang, Hao Fang, Jiawei Kong, Yaoxin Mao, Bin Chen, and Shu-Tao Xia. Looking back and forth: Cross-image attention calibration and attentive preference learning for multi-image hallucination mitigation. In Paolo Favaro, Zuzana Kukelova, Atsuto Maki, Anna Rohrbach, Konrad Schindler, and Federico Tombari (eds.), *Computer Vision – ECCV 2026*, pp. 259–277, Cham, 2026. Springer Nature Switzerland. ISBN 978-3-032-37592-6.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, volume 32, pp. 12360–12371, 2019.

Bingchen Zhao, Yongshuo Zong, Letian Zhang, and Timothy Hospedales. Benchmarking multi-image understanding in vision and language models: Perception, knowledge, reasoning, and multi-hop reasoning. *arXiv preprint arXiv:2406.12742*, 2024.

A Proofs

A.1 Proof of Proposition 1

The partition

$$\mathcal{T}, \mathcal{V}_1, \ldots, \mathcal{V}_K$$

implies

$$\sum_s = \sum_{s\in\mathcal{T}} + \sum_{k=1}^{K} \sum_{s\in\mathcal{V}_k} .$$

Substituting this equality into the multi-head attention output gives

$$o_t^{(\ell)} = b_O^{(\ell)} + \sum_h W_O^{(\ell,h)} \sum_{s\in\mathcal{T}} \alpha_{ts}^{(\ell,h)} v_s^{(\ell,h)} + \sum_{k=1}^{K} \sum_h W_O^{(\ell,h)} \sum_{s\in\mathcal{V}_k} \alpha_{ts}^{(\ell,h)} v_s^{(\ell,h)}.$$

The definitions of

$$c_{t\leftarrow\text{other}}^{(\ell)}$$

and

$$c_{t\leftarrow I_k}^{(\ell)}$$

therefore yield

$$o_t^{(\ell)} = c_{t\leftarrow\text{other}}^{(\ell)} + \sum_{k=1}^{K} c_{t\leftarrow I_k}^{(\ell)}.$$

Differentiating

$$\widetilde{o}_t^{(\ell)}(\mathbf{g}) = c_{t\leftarrow\text{other}}^{(\ell)} + \sum_{k=1}^{K} g_k c_{t\leftarrow I_k}^{(\ell)}$$

with respect to g_k gives

$$\frac{\partial \widetilde{o}_t^{(\ell)}}{\partial g_k} = c_{t\leftarrow I_k}^{(\ell)}.$$

The gate is applied after the attention weights and source values at layer ℓ have been computed. No g_k therefore appears in

$$\alpha_{ts}^{(\ell,h)} = \text{softmax}_s \left(\frac{q_t^{(\ell,h)\top} k_s^{(\ell,h)}}{\sqrt{d_h}} \right)$$

at the intervened layer, so the layer's attention weights remain unchanged. Downstream attention distributions may change because the intervention alters subsequent hidden states; this downstream response is part of the intended causal effect. □

A.2 Proof of Theorem 1

After the hard reset, the structural equation for the response is

$$Y = f_Y(M, q, U_Y).$$

The equation contains M, q, and U_Y, but contains neither X nor any state generated from X outside M. Consequently, for fixed $M = m$ and $Q = q$, changing X cannot change the conditional distribution of Y. Hence,

$$Y \perp X \mid M, Q.$$

Under the intervention $do(X = x)$, the law of total probability gives

$$p_\theta(y \mid do(X = x), q) = \sum_m p_\theta(m \mid do(X = x), q) p_\theta(y \mid m, do(X = x), q).$$

The established conditional independence implies

$$p_\theta(y \mid m, do(X = x), q) = p_\theta(y \mid do(M = m), q).$$

The equality also uses modularity of the response-stage mechanism: after the reset, the stage receives m but no other descendant of the original visual computation, and its randomness U_Y is independent of the Bridge-stage randomness U_M. Substitution yields

$$p_\theta(y \mid do(X = x), q) = \sum_m p_\theta(m \mid do(X = x), q) p_\theta(y \mid do(M = m), q).$$

For autoregressive generation, the next-token mechanism at step t receives only

$$m, \quad q, \quad y_{<t}, \quad \text{and response-stage randomness.}$$

The original visual state x is absent once m is fixed. Therefore,

$$p_\theta\left(y_t \mid do(X = x), M = m, q, y_{<t}\right) = p_\theta\left(y_t \mid do(M = m), q, y_{<t}\right).$$

□

A.3 Expressive Sufficiency of Question-Specific Inference DAGs

Our formulation uses the question-conditioned specification $\mathcal{S}_Q = (R_Q, r_Q)$, where $R_Q = (\rho_1, \ldots, \rho_J)$ is the ordered tuple of required evidence roles. For a composite operator, let G_Q denote a directed acyclic graph representing r_Q. The atomic facts supplied to these roles form a finite set $\mathcal{A}_Q = \{a_1, \ldots, a_n\}$. The role order is retained by R_Q, while G_Q specifies how the supplied facts are bound and composed. This section establishes the expressive sufficiency of this graph representation for finite, non-recursive inference.

Definition (Finite compositional inference). Let $A_Q = \{a_1, \ldots, a_n\}$ be a finite set of atomic facts. A finite compositional inference over A_Q is a finite sequence

$$z_j = r_j\big(x_{j1}, \ldots, x_{jk_j}\big), \qquad j = 1, \ldots, m,$$

where r_j is a k_j-ary operator and every argument satisfies

$$x_{j\ell} \in A_Q \cup \{z_1, \ldots, z_{j-1}\}.$$

Thus, each inference step may depend directly on atomic facts or on intermediate results produced by earlier steps. Without loss of generality, the final variable z_m is taken to be the answer-relevant output; multiple outputs can be handled by adding a final aggregation operator.

Importantly, the arguments of r_j are ordered. More generally, each argument position may be associated with a semantic role $\rho_{j\ell}$, such as *agent*, *patient*, *reference*, or *target*. Hence, asymmetric relations are represented without assuming commutativity.

Definition (Question-specific inference DAG). A question-specific inference DAG is a tuple

$$G_Q = (V_Q, E_Q, \lambda_Q, \rho_Q),$$

where the leaf nodes correspond to the atomic facts in A_Q, each internal node $v \in V_Q$ is labelled by an operator $\lambda_Q(v)$, and each incoming edge is labelled by its argument position or semantic role through ρ_Q. An edge

$$u \xrightarrow{\rho} v$$

therefore states not only that the value of u is an input to v, but also the role that this input plays in the operator at v.

Proposition 2 (Representation of finite compositional inference). Every finite compositional inference over A_Q admits an exactly equivalent question-specific inference DAG. Conversely, every finite operator-labelled DAG defines a finite compositional inference.

Proof. We first prove the forward direction. For every atomic fact $a_i \in A_Q$, create a leaf node v_{a_i}. For every inference step

$$z_j = r_j(x_{j1}, \ldots, x_{jk_j}),$$

create an internal node v_j labelled by r_j. For each argument $x_{j\ell}$, add an edge from the node representing $x_{j\ell}$ to v_j:

$$v(x_{j\ell}) \xrightarrow{\rho_{j\ell}} v_j,$$

where $\rho_{j\ell}$ records its argument position or semantic role.

We first show that the resulting graph is acyclic. By definition, every intermediate argument used in the computation of z_j must belong to

$$\{z_1, \ldots, z_{j-1}\}.$$

Therefore, every edge between two operator nodes has the form

$$v_i \to v_j \qquad \text{with} \qquad i < j.$$

Suppose, for contradiction, that the constructed graph contains a directed cycle. Then the cycle must contain operator nodes $v_{j_1}, \ldots, v_{j_s}$ satisfying

$$j_1 < j_2 < \cdots < j_s < j_1,$$

because every directed edge strictly increases the operator index. This is impossible. Hence, the graph is acyclic.

It remains to show that the graph computes exactly the same result as the original inference. We proceed by induction over j. For $j = 1$, all inputs to r_1 are atomic facts, so evaluating v_1 yields exactly

$$z_1 = r_1(x_{11}, \ldots, x_{1k_1}).$$

Assume that all nodes $v_1, \ldots, v_{j-1}$ reproduce the corresponding intermediate values $z_1, \ldots, z_{j-1}$. Every input to v_j is then either an atomic fact or one of these correctly reproduced intermediate values. Since v_j applies the same operator r_j to the same ordered arguments,

$$v_j = r_j(x_{j1}, \ldots, x_{jk_j}) = z_j.$$

By induction, all intermediate values and, in particular, the final output z_m are preserved exactly. Thus, every finite compositional inference admits an equivalent DAG representation.

For the converse direction, let G_Q be any finite directed acyclic operator graph. Every finite DAG admits a topological ordering

$$v_1, \ldots, v_m$$

such that every predecessor of v_j occurs before v_j. We may therefore evaluate the internal nodes in this order. For each v_j, let its ordered predecessors be $x_{j1}, \ldots, x_{jk_j}$ and define

$$z_j = \lambda_Q(v_j)\big(x_{j1}, \ldots, x_{jk_j}\big).$$

Because every predecessor is either an atomic-fact leaf or the output of an earlier node in the topological ordering, this sequence satisfies the definition of finite compositional inference. Therefore,

$$\text{finite non-recursive compositional inference} \iff \text{finite operator-labelled DAG}.$$

□

Role-sensitive and asymmetric operators. The graph representation does not require operators to be commutative. For example,

$$\mathrm{chase}(a_{\mathrm{cat}}, a_{\mathrm{dog}})$$

and

$$\mathrm{chase}(a_{\mathrm{dog}}, a_{\mathrm{cat}})$$

use the same two atomic facts and the same operator, but correspond to different graphs because their argument bindings differ. Specifically, the first structure contains

$$a_{\mathrm{cat}} \xrightarrow{\text{agent}} \mathrm{chase}, \qquad a_{\mathrm{dog}} \xrightarrow{\text{patient}} \mathrm{chase},$$

whereas the second reverses these bindings. Hence, argument order and semantic roles are part of G_Q, rather than properties that must be encoded by introducing additional hallucination categories.

Likewise, multiple operators are naturally represented. For example, an inference of the form

$$r_3\Big(r_1(a_1, a_2), r_2(a_3, a_4)\Big)$$

is represented by two intermediate operator nodes feeding a third node. Thus, the formulation does not assume that a question requires only a single relation.

Proposition 3 (Finite-domain expressive completeness). Consider a fixed question Q. Suppose that each atomic fact a_i takes values in a finite domain $\mathcal{D}_i$, and that the answer space $\mathcal{Y}$ is finite. Then every deterministic question-answer mapping

$$f_Q : \mathcal{D}_1 \times \cdots \times \mathcal{D}_n \to \mathcal{Y}$$

admits a finite DAG representation after a finite Boolean encoding of its inputs and outputs.

Proof. Because every $\mathcal{D}_i$ is finite, each atomic fact can be represented by a finite binary encoding. Concatenating these encodings gives a finite binary vector

$$\mathbf{b} = (b_1, \ldots, b_N) \in \{0,1\}^N.$$

Similarly, because $\mathcal{Y}$ is finite, each possible answer can be assigned a finite binary code. It therefore suffices to show that every Boolean output bit can be represented by a finite DAG.

Consider an arbitrary Boolean function

$$g : \{0,1\}^N \to \{0,1\}.$$

For each binary assignment

$$\mathbf{u} = (u_1, \ldots, u_N)$$

such that $g(\mathbf{u}) = 1$, define the minterm

$$M_{\mathbf{u}}(\mathbf{b}) = \bigwedge_{\ell=1}^{N} L_\ell^{(\mathbf{u})}(b_\ell),$$

where

$$L_\ell^{(\mathbf{u})}(b_\ell) = \begin{cases} b_\ell, & u_\ell = 1, \\ \neg b_\ell, & u_\ell = 0. \end{cases}$$

By construction,

$$M_{\mathbf{u}}(\mathbf{b}) = 1 \quad \Longleftrightarrow \quad \mathbf{b} = \mathbf{u}.$$

Now define

$$G(\mathbf{b}) = \bigvee_{\mathbf{u}:\, g(\mathbf{u})=1} M_{\mathbf{u}}(\mathbf{b}).$$

If $g(\mathbf{b}) = 1$, then the term $M_{\mathbf{b}}$ appears in the disjunction and evaluates to 1, so $G(\mathbf{b}) = 1$. If $g(\mathbf{b}) = 0$, no satisfying assignment in the disjunction equals $\mathbf{b}$, and therefore every minterm evaluates to 0, giving $G(\mathbf{b}) = 0$. Hence,

$$G(\mathbf{b}) = g(\mathbf{b}) \qquad \forall\, \mathbf{b} \in \{0,1\}^N.$$

The expression G is a finite composition of$\{\neg, \wedge, \vee\}$. It therefore has a finite operator-DAG representation by Proposition 2. Applying this construction independently to the finitely many output bits encoding f_Q and sharing their atomic-fact inputs yield a finite multi-output DAG representing the entire mapping. A final deterministic decoding node maps these output bits to an element of $\mathcal{Y}$. Consequently, every deterministic mapping from finite-domain atomic facts to a finite answer space admits a finite DAG representation.

□

Implication for the evidence structure. Propositions 2–3 show that, for finite compositional inference, the question-specified operator r_Q in $\mathcal{S}_Q = (R_Q, r_Q)$ admits a DAG representation G_Q. Atomic facts from $\mathcal{A}_Q$ fill the ordered roles in R_Q, while G_Q specifies operator labels, argument bindings, dependencies, and intermediate operations. An operand error changes the fact supplied to a required role; an operator-structure error changes the graph representing r_Q. Thus, simple operators and composite DAGs use the same evidence specification.

Scope of the result. The above results concern *expressive existence*, rather than efficiency. The Boolean construction in Proposition 3 can require exponentially many nodes in the worst case, so the proposition does not imply that every reasoning problem has a compact DAG representation or that such a graph can be efficiently learned. The formulation also assumes finite, question-specific, non-recursive inference. Problems requiring an unbounded number of recursive or fixed-point inference steps are outside this scope. These restrictions are consistent with the setting considered here, where each question operates over a finite set of visual inputs and a finite set of question-relevant atomic facts.

A.4 Algorithms

Algorithm 1 summarizes the end-to-end CARE-Flow procedure. It first fixes the evidence roles and composition operator required by the question, then proposes source-indexed operands and tests their visual dependence through image-path interventions. Uncertified operands remain in their required roles as `unknown`. After a hard reset removes the multimodal states, the fixed operator is applied to the certified CARE-Bridge in a fresh text-only computation. Complete responses are then verified against both this Bridge and a structure-matched null Bridge before selection or abstention.

B Efficiency Analysis

B.1 Efficiency Tests

Evaluation protocol. We measure the practical inference cost of CARE-Flow on the frozen 60-example CARE-Bench test split to evaluate performance. Vanilla and CARE-Flow use the same frozen Qwen3.5-9B backbone, default precision, batch size 1, a maximum image side length of 768 pixels, and deterministic decoding. CARE-Flow uses the confirmed configuration, including source-isolated Bridge proposal, exact source-grouped certification, hard reset, and factual/null Bridge verification. The proposal and answer budgets are 320 and 96 new tokens, respectively. All measurements are performed on a single NVIDIA RTX PRO 6000 Blackwell Max-Q GPU (96 GB) with PyTorch 2.12.0, Transformers 5.8.1, and CUDA 13.0. We report measured wall-clock time.

The model is loaded once, and two complete examples are used for warm-up and excluded from the statistics. For each method and example, we synchronize the GPU immediately before and after inference and reset the peak-memory counter. Latency includes image loading, image preprocessing, and every model call in the corresponding inference procedure. Model loading, warm-up, and cache clearing between measurements are excluded. We alternate the execution order of Vanilla and CARE-Flow across examples to reduce order effects. A process-level guard postpones measurement whenever another compute process is present on the target GPU. Peak memory is the maximum allocated GPU memory reported by `torch.cuda.max_memory_allocated`.

Efficiency–accuracy trade-off. CARE-Flow improves accuracy from 48.33% to 58.33% on the same test split while increasing mean latency from 0.870 to 1.189 seconds per example. This is an absolute overhead of 0.319 seconds and a $1.37\times$ relative cost. Sequential throughput decreases from

Algorithm 1 CARE-Flow inference

Require: Images $V = \{I_1, \ldots, I_K\}$, question Q, optional set of complete responses $\mathcal{H}$, frozen p_θ, intervention strength ϵ

$\mathcal{S}_Q = (R_Q, r_Q) \leftarrow \mathrm{Specify}_\theta(Q)$, where $R_Q = (\rho_1, \ldots, \rho_J)$
$\widetilde{\mathcal{E}} = \{\widetilde{e}_j = (\rho_j, S_j, c_j)\}_{j=1}^{J} \leftarrow \mathrm{Propose}_\theta(V, Q, \mathcal{S}_Q)$
for $j = 1, \ldots, J$ **do**
    Compute $F_j(\mathbf{1})$ for c_j in an independent teacher-forced branch
    for $k = 1, \ldots, K$ **do**
        Compute $F_j(\mathbf{1} - \epsilon \mathbf{e}_k)$ by gating Image k's outgoing value-output paths
        $N_{jk}^{(\epsilon)} \leftarrow \big[F_j(\mathbf{1}) - F_j(\mathbf{1} - \epsilon \mathbf{e}_k)\big]/\epsilon$
    end for
    $D_j \leftarrow \min_{k \in S_j} N_{jk}^{(\epsilon)} - \max\big(\{0\} \cup \{N_{jr}^{(\epsilon)} : r \notin S_j\}\big)$
    if $D_j > 0$ **then**
        $\bar{c}_j \leftarrow c_j$
    else
        $\bar{c}_j \leftarrow$ `unknown`
    end if
end for
$\mathcal{E}^\star \leftarrow \{(\rho_j, S_j, \bar{c}_j)\}_{j=1}^{J}$; $M_1 \leftarrow \mathcal{E}^\star$
Discard images, visual tokens, multimodal and intervention states, and all key/value caches
Start a fresh text-only computation
$z_1 \leftarrow \mathrm{Compose}_\theta(r_Q, M_1)$
if $\mathcal{H} = \varnothing$ **then**
    $\mathcal{H} \leftarrow \mathrm{Generate}_\theta(Q, M_1, z_1)$
end if
Construct M_0 with the roles, sources, ordering, and structure of M_1, replacing factual values with `unavailable`
$z_0 \leftarrow \mathrm{Compose}_\theta(r_Q, M_0)$
for each $a \in \mathcal{H}$ **do**
    Compute the Yes–No log-likelihood margins $R_1(a)$ and $R_0(a)$ using (M_1, z_1) and (M_0, z_0), respectively
    $\Delta_{\mathrm{FD}}(a) \leftarrow R_1(a) - R_0(a)$
end for
$\mathcal{H}_{\mathrm{valid}} \leftarrow \{a \in \mathcal{H} : R_1(a) > 0 \ \wedge \ \Delta_{\mathrm{FD}}(a) > 0\}$
if $\mathcal{H}_{\mathrm{valid}} = \varnothing$ **then**
    return `abstain` ▷ Alternatively, regenerate from Q, M_1, z_1
end if
return $\arg \max_{a \in \mathcal{H}_{\mathrm{valid}}} R_1(a)$

Table 4: End-to-end inference efficiency on a randomly selected CARE-Bench test split ($N = 60$). Latency is reported in seconds per example as mean $\pm$ standard deviation, together with its median and 95th percentile (P95). Relative time is normalized to Vanilla. Peak memory includes the resident model weights. Accuracy is included to show the performance–efficiency trade-off under the same frozen evaluation protocol.

Method	Acc. ↑	Mean latency ↓	Median ↓	P95 ↓	Rel. time ↓	Ex./min ↑	Peak mem. ↓
Vanilla	48.33	0.870 ± 0.186	0.951	1.059	1.00×	68.98	17.994 GiB
CARE-Flow	58.33	1.189 ± 0.190	1.212	1.293	1.37×	50.46	18.030 GiB

68.98 to 50.46 examples per minute. The P95 latency is 1.293 seconds, reflecting the additional cost incurred by examples that require more evidence atoms or a free-form Bridge proposal. In contrast, peak allocated memory changes only from 17.994 to 18.030 GiB, an increase of 0.036 GiB (approximately 37 MiB or 0.20%). For completeness, the maximum memory reserved by the PyTorch caching allocator is 20.494 GiB for Vanilla and 21.172 GiB for CARE-Flow. CARE-Flow,

therefore, trades additional sequential model passes for accuracy, but does not require a second backbone or a material increase in peak allocated GPU memory.

Source of the runtime overhead. Instrumentation of the model backend shows that exact source-grouped causal certification accounts for 65.4% of CARE-Flow's backend time. Multimodal candidate scoring, which includes the initial hypothesis and schema-constrained source proposals, accounts for 23.6%. Factual/null Bridge verification contributes 7.8%, while free-form multimodal proposal contributes 3.2%. For example, CARE-Flow scores 8.23 multimodal candidate completions and 15.07 text-only verifier completions on average, with 0.27 autoregressive multimodal proposal calls. Vanilla scores 3.77 multimodal candidate completions. The measured slowdown is thus primarily due to the cost of causal certification rather than to parameter or memory growth. Since certification calls for different evidence atoms and source interventions are independent, they also provide a direct target for future batching or parallel execution without changing the CARE-Flow decision rule.

Dependence on the number of images. On the 14 three-image examples, the mean CARE-Flow latency is 1.120 seconds, whereas on the 46 four-image examples, it is 1.210 seconds. The four-image subset requires 4.78 source interventions per example on average, compared with 3.71 for the three-image subset, which helps explain the higher latency. These subset figures are descriptive only because image count and task composition co-vary in CARE-Bench.

B.2 Computational Complexity

Let K denote the number of images, J the number of proposed evidence operands, and $B = |\mathcal{H}|$ the number of complete responses evaluated during verification. Let C_{prop} denote the decoder cost of proposing CARE-Bridge, $C_{\mathrm{mm}}(j)$ the cost of one multimodal teacher-forced pass for operand j, and $C_{\mathrm{text}}(a)$ the cost of one text-only verification pass for response a. Visual embeddings are computed once by the frozen vision encoder and reused across all intervention branches.

Time complexity. Operand certification requires one baseline evaluation $F_j(\mathbf{1})$ and one intervention for each of the K image paths. Hence, operand j requires $K+1$ decoder passes, giving

$$C_{\mathrm{cert}} = \sum_{j=1}^{J} (K+1) C_{\mathrm{mm}}(j).$$

Multi-source operands introduce no combinatorial enumeration over image subsets because the same K image-specific interventions provide all $N_{jk}^{(\epsilon)}$. Source-grouped attention decomposition also preserves the asymptotic cost of a standard attention pass: every attention value contribution is computed once and accumulated according to its image source.

For verification, CARE-Flow evaluates each complete response once with the certified CARE-Bridge M_1 and once with the matched null mediator M_0. The total inference cost is therefore

$$C_{\mathrm{CARE}} = C_{\mathrm{enc}} + C_{\mathrm{prop}} + \sum_{j=1}^{J} (K+1) C_{\mathrm{mm}}(j) + C_{\mathrm{gen}} + 2 \sum_{a \in \mathcal{H}} C_{\mathrm{text}}(a),$$

where C_{enc} is the one-time vision-encoding cost and C_{gen} is present only when CARE-Flow generates response hypotheses. Consequently, the additional multimodal computation scales linearly as $O(JK)$, while response verification scales as $O(B)$. CARE-Flow does not enumerate possible evidence operators; question-specified composition is performed by the frozen language model over the certified operands.

For a Transformer with L decoder layers, hidden width d, and multimodal sequence length bounded by S, a standard full-sequence decoder pass has complexity

$$O\big(L(S^2 d + S d^2)\big).$$

Operand certification, therefore, has the upper bound

$$O\big(J(K+1)L(S^2 d + S d^2)\big),$$

excluding the shared vision encoding. Batching operands or intervention branches can reduce wall-clock latency, but does not change the total asymptotic computation.

Space complexity. CARE-Flow uses forward interventions only and does not retain a backward computation graph. Intervention branches can therefore be evaluated sequentially, so their activation memory is reused instead of growing with JK. Beyond the memory required by one ordinary MLLM inference pass, CARE-Flow stores the visual embeddings, the certified textual mediator, and the intervention scores $N \in \mathbb{R}^{J\times K}$. The additional storage is

$$O(S_V d + JK + |M_1|),$$

where S_V is the number of visual tokens and $|M_1|$ is the length of the certified CARE-Bridge. Processing one operand at a time reduces the storage of the intervention scores from $O(JK)$ to $O(K)$. The peak model-state memory thus remains comparable to a single inference branch, with only linear auxiliary storage in the number of images and evidence operands.

Summary. CARE-Flow trades additional forward computation for explicit causal certification. The dominant overhead is $J(K+1)$ multimodal scoring passes, linear in both the number of images and proposed operands. Neither multi-source evidence nor composite operators introduces exponential search, and sequential intervention evaluation prevents the additional space cost from scaling with the number of intervention branches.

C CARE-BENCH

C.1 CARE-BENCH EXAMPLES

COCO-VQA · 000023



Question Across Image 1 and Image 4, where is a laptop on a table visible?

A. Image 4 only — Operand omission
B. Image 1 only — Operand omission
C. Both images — Correct
D. Neither image — Operator error

COCO-VQA · 000029



Question Looking at Image 1 and Image 3, where is a shower curtain visible?

A. Both images — Operator error
B. Image 3 only — Correct
C. Neither image — Operator error
D. Image 1 only — Operator error

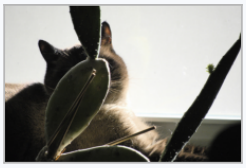
COCO-VQA · 000078

I1  I2  I3 

Question Focus on Image 1. What object does it show?

A. surfboard — Correct
B. snowboard — Operand substitution
C. kite — Operand substitution

COCO-VQA · 000094

I1  I2  I3 

Question Can you identify the object shown in Image 3?

A. cat — Operand substitution
B. bear — Operand substitution
C. dog — Correct

COCO-VQA · 000096

I1  I2  I3 

Question What is the visible vehicle in Image 2?

A. truck — Operand substitution
B. motorcycle — Correct
C. bicycle — Operand substitution

Te Papa · 000348

I1  I2  I3  I4 

Question Where can you see a geyser actively erupting steam?

A. Image 2 — Operator error
B. Image 4 — Correct
C. Image 3 — Operator error
D. Image 1 — Operator error

Te Papa · 000219

I1  I2  I3  I4 

Question Look for a bird mounted on a stand. Which image is it in?

A. Image 2 — Operator error
B. Image 3 — Operator error
C. Image 1 — Operator error
D. Image 4 — Correct

Te Papa · 000367

I1  I2  I3 

Question Which feature is shown in Image 2?

A. church — Operand substitution
B. horse-drawn carriage — Operand substitution
C. bridge — Correct

Te Papa · 000275

I1  I2  I3  I4 

Question Where does a sailboat on the water appear: Image 1, Image 3, both, or neither?

A. Image 3 only — Correct
B. Image 1 only — Operator error
C. Both images — Operator error
D. Neither image — Operator error

Te Papa · 000350

I1  I2  I3  I4 

Question Where can you see a vase holding flowers when comparing Image 3 and Image 4?

A. Image 4 only — Operator error
B. Neither image — Operator error
C. Both images — Operator error
D. Image 3 only — Correct

Generated · S0020

I1  I2  I3  I4 

Question Look at the notebook next to the laptop in Image 1, the umbrella next to the bench in Image 2, and the suitcase next to the bench in Image 3. What colors are they, in that order?

A. red; green; orange — Operator error
B. purple; red; green — Operand substitution
C. orange; red — Operand omission
D. orange; red; green — Correct

Gold reasoning

A_Q^*
```
I1: orange(notebook);
    beside(notebook, laptop)
I2: red(umbrella);
    beside(umbrella, bench)
I3: green(suitcase);
    beside(suitcase, bench)
```

r_Q^*
```
return (color(I1 notebook),
    color(I2 umbrella), color(I3
    suitcase)) in source order
```

Generated · S0136

I1  I2  I3 

Question Where is the bicycle in relation to the lamp post in Image 2, and where is the book in relation to the wall lamp in Image 3?

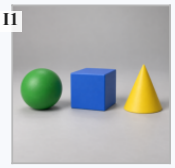
A. beside; beside — Operand substitution
B. below; beside — Operator error
C. beside — Operand omission
D. beside; below — Correct

Gold reasoning

A_Q^*
```
I2: beside(bicycle, lamp_post)
I3: below(book, wall_lamp)
```

r_Q^*
```
r1=relation
(bicycle,lamp_post,I2);
    r2=relation(book,wall_lamp,I3);
    return (r1,r2)
```

Generated · S0161

I1  I2  I3  I4  I5 

Question What is the woman doing with the umbrella in Image 2, and what is the man doing with the suitcase in Image 3?

A. holds umbrella; carries suitcase — Correct
B. holds suitcase; carries umbrella — Operand substitution
C. carries suitcase; holds umbrella — Operator error
D. holds umbrella — Operand omission

Gold reasoning

A_Q^*
```
I2: holds(woman, umbrella)
I3: carries(man, suitcase)
```

r_Q^*
```
retrieve
    r1=holds(woman,umbrella,I2),
    r2=carries(man,suitcase,I3);
    return (r1,r2)
```

Generated · S0215



Question Which image has more sailboats, Image 1 or Image 2, and how many sailboats does it have?

A. Image 1; 2 — Operator error
B. Image 2; 2 — Operand substitution
C. Image 2; 4 — Correct
D. Image 2 only — Operand omission

Gold reasoning

A_Q^*
```
I1: count(sailboat)=2
I2: count(sailboat)=4
```

r_Q^*
```
c1=count
(sailboat,I1)=2;
    c2=count(sailboat,I2)=4;
    i*=argmax_{i in {1,2}} c_i;
    return (Image_i*,c_i*)
```

Generated · S0454



Question How many yellow flowers are inside the vases in Image 3 and Image 4 altogether?

A. 5 — Operator error
B. 3 — Correct
C. 2 — Operand omission
D. 4 — Operand substitution

Gold reasoning

A_Q^*
```
I3: count(yellow_flower inside
    vase)=1
I4: count(yellow_flower inside
    vase)=2
```

r_Q^*
```
c3=count
(yellow_flower inside vase,I3)=1;
    c4=count(yellow_flower inside
    vase,I4)=2; return c3+c4=3
```

C.2 CARE-BENCH GENERATION PIPELINE

Common representation. CARE-Bench is designed to evaluate whether an MLLM follows the evidence structure required by a multi-image question, in addition to producing the correct answer. Each benchmark instance contains an ordered image set $\mathcal{I} = (I_1, \ldots, I_K)$, a question Q, answer candidates $\mathcal{C}$, and a unique correct answer Y. We additionally record the verified atomic facts A_i^* for each image, the question-required operands $A_Q^* = (a_1^*, \ldots, a_m^*)$, and the question-specified operator r_Q^* such that

$$Y = r_Q^*(A_Q^*), \qquad \{a_j^* : 1 \le j \le m\} = \bigcup_{i=1}^{K} A_{Q,i}^*, \qquad A_{Q,i}^* \subseteq A_i^*. \tag{1}$$

The set equality collects the values in the ordered tuple A_Q^*; each tuple position retains its evidence role and source index. The source index remains attached to every operand, allowing the benchmark to distinguish an incorrect visual fact from a correct fact assigned to the wrong image or semantic role. Each instance contains three to five related images and is retained only when the displayed evidence supports a unique answer. Captions, metadata, and image-generation specifications may assist retrieval, organization, or candidate construction, but they are never treated as ground-truth visual evidence.

Domain-specific source and evidence acquisition. CARE-Bench uses three complementary domains to balance visual diversity with control over the underlying evidence.

COCO-VQA. COCO-VQA provides common objects, activities, attributes, and everyday scenes. We begin with licensed training and validation images, along with their instance annotations, captions, and VQA responses. Source filtering removes low-resolution images,

images containing person instances or person-related caption cues, OCR-dependent questions, subjective attributes, specialist labels, and VQA anchors with annotator agreement below 0.8. The retained annotations identify candidate visual cues rather than fixing the benchmark answer.

Within each coarse subject group, pixel-derived similarity retrieves related images to form three- to five-image sets. Captions and instance labels assist only with retrieval and candidate organization. Because the original annotations do not provide a complete inventory of visually available facts, Gemma 3 27B, an open-source multimodal model excluded from benchmark evaluation, extracts candidate atomic facts from the image pixels. These facts are subsequently subjected to independent visual validation before being added to A_i^*.

Te Papa. The museum branch extends the benchmark to less familiar objects, species, artifacts, artworks, specimens, and photography. We begin by collecting records that satisfy the applicable rights and sensitivity policies. Records with missing media, insufficient resolution, text-only content, restricted use, or disallowed photographic portraits are removed before question construction.

Pixel-derived similarity retrieves visually related works, while collection metadata is used only to organize the source pool. It is not accepted as visual evidence. As in the COCO-VQA branch, Gemma 3 27B proposes visually verifiable atomic facts directly from the images, and an independent visual validation stage determines which of these may be included in A_i^*. Because museum images may contain uncommon or historically specific content, candidate questions are restricted to visible cues that can be identified without reading labels or relying on curatorial or specialist knowledge.

Controlled Generated. The controlled branch provides direct control over the facts used to construct each image. We first define a symbolic scene specification A_{prompt}, the required operand roles, the operator, the intended answer, and the targeted distractor errors. Construction is stratified across attribute binding, counting and aggregation, spatial relations, actions and semantic roles, comparison, set reasoning, logical quantification, multi-hop composition, source tracking, and mixed reasoning.

Each image is generated independently rather than as part of a collage, with both contextual everyday scenes and tightly controlled abstract scenes represented. The rendered image is then treated as a new visual source rather than assuming that the generation prompt was faithfully realized. Reviewers reconstruct the verified facts A_i^* from the rendered pixels. Images are rejected or regenerated when an intended object, count, attribute, action, or relation is absent or ambiguous.

This produces the construction chain

$$A_{\text{prompt}} \to I \to A^* \to (A_Q^*, r_Q^*) \to Y, \tag{2}$$

which makes it possible to evaluate atomic-fact extraction separately from errors arising when evidence is selected or composed during multi-image reasoning.

Evidence-structured item construction. After domain-specific evidence acquisition, all three branches follow the same evidence-structured construction process.

1. **Form the image set.** We assemble three to five related images and assign their displayed source indices. Each set combines question-relevant sources with visually plausible distractor sources. Depending on the question, the required operator may select evidence from a single source, compare two sources, aggregate several sources, or compose multiple relations across images.
2. **Specify operands and operator.** For each relevant image, we select the question-required facts $A_{Q,i}^*$ from its verified atomic evidence. The operator r_Q^* is stored separately from the operands. Computed quantities or relations, including totals, extrema, set intersections, quantified decisions, comparisons, and relation compositions, are therefore represented as operations over visual facts rather than as facts directly visible in an image. This separation ensures that the annotated evidence is sufficient to derive the answer while preserving the distinction between evidence acquisition and evidence composition.
3. **Write the question and reference answer.** The question is written in natural English and specifies the required scope of sources without revealing the answer. Candidate answers

use a common semantic form and comparable level of specificity. An item is retained only when its reference answer follows from (A_Q^*, r_Q^*), is supported by the displayed pixels, and is unique among the candidates.

4. **Construct diagnostic incorrect choices.** Incorrect candidates are generated through controlled changes to the gold evidence structure. *Operand Substitution* replaces a required role with another grounded fact, for example, from the wrong image, object, or semantic role. *Operand Omission* leaves one or more required roles unfilled, producing an incomplete evidence structure. *Operator Error* preserves the relevant operands but applies a composition that is inconsistent with the question, such as an incorrect comparison, ordering, arithmetic operation, quantifier, or relation.

   These three error types directly instantiate the operand–operator formulation: a required operand can be replaced or omitted, while a remaining mismatch after operand alignment is attributed to the operator. The controlled branch additionally permits unsupported atomic substitutions to evaluate errors in fact extraction itself; these are recorded separately from the three inter-image evidence-structure errors.

Visual evidence validation. For the two real-image domains, candidate atomic evidence proposed by Gemma 3 27B is validated by an independent closed-source multimodal LLM-as-a-Judge. The judge examines the relevant image–fact pairs independently and determines whether each proposed fact is visibly supported. Positive judgments additionally record the visible location of the supporting cue, and a fact is accepted only when its confidence is at least 0.90.

The resulting support matrix is checked against the declared source indices and required operands. This prevents a fact that is visually valid in one image from being incorrectly attributed to another source. Captions, museum metadata, and original VQA annotations are not supplied as substitutes for pixel-level confirmation at this stage.

A separate adversarial validation pass searches for alternate valid answers, missing evidence, unsupported relations, ambiguous attributes, counting errors, tiny or unreliable visual details, OCR leakage, and specialist assumptions. For the controlled domain, reviewers also compare each rendered image to its intended scene and reconstruct A_i^* directly from the visible content. Generation failures are rejected or regenerated rather than accommodated by weakening the evidence annotation.

Judge reliability and controlled corruption. We additionally test the reliability of the validation procedure through controlled error injection. Known corruptions are inserted into validation inputs by altering operands, operators, source assignments, or answer validity. The judging stage is then required to reject the corrupted construction or identify the corresponding inconsistency. These tests provide direct checks that the validator is sensitive not only to unsupported visual facts but also to the evidence-structure errors that CARE-Bench is designed to diagnose.

Final quality control, balancing, and release. Before inclusion in CARE-Bench, each item undergoes deterministic checks for answer uniqueness, evidence sufficiency, source attribution, operand–operator consistency, distractor validity, and potential shortcuts. The declared evidence must be sufficient to recover the reference answer, each incorrect choice must instantiate its assigned error type, and no alternative candidate may remain compatible with the verified evidence structure.

Accepted items are deduplicated at the question, object, and image-group levels. We cap repeated image and target use and balance task families, source scopes, difficulty, image count, and correct-option position. Images and answer candidates are then independently randomized, with all source-indexed operands remapped after image reordering.

Dataset splits are defined by source identity, so the same underlying image cannot appear in multiple splits. The released record stores source and rights information, verified atomic evidence, required operands and operators, the diagnostic label for each incorrect choice, and the corresponding validation decisions. Only instances with a unique evidence-supported answer, sufficient visual evidence, valid source attribution, consistent operand–operator annotations, and no detected shortcut enter the final CARE-Bench release.

C.3 Prompts Used for CARE-Bench Construction

This section reports the instruction templates used by the multimodal models during benchmark construction. It complements the generation pipeline in Appendix C.2: the pipeline specifies when each model is called and how its output is filtered, whereas the templates below specify what the model receives. Angle-bracketed fields are filled programmatically for each instance. Images are passed through the model's native multimodal interface in their displayed order.

Atomic-fact extraction from real images. We use Gemma 3 27B to produce candidate atomic facts for one image at a time. COCO-VQA and Te Papa share the following template, with a domain-specific policy appended at <DOMAIN_POLICY>.

```
You are extracting atomic visual evidence for a multi-image QA
benchmark. Exactly one image is attached and is identified as Image
<IMAGE_INDEX>. Describe only facts directly supported by its pixels.

A fact must contain one visually testable claim. Use ordinary object
names and coarse attributes. Record the source image, a short factual
statement, and the location of its visible support. Do not infer
intent, identity, history, location, or any fact requiring external
knowledge. Do not use captions, filenames, metadata, or dataset
annotations. Reject tiny, occluded, subjective, or ambiguous details.
State an exact count only when every counted object is visible and
additional instances can be ruled out.

<DOMAIN_POLICY>

Return JSON only:
{
  "image_index": <IMAGE_INDEX>,
  "reject": false,
  "facts": [
    {
      "fact_id": "I<IMAGE_INDEX>_F1",
      "predicate": "object_presence|attribute|action|spatial_
      relation|count",
      "arguments": ["..."],
      "statement": "one atomic visible claim",
      "evidence_location": "specific image region",
      "confidence": 0.0
    }
  ],
  "rejection_reason": ""
}
```

For COCO-VQA, <DOMAIN_POLICY> reads: "Use common objects, actions, coarse attributes, and ordinary scene relations. Ignore COCO captions, category labels, and VQA answers." For Te Papa, it reads: "Use broad visual categories and ordinary visible properties. Do not use collection titles, catalog descriptions, artist names, dates, exact places, species-level names, cultural attributions, or curatorial knowledge." Thus, the same fact schema is retained across the real-image domains, while the museum branch applies a stricter boundary against metadata and specialist identification.

Evidence-structured item authoring. The authoring model receives the ordered images, their verified fact inventories, a requested question family, and a target error profile. It is instructed to construct the question from the evidence representation rather than from captions or metadata.

```
You are constructing one CARE-Bench multiple-choice item from
<NUM_IMAGES> ordered images and their verified atomic facts.

Verified facts:
<VERIFIED_FACTS_BY_IMAGE>

Requested question family: <QUESTION_FAMILY>
```

```
Requested difficulty: <DIFFICULTY>
Requested distractor labels: <TARGET_ERROR_PROFILE>

Select the question-required operands first. Every operand must cite a
fact_id and retain its source_image and semantic role. Then specify one
operator that maps those operands to the answer. Write a concise,
natural English question whose answer cannot be obtained by ignoring
the declared source scope. A valid item may contrast one target source
against distractor sources or compose evidence from multiple images.

Create 3-5 grammatically parallel options with exactly one correct
answer. Every incorrect option must differ from the gold construction
by one primary controlled error:

- operand_substitution: replace a required role with another grounded
  fact. Using a correct fact from the wrong image is subtype
  wrong_source.
- operand_omission: leave at least one required role unfilled.
- operator_error: keep the required operands but apply a non-equivalent
  comparison, ordering, arithmetic, quantifier, set, or relation rule.
- atom_veracity_error: use a fact unsupported by every image. This
  label is permitted only when DOMAIN is Controlled.

Do not invent evidence, combine two errors in one option, reveal the
answer in the question, or use options with different specificity.
Reject the item if the pixels and verified facts do not support a
unique answer or a requested diagnostic distractor.

Return JSON only:
{
  "status": "ok|reject",
  "question": "...",
  "operands": [
    {"role": "...", "source_image": 1, "fact_id": "I1_F1"}
  ],
  "operator": {"name": "...", "expression": "..."},
  "options": [
    {
      "label": "A",
      "text": "...",
      "is_correct": true,
      "error_type": null,
      "error_subtype": null,
      "evidence_edit": "none"
    }
  ],
  "correct_option": "A",
  "correct_answer": "...",
  "reject_reason": ""
}
```

Here, <QUESTION_FAMILY> selects among source attribution, attribute binding, counting and aggregation, comparison, spatial or action relations, set reasoning, logical quantification, multi-hop composition, and mixed reasoning. <TARGET_ERROR_PROFILE> controls dataset-level error balance without forcing an error type that the available evidence cannot instantiate.

Independent pixel-level validation. The closed-source multimodal judge evaluates one proposed fact against one image at a time. This produces the image–fact support matrix without exposing captions, metadata, or support claims from other images.

```
You are an independent pixel-level verifier. Exactly one image is
attached. Evaluate the proposition below against this image only.

Global image index: <IMAGE_INDEX>
```

```
Proposition: <ATOMIC_FACT>

Return true only when the complete proposition is directly and
unambiguously visible to an ordinary person at normal viewing size.
Return false when the pixels contradict it. Return ambiguous when the
relevant object or relation is hidden, tiny, mixed, subjective, or
otherwise uncertain. Do not infer from filenames, captions, metadata,
dataset labels, or world knowledge. A true verdict must name the
specific image region containing the evidence.

Return JSON only:
{
  "image_index": <IMAGE_INDEX>,
  "verdict": "true|false|ambiguous",
  "confidence": 0.0,
  "evidence_location": "...",
  "visible_evidence": "...",
  "reason": "..."
}
```

The returned confidence and concrete evidence location make each verdict auditable. The prompt payload contains no caption or metadata field.

Item-level adjudication. The item-level template gives the judge the complete candidate structure and asks it to check the mapping from verified operands to the answer. This prompt is evaluative only: it cannot rewrite or repair a rejected item.

```
You are the final adjudicator for a multi-image visual QA item. The
attached images are Image 1 through Image <NUM_IMAGES>.

Question: <QUESTION>
Options: <OPTIONS>
Declared operands: <OPERANDS_WITH_SOURCE_AND_ROLE>
Declared operator: <OPERATOR>
Verified image-fact support matrix: <SUPPORT_MATRIX>
Declared distractor annotations: <DISTRACTOR_ANNOTATIONS>

Check all of the following from the pixels and supplied verified facts:
1. Each operand is supported by its declared image and fills the role
   required by the question.
2. The operator matches the operation requested by the question.
3. Applying the operator to the operands yields the declared answer.
4. Exactly one option is correct.
5. Each distractor is invalid for its declared primary error and does
   not introduce an additional primary error.
6. The question is natural and answerable without captions, metadata,
   specialist knowledge, subjective judgement, or unreadable detail.

Return JSON only:
{
  "operands_valid": true,
  "operator_valid": true,
  "answer_derivable": true,
  "answer_unique": true,
  "distractor_labels_valid": true,
  "ordinary_human_answerable": true,
  "alternative_valid_options": [],
  "failed_checks": [],
  "confidence": 0.0,
  "verdict": "accept|reject",
  "reason": "..."
}
```

Controlled-domain symbolic design. The controlled-domain design template defines a provisional scene specification and its intended evidence structure. Its output represents A_{prompt}, not A^*.

```
Design one controlled multi-image CARE-Bench item.

Number of images: <NUM_IMAGES>
Question family: <QUESTION_FAMILY>
Difficulty: <DIFFICULTY>
Scene-mode allocation: <SCENE_MODE_ALLOCATION>
Target distractor labels: <TARGET_ERROR_PROFILE>

Define three to five independent scenes. Use at least two images as
question evidence and place true but irrelevant facts in the remaining
visual content. For a contextual scene, choose people, objects, actions,
and surroundings that naturally belong together. For an abstract scene,
use a small set of high-contrast geometric objects. Avoid labels, logos,
specialist knowledge, and relations that depend on small distance or
fine visual differences.

Assign a unique object ID to every instance. Write each intended fact as
one predicate over those IDs. Select the question-required operands from
different source images, specify the operator, and compute the intended
answer. Design three parallel distractors according to the requested
error profile. Reject the design if the symbolic specification permits
more than one answer.

Return JSON only:
{
  "status": "ok|reject",
  "images": [
    {
      "image_index": 1,
      "scene_mode": "contextual|abstract",
      "scene_description": "...",
      "prompt_atomic_facts": ["predicate(arguments)"],
      "question_relevant_fact_ids": [],
      "distractor_fact_ids": []
    }
  ],
  "question_draft": "...",
  "operands": [],
  "operator": {"name": "...", "expression": "..."},
  "intended_answer": "...",
  "options": [],
  "reject_reason": ""
}
```

Controlled-domain image generation. The image-generation request instantiates one symbolic scene at a time. Separate policies for contextual and abstract scenes preserve visual diversity while retaining control over atomic facts.

```
Generate one standalone square image, not a collage or contact sheet.

Scene mode: <CONTEXTUAL_OR_ABSTRACT>
Required objects, attributes, and actions: <OBJECT_SPECIFICATION>
Required spatial relations: <RELATION_SPECIFICATION>
Visual style: <STYLE_SPECIFICATION>

Render every required object completely and make each required
attribute, action, count, and relation easy to verify. Preserve the
specified subject and reference order for directional relations. Do
not add salient objects that could change a count, satisfy a queried
category, or create a competing relation. Avoid occlusion, ambiguous
```

```
colors, reflections, labels, captions, logos, and watermarks.

<SCENE_MODE_POLICY>
```

For contextual scenes, <SCENE_MODE_POLICY> requests a coherent real-world environment in which the people, objects, and actions are plausible for the setting. It explicitly forbids placing unrelated props in front of an arbitrary scenic background. For abstract scenes, the policy requests a clean studio-like composition with high-contrast geometric objects and a neutral background. The blind reconstruction request reuses the atomic-fact extraction template with DOMAIN=Controlled. Its input contains the rendered image but no A_{prompt} field.

Blinded corruption test for judge reliability. The corruption-audit template is identical for clean and deliberately corrupted records. Its input omits the expected corruption label.

```
You are auditing one CARE-Bench evidence record. The record may be
valid or may contain one controlled corruption. Inspect the attached
images and determine whether the complete construction is consistent.

Question: <QUESTION>
Candidate answer: <CANDIDATE_ANSWER>
Operands with source and role: <OPERANDS>
Operator: <OPERATOR>

Check visual support, source assignment, role alignment, operand
completeness, operator-question consistency, and answer derivability.
Do not assume that the supplied record is valid. Identify one primary
failure and a subtype when possible.

Allowed failure_type values are:
none, atom_veracity_error, operand_substitution,
operand_omission, operator_error, answer_invalid.

Return JSON only:
{
  "valid": true,
  "failure_type": "<FAILURE_TYPE>",
  "failure_subtype": "none|wrong_source|wrong_object|wrong_role|other",
  "affected_operands": [],
  "confidence": 0.0,
  "reason": "..."
}
```

Source reassignment is scored as the wrong_source subtype of Operand Substitution. Answer-validity corruption is used to test the adjudicator rather than to define an additional CARE-Bench diagnostic category.

C.4 CARE-Bench Composition and Construction Reliability

Table 5 summarizes the composition of CARE-Bench across its three visual domains. COCO-VQA provides most examples and covers broad everyday visual content, while Te Papa introduces less familiar real-world material, and the Generated domain provides controlled visual evidence. Most questions contain four images, although the generated domain deliberately spans three, four, and five-image settings more evenly. Evidence complexity also differs substantially across domains: COCO-VQA contains the largest proportion of high-complexity questions, Te Papa is weighted toward lower-complexity cases, and the Generated domain contains a larger share of medium-complexity questions. Such variation prevents the benchmark from testing only a single source burden or composition pattern.

Domain imbalance also motivates reporting domain-macro results alongside aggregate accuracy. COCO-VQA accounts for 11,132 of the 12,000 questions, so a question-weighted score would otherwise be dominated by that domain. Source statistics further show that the required visual support

varies across domains, ranging from relatively sparse use of sources in Te Papa to substantially broader use of multi-image evidence in COCO-VQA and the Generated domain.

Table 5: CARE-Bench composition statistics. Image mix, E/M/H, and distractor-error columns are percentages. Avg. sources counts distinct question-relevant image sources. E/M/H follows the evidence-complexity policy defined by source burden, operator complexity, negation, global image use, input size, and visual ambiguity. Note that in the Generated line, three error types are not exhaustive.

Domain	Q	Image slots	Unique	3/4/5 images	Avg. sources	E/M/H	Sub.	Omit.	Op.
COCO-VQA	11,132	44,959	32,507	0.0/96.1/3.9	3.16	34.7/24.5/40.8	37.52	24.61	37.87
Te Papa	368	1,405	1,248	18.2/81.8/0.0	1.29	67.1/15.2/17.7	69.71	10.92	19.37
Generated	500	1,925	1,925	40.0/35.0/25.0	2.91	30.0/50.0/20.0	38.41	17.68	27.61
Total	12,000	48,289	35,680	2.2/93.1/4.6	3.09	35.5/25.3/39.2	38.63	23.71	36.86

Table 6 evaluates the reliability of the automatic validation procedure through controlled structural corruptions. Detection reaches at least 98.00% for operand substitution, operand omission, and operator errors, while error-type accuracy reaches at least 97.00% in every category. Similar performance across the three corruption types indicates that validation is not driven only by easily detectable missing evidence. Operator corruption is slightly more difficult, but still reaches 98.00% detection and 97.00% type accuracy.

Controlled injection provides a targeted check of whether the judge recognizes known structural violations while all unaffected parts of an instance remain fixed. High injection accuracy, therefore, supports the use of the judge for automatic quality control, although it does not, by itself, ensure perfect detection of every naturally occurring construction error.

Table 6: Judge reliability under controlled structural-error injection (%). Detection measures whether the judge rejects a corrupted instance; Type Acc. additionally requires the predicted label to match the injected error. Each corruption changes only the indicated operand or operator while holding the remaining instance fixed.

Injected error	Construction	n	Detection ↑	Type Acc. ↑
Operand substitution	Replace one required operand with a grounded fact from the wrong object or source.	300	99.33	98.00
Operand omission	Remove one required operand while leaving the remaining evidence unchanged.	300	98.67	97.67
Operator error	Keep the operands fixed but use a composition inconsistent with the question.	300	98.00	97.00
Macro average	–	900	98.67	97.56

C.5 Domain and Evidence-Complexity Breakdown

Table 7 reports CARE-Bench performance separately for each visual domain. CARE-Flow improves accuracy across all backbone-domain combinations, with gains ranging from 6.40 to 14.19 percentage points. Improvements, therefore, do not depend on a single image source or content distribution. Te Papa yields the largest gains for LLaVA-NeXT-7B and Qwen3.5-9B, while substantial improvements remain on both COCO-VQA and the controlled-generated domain.

Error reductions differ across backbones and domains, reflecting different underlying failure profiles. LLaVA-NeXT-7B reduces operand substitution on Te Papa by 9.95 points, while its largest reduction on the Generated domain occurs in operator errors at 7.80 points. Qwen3.5-9B similarly reduces Generated-domain operator errors by 7.80 points, and InternVL3-8B reduces operator errors on Te Papa by 7.44 points. CARE-Flow, therefore, does not merely shift all errors uniformly but affects the structural failure modes that dominate each model and domain. Other errors in the Generated domain decrease by 1.20, 0.40, and 0.20 percentage points for LLaVA-NeXT-7B, Qwen3.5-9B, and InternVL3-8B, respectively.

Table 7: Domain-wise CARE-Bench results (%) on the full releases. Error columns are fractions of all questions. Other denotes errors not led by inter-image hallucination, and ΔAcc. compares CARE-Flow with the corresponding Vanilla row.

Backbone	Domain	n	Method	Acc.	Sub.	Omit.	Op.	Other	ΔAcc.
LLaVA-NeXT-7B	COCO-VQA	11,132	Vanilla	27.18	33.31	16.83	22.68	–	–
			CARE-Flow	38.72	29.69	16.26	15.33	–	11.54
	Te Papa	368	Vanilla	45.92	35.60	8.37	10.11	–	–
			CARE-Flow	60.11	25.65	7.83	6.41	–	14.19
	Generated	500	Vanilla	26.40	28.40	4.80	23.60	16.80	–
			CARE-Flow	39.20	24.80	4.60	15.80	15.60	12.80
Qwen3.5-9B	COCO-VQA	11,132	Vanilla	41.53	19.18	31.46	7.83	–	–
			CARE-Flow	50.21	16.82	27.43	5.54	–	8.68
	Te Papa	368	Vanilla	47.61	14.18	24.13	14.08	–	–
			CARE-Flow	59.51	12.39	19.46	8.64	–	11.90
	Generated	500	Vanilla	43.00	6.20	9.80	26.40	14.60	–
			CARE-Flow	51.40	6.20	9.60	18.60	14.20	8.40
InternVL3-8B	COCO-VQA	11,132	Vanilla	29.02	23.11	34.82	13.05	–	–
			CARE-Flow	36.63	22.09	32.12	9.16	–	7.61
	Te Papa	368	Vanilla	33.32	17.72	23.04	25.92	–	–
			CARE-Flow	40.71	17.66	23.15	18.48	–	7.39
	Generated	500	Vanilla	30.20	8.60	21.80	21.20	18.20	–
			CARE-Flow	36.60	8.80	21.40	15.20	18.00	6.40

Table 8 further stratifies CARE-Bench by evidence complexity. CARE-Flow gains increase consistently from Easy to Medium to Hard across all three backbones. Improvements for LLaVA-NeXT-7B increase from 8.69 to 11.52 and 14.50 points; those for Qwen3.5-9B increase from 5.30 to 8.29 and 12.20 points; and those for InternVL3-8B increase from 4.96 to 8.13 and 9.52 points. CARE-Flow also substantially narrows the Easy-to-Hard performance gap, from 11.69 to 5.88 points for LLaVA-NeXT-7B, 13.51 to 6.61 for Qwen3.5-9B, and 12.59 to 8.03 for InternVL3-8B. These larger gains on structurally demanding questions, together with the reduced difficulty gap, are consistent with CARE-Flow improving evidence selection and composition as the number and complexity of evidence dependencies increase.

Evidence complexity is defined by structural properties such as source burden, operator complexity, negation, and global image use, rather than by empirical model accuracy. Absolute accuracy is therefore not required to decrease monotonically from Easy to Hard. Results should instead be interpreted as indicating that the advantage of CARE-Flow increases with the complexity of structural evidence.

Table 8: Backbone performance across difficulty levels. Accuracy (%) is CARE-Bench difficulty breakdown. Each entry reports the mean value over five runs. Avg. is the unweighted mean over Easy, Medium, and Hard, while E–H Gap is the absolute accuracy decrease from Easy to Hard.

Backbone	Method	Easy	Medium	Hard	Avg.	E–H Gap
LLaVA-NeXT-7B	Vanilla	33.48	28.83	21.79	28.03	11.69
	CARE-Flow	42.17	40.35	36.29	39.60	5.88
Qwen3.5-9B	Vanilla	48.42	43.11	34.91	42.15	13.51
	CARE-Flow	53.72	51.40	47.11	50.74	6.61
InternVL3-8B	Vanilla	35.48	30.16	22.89	29.51	12.59
	CARE-Flow	40.44	38.29	32.41	37.05	8.03

C.6 Scaling with Backbone Size

Table 9 examines whether CARE-Flow remains effective as the Qwen3.5 backbone scales from 2B to 27B parameters. CARE-Flow improves domain-macro accuracy across all model sizes, with gains of 11.33, 6.65, 9.66, and 5.94 percentage points for the 2B, 4B, 9B, and 27B models, respectively. Benefits are therefore preserved across substantially different model capacities, although their magnitude is not monotonic with parameter count.

Table 9: Qwen3.5 scaling study on CARE-Bench (%). Vanilla and CARE-Flow share the same frozen checkpoint and candidate-commitment scoring. CARE-Flow applies exact source-grouped value-output intervention to Qwen3.5's full-attention blocks, while hybrid linear-attention blocks remain unchanged. Avg. is the unweighted domain mean; Sub., Omit., and Op. are domain-macro error rates. Δ_{CARE} compares CARE-Flow with Vanilla at a fixed model size.

Backbone	Method	COCO-VQA ↑	Te Papa ↑	Generated ↑	Avg. ↑	Sub. ↓	Omit. ↓	Op. ↓	Δ_{CARE} ↑
Qwen3.5-2B	Vanilla	33.31	16.79	26.60	25.57	30.29	11.91	28.90	–
	CARE-Flow	37.28	26.63	46.80	36.90	29.58	8.04	25.48	+11.33
Qwen3.5-4B	Vanilla	41.22	30.00	40.20	37.14	28.80	12.49	21.57	–
	CARE-Flow	46.73	38.04	46.60	43.79	28.02	9.13	18.95	+6.65
Qwen3.5-9B	Vanilla	41.53	47.61	43.00	44.05	13.19	21.80	16.10	–
	CARE-Flow	50.21	59.51	51.40	53.71	11.80	18.83	10.93	+9.66
Qwen3.5-27B	Vanilla	49.97	60.11	49.80	53.29	19.52	13.23	13.95	–
	CARE-Flow	53.21	66.09	58.40	59.23	23.97	5.49	11.31	+5.94

For the 2B, 4B, and 9B backbones, CARE-Flow reduces all three structural error categories. The 27B model exhibits a different redistribution: operand omission decreases from 13.23% to 5.49% and operator error from 13.95% to 11.31%, while operand substitution increases from 19.52% to 23.97%. Accuracy nevertheless improves by 5.94 points, despite this increase in substitution. Larger models can therefore retain a different residual error profile even when the overall intervention remains beneficial.

C.7 Atom Veracity and the Scope of CARE-Flow

Table 10 separates structural evidence-flow errors from failures in the visual atoms themselves. The controlled generated domain provides an exhaustive, verified fact inventory, allowing every question-relevant model-generated atom to be aligned with known visual evidence. Atom output and judge coverage exceed 98% across all backbones and methods, ensuring broad coverage of the resulting analysis.

Table 10: Atom-veracity analysis on the Generated domain. n and Err. are counts; the remaining diagnostic columns are percentages. An independent model aligns each generated question-relevant atom to the exhaustive verified inventory. Atom Conflict marks examples containing at least one unsupported or contradictory atom. A-V requires both an incorrect final answer and an atom conflict. A fact aligned to another image remains a grounded operand substitution and is not counted as A-V.

Backbone	Method	n	Err.	Atom output	Judge cov.	Atom Conflict	A-V	A-V / Err.
LLaVA-NeXT-7B	Vanilla	500	368	98.60	98.20	22.40	16.80	22.83
LLaVA-NeXT-7B	CARE-Flow	500	304	99.00	98.60	22.00	15.60	25.66
Qwen3.5-9B	Vanilla	500	285	99.20	98.80	19.20	14.60	25.61
Qwen3.5-9B	CARE-Flow	500	243	99.40	99.00	18.80	14.20	29.22
InternVL3-8B	Vanilla	500	349	98.80	98.40	23.60	18.20	26.07
InternVL3-8B	CARE-Flow	500	317	99.00	98.60	23.40	18.00	28.39

CARE-Flow substantially reduces the number of incorrect final responses but changes atom-conflict rates only marginally. Atom Conflict decreases from 22.4% to 22.0% for LLaVA-NeXT-7B, from 19.2% to 18.8% for Qwen3.5-9B, and from 23.6% to 23.4% for InternVL3-8B. A-V errors show similarly small absolute changes. By contrast, total error counts fall from 368 to 304, 285 to 243, and 349 to 317, respectively.

Such behavior matches the intended scope of CARE-Flow. Source certification tests whether an image path supports a proposed operand, and complete mediation controls how certified operands enter downstream inference, but neither operation guarantees that the model's underlying visual representation has correctly extracted a fact. A strongly supported but visually incorrect atom can therefore survive certification. CARE-Flow primarily reduces errors arising from source selection, omission, and evidence composition, while atom-level visual veracity remains a separate limitation.

A-V consequently accounts for a larger fraction of the remaining errors after CARE-Flow, increasing from 22.83% to 25.66%, 25.61% to 29.22%, and 26.07% to 28.39% for LLaVA-NeXT-7B, Qwen3.5-9B, and InternVL3-8B, respectively. Higher conditional proportions do not indicate that CARE-Flow creates more atom-veracity failures; absolute A-V rates decrease slightly in every case. Instead, structural errors are removed more rapidly than atom-veracity errors, leaving visual fact extraction as a larger component of the residual error set.

CARE-Flow consistently improves performance by reducing inter-image hallucination through more reliable evidence selection and use. All models are evaluated under their default inference settings, without task-specific system-prompt engineering or additional optimization for CARE-Bench. This controlled setup isolates the contribution of CARE-Flow, but also means that the reported absolute accuracy should not be interpreted as the performance ceiling of the underlying models on multi-image reasoning. The remaining performance gap, therefore, points to complementary opportunities beyond the flow of evidence itself and partly reflects task difficulty. In particular, future work may further improve the fidelity of visual-to-atomic evidence extraction, strengthen general multimodal reasoning over certified evidence, and explore more adaptive inference strategies for questions of varying complexity. These directions are complementary to CARE-Flow and could further improve overall multi-image reasoning while preserving its gains in hallucination mitigation.