COMPOSITION, NOT CONVERSATION: VLMS LOSE THE SCENE, NOT THE THREAD

Anonymous authors
Paper under double-blind review

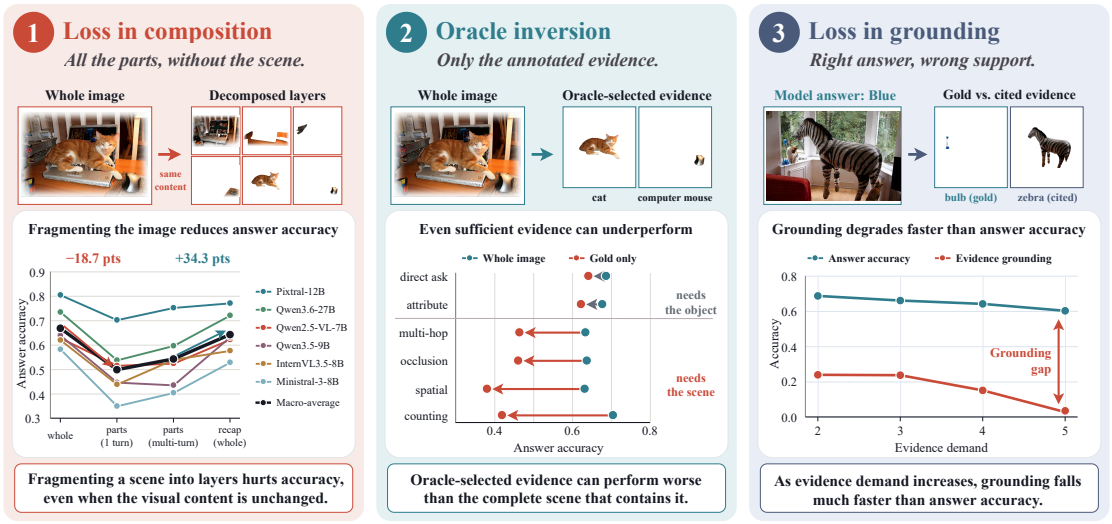

Figure 1: **Vision-language models lose the scene, not the thread.** ***Left:*** Scene fragmentation hurts accuracy, while recomposition recovers it. ***Middle:*** Oracle-selected minimal evidence can perform worse than the complete scene. ***Right:*** As evidence demand grows, grounding falls much faster than answer accuracy.

ABSTRACT

Vision-language models (VLMs) increasingly reason over visual evidence that is cropped, segmented, retrieved, or revealed over time. Yet most VQA benchmarks present the complete image and question at once. We ask what models lose when the same information is fragmented. We introduce **Layered-VQA**, with 93 scenes and 300 questions. Each image is decomposed into ordered RGBA layers that exactly recompose the original scene, and each question is annotated with supporting, minimal-sufficient, and distractor layers. We evaluate eleven open-weight VLMs from 3B to 32B parameters and two proprietary models with a scale of $187{,}200$ conversations, graded by $1.74M$ open-model cross-judgments. We find three consistent failures. ***Loss in Composition:*** fragmenting the question has a small effect, but fragmenting the scene substantially reduces accuracy; recomposing the same layers largely restores performance. ***Oracle Inversion:*** even oracle-selected sufficient evidence can perform worse than the complete scene. ***Loss in Grounding:*** as more evidence is required, grounding degrades much faster than answer accuracy. Together, these results show that having the right visual evidence is not enough. How that evidence is composed and presented determines whether models can use and ground it. ***The right evidence is not enough: VLMs need the scene it came from.***

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.

1 INTRODUCTION

The performance of a VLM may depend not only on what information is available but also on *how* that information is presented. This question is becoming increasingly relevant with the rise of multimodal agents, which reason and act over multiple steps while interacting with tools and evolving visual observations (Koh et al., 2024). In practical settings, visual evidence may be cropped, segmented, retrieved, or revealed incrementally (Gupta & Kembhavi, 2023; Surís et al., 2023; Hu et al., 2023; Liu et al., 2024a), while user queries may also unfold or be revised across turns (Das et al., 2017; Liu et al., 2024d). This creates a growing mismatch between how VLMs are evaluated and how they increasingly operate in practice. Existing visual question answering (VQA) and general-purpose VLM benchmarks typically evaluate models in a standard complete-input setting: one complete image, one complete question, and one turn (Goyal et al., 2017; Hudson & Manning, 2019; Liu et al., 2024c; Yu et al., 2023). We therefore ask a simple question: *How does the delivery of visual information affect VLM performance when the underlying information remains unchanged?* In particular, we seek to understand whether the same visual information leads to different model behavior when its composition, delivery across turns, or alignment with the evolving query changes.

Recent work on language models provides an explanation: when a complete instruction is fragmented across turns, performance degrades as models commit prematurely and fail to revise their answers as the interaction unfolds (Laban et al., 2026). A similar effect may be expected for VLMs, particularly as visual evidence is increasingly distributed across long, multi-turn, or retrieval-based contexts (Song et al., 2024; Wang et al., 2025a; 2024; Wu et al., 2025). Yet visual information introduces an additional source of complexity: distributing an image does not only change when information arrives, but can also break the composition of the scene itself. Consequently, a drop in VLM performance can have several explanations: the model may struggle with information arriving across turns, with integrating fragmented visual content into a coherent scene, or with information that is lost during evidence selection. Existing evaluations do not disentangle these effects, making it difficult to determine whether the primary challenge lies in the conversation, the visual composition, or the evidence available to the model. Isolating these factors therefore requires a setting in which the underlying visual information is controlled while only its form and delivery are varied.

To isolate these factors, we introduce ***Layered-VQA***, a controlled benchmark in which the same underlying visual information can be presented in different forms. Each scene consists of a composite image and an ordered set of RGBA layers that exactly recompose the original scene (Tudosiu et al., 2024). We manually author every question and annotate the layers that carry the answer, the smallest set of layers sufficient to answer it, and plausible but irrelevant layers. This structure allows us to systematically vary how the same visual information is delivered: as a complete scene or fragmented into layers, within a single turn or across multiple turns, and aligned or misaligned with the question components it supports. We additionally recompose fragmented evidence and construct oracle conditions containing only the minimally sufficient layers, allowing us to separate the effects of visual composition, conversational delivery, and evidence selection. Across these controlled conditions, the underlying scene and question remain unchanged; what changes is how the information is presented to the model.

Our experiments reveal three main findings, Figure 1. First, we find a clear ***Loss in Composition***: fragmenting the question has relatively little and inconsistent effect, whereas fragmenting the visual scene into layers consistently degrades performance. This loss already emerges when all layers are presented together in a single turn, becomes larger when the same layers are distributed across turns, and is largely recovered when the layers are recomposed into the original image. These results separate the cost of visual fragmentation from the additional cost of multi-turn interaction, pointing to scene composition as a central bottleneck. Second, we uncover an ***Oracle Inversion***: even when models are given only the oracle-selected minimal set of layers sufficient to answer the question, they perform worse than when given the complete scene. Thus, removing seemingly unnecessary visual content can make reasoning harder, indicating that surrounding scene structure contributes to how VLMs interpret otherwise sufficient evidence. Finally, we observe a ***Loss in Grounding***:

models lose the ability to identify the visual evidence supporting their predictions much faster than they lose answer accuracy. Correct answers can therefore mask substantial failures in evidence attribution, revealing a gap between producing the right answer and producing it for the right visual evidence.

2 Related Work

Multi-turn Interaction and Visual Composition: Multi-turn benchmarks ask whether models can carry information across an interaction. In text, Laban et al. (2026) show that splitting one instruction across turns can sharply reduce performance. *Visual Dialog* keeps one image while questions unfold through dialogue history (Das et al., 2017), *ConvBench* chains perception, reasoning, and creativity (Liu et al., 2024b), and *MMDU* extends this setting to long multi-image conversations (Liu et al., 2024d). At longer contexts, *MileBench* measures reasoning over many images (Song et al., 2024), while *MM-NIAH*, *MMNeedle*, and *Visual Haystacks* place relevant visual evidence among increasingly large or distracting contexts (Wang et al., 2024; 2025a; Wu et al., 2025). These works vary dialogue history, context length, image count, or distractors, where as Layered-VQA preserves the source scene content and object coordinates, and varies only composition and delivery across turns.

Compositional benchmarks ask whether models bind objects, attributes, and relations correctly, Layered-VQA instead asks whether these relations survive scene fragmentation. *CLEVR* and *GQA* use structured scenes and questions to diagnose compositional reasoning (Johnson et al., 2017; Hudson & Manning, 2019). *Winoground* tests relations through matched image-caption pairs (Thrush et al., 2022), *ARO* tests attribution, relations, and word order (Yuksekgonul et al., 2022), and *SugarCrepe* builds harder contrastive negatives while reducing linguistic shortcuts (Hsieh et al., 2023). Campbell et al. (2024) further connect multi-object failures to feature-binding interference. Layered-VQA removes only the composed view, asking whether models can recover the scene when the same visual content is supplied as separate layers.

Decomposed Evidence and Grounding: Image decomposition and visual tools change what evidence a model receives. *SAM* produces object masks (Kirillov et al., 2023), *pix2gestalt* completes partially hidden objects (Ozguroglu et al., 2024), *MuLAn* provides ordered RGBA decompositions for controllable image generation (Tudosiu et al., 2024), and Transparent Image Layer Diffusion generates images as consistent transparent layers (Zhang & Agrawala, 2024). Layered-VQA uses the layered scenes from *MuLAn*, manually authors questions over them, and instead asks what visual reasoning is lost when these components replace the composed scene. Other work selects evidence during reasoning. *VisProg* and *ViperGPT* execute modular visual programs (Gupta & Kembhavi, 2023; Surís et al., 2023), *AVIS* and V* search for useful visual information (Hu et al., 2023; Wu & Xie, 2024), and *MuRAG* retrieves relevant multimodal evidence from external memory (Chen et al., 2022). Layered-VQA does not test whether a model can discover evidence. *The evidence is fixed and annotated, allowing us to control for retrieval as a potential confound and ask a different question: once the relevant visual evidence is available, does selecting or separating it preserve the reasoning supported by the complete scene?*

This distinction also matters for grounding. Jacovi & Goldberg (2020) separate plausible explanations from faithful ones, *ERASER* evaluates whether annotated rationales are sufficient and necessary (DeYoung et al., 2020), and Turpin et al. (2023) show that model explanations can omit factors that changed the answer. In VQA, Reich & Schultz (2024) show that the benefits of visual grounding methods are underestimated because visual features frequently omit the question-relevant information such methods rely on. *Hallusion-Bench* diagnoses visual illusion and language hallucination through controlled question pairs (Guan et al., 2024). Layered-VQA makes the supporting visual evidence explicit. Each question has gold, minimal-sufficient, and distractor layers, and models must cite the layers supporting their answer. This exposes a failure that answer accuracy alone cannot show: a model can give the right answer while citing the wrong visual evidence, without implying that the cited layers faithfully reflect its internal reasoning.

3 LAYERED-VQA

Layered-VQA turns each VQA example into a fixed set of visual and linguistic components that can be presented in different forms. Each example contains a composite scene I, its RGBA layers $\mathcal{L}$, a question Q, an answer a, question shards $\mathcal{Q}$, and explicit evidence annotations. The scene, question, and answer stay fixed across conditions. Only how they are composed and delivered changes. We begin with 200 candidate scenes, retain 93 that support at least one reasoning category, and construct 300 questions across six categories. The entire benchmark preparation is given in Figure 2.

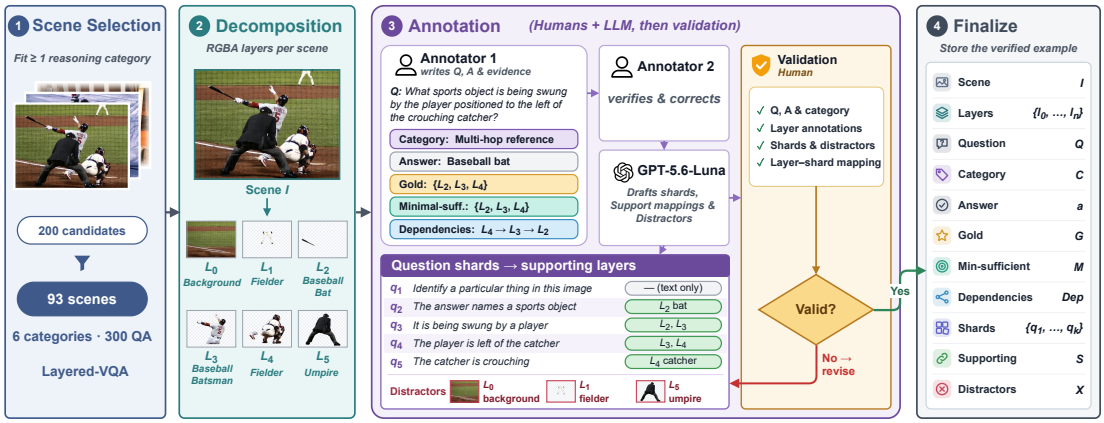

Figure 2: **Layered-VQA construction.** Scenes are decomposed into RGBA layers, manually annotated, expanded into question shards and evidence mappings with GPT-5.6-Luna, and verified before finalization.

Layered Scenes: Each scene comes from MuLAn (Tudosiu et al., 2024) and contains a composite image I with an ordered stack of RGBA layers $\mathcal{L} = \{L_0, \ldots, L_n\}$. MuLAn layers are object instances on the full canvas. The layers exactly recompose the original scene. L_0 contains the background, while the remaining layers contain individual objects on the original full-size canvas. The source scene and canvas coordinates therefore remain fixed when moving between the composite and its layers. This gives two views of the same scene: the composed image I and its decomposed representation $\mathcal{L}$.

Questions, Evidence, and Verification: We write questions in six categories: *Direct ask*, *Attribute binding*, *Counting*, *Spatial*, *Occlusion*, and *Multi-hop reference*. For each scene, Annotator 1 writes the category C, complete question Q, answer a, gold evidence G, minimal-sufficient evidence M, and evidence dependencies Dep. G contains the layers supporting the complete question. M is the smallest layer set from which a person can still determine the answer. Dep records dependencies between evidence layers. Annotator 2 checks all of these annotations and corrects disagreements after discussion. GPT-5.6-Luna then decomposes the verified question into ordered shards $\mathcal{Q} = \{q_1, \ldots, q_m\}$. For each shard q_i, it also assigns the supporting layer set S_i and proposes distractor layers X. A shard may be text-only, in which case $S_i = \varnothing$. Together, the shards preserve the information in the complete question, while the support mapping makes the correspondence between each question part and its visual evidence explicit. Distractors belong to the same scene but are not required to determine the answer. The complete annotation is then checked by human. Validation covers the question, answer, and category; the gold, minimal-sufficient, and dependency annotations; the shards and distractors; and every shard-to-layer mapping. Incorrect annotations are revised and checked again before the sample is finalized. Each verified example stores $(I, \mathcal{L}, Q, C, a, G, M, \text{Dep}, \mathcal{Q}, S, X)$. When layers are shown to a model, they are referred to only by numerical indices, so semantic layer names

cannot reveal the answer. Layered-VQA contains 83 *Direct ask*, 65 *Attribute binding*, 54 *Spatial*, 44 *Counting*, 37 *Multi-hop reference*, and 17 *Occlusion* questions. Questions contain 3 to 6 shards, most commonly 5, and require 1 to 6 gold layers. Every question contains at least one distractor, and 197 questions contain an evidence dependency. These annotations let the same VQA problem be evaluated under different forms of composition, timing, evidence order, and question-to-evidence alignment.

4 Experimental Setup

Starting from the Layered-VQA representation introduced in Section 3, we construct controlled conditions that vary how the question shards and visual layers are presented to a VLM. Our experiments are designed to separate three factors: whether the visual scene is composed or fragmented, whether information is delivered in one turn or across multiple turns, and which visual evidence is available to the model.

Controlled Delivery Conditions: We evaluate sixteen conditions, summarized in Figure 3. Fourteen preserve the complete question and visual content and change only how they are composed or delivered. The remaining two, ALL-LAYERS and ORACLE, modify the available evidence for grounding and oracle-evidence evaluation. We organize the conditions according to which component of the Layered-VQA representation is manipulated.

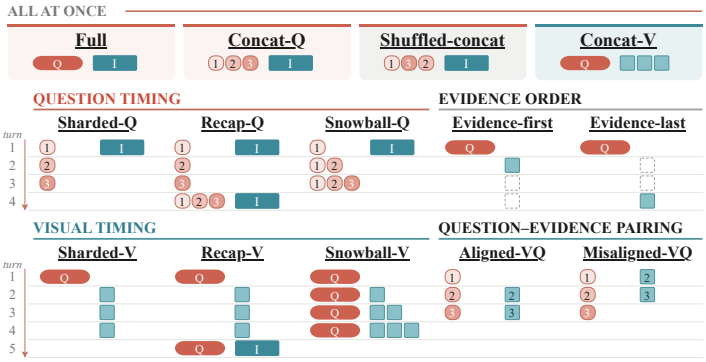

Figure 3: **Controlled delivery conditions.** The sample is presented under different question, visual, alignment, and evidence schedules.

Question Delivery: The 1st family varies how the question is presented while keeping the composite scene available. **FULL** presents the complete question Q and composite image I in one turn. **CONCAT-Q** presents all question shards together, while **SHUFFLED-CONCAT** presents the same shards in random order. **SHARDED-Q** reveals one shard per turn. **RECAP-Q** ends the sharded conversation by restating the complete question. **SNOWBALL-Q** repeats all previous shards whenever a new shard arrives. **Visual Delivery:** The 2nd family keeps the complete question fixed and varies the presentation of the visual scene. **CONCAT-V** presents every layer together without the composite image. **SHARDED-V** reveals one layer per turn. **RECAP-V** ends the sharded sequence by showing the recomposed scene. **SNOWBALL-V** repeats all previously revealed layers at every turn. **EVIDENCE-FIRST** and **EVIDENCE-LAST** keep the same layers but place the annotated evidence early or late.

Joint Question-Visual Alignment: The 3rd family fragments both streams. **ALIGNED-VQ** pairs each question shard with its annotated supporting layers introduced in Section 3. **MISALIGNED-VQ** preserves the same layers and per-turn layer counts but pairs each shard with layers that do not support it. This isolates whether local question-evidence alignment can compensate for a fragmented scene.

Evidence Intervention: The final family changes which representation of the visual evidence is available while keeping the complete question in a single turn. **ALL-LAYERS** presents the composite image together with every layer and asks the model to identify the layers supporting its answer. **ORACLE** removes the composite and distractors and presents only the annotated minimal-sufficient layers. These two measures: grounding when all evidence is available and can the annotated evidence replace the complete scene.

MULTI-TURN SIMULATION: For multi-turn conditions, we simulate an interaction between a user agent and the VLM being evaluated. The user agent holds the unrevealed question shards and the condition-specific delivery schedule, while the VLM has access only to the information revealed up to the current turn. The VLM sees only the information revealed so far, and each response is appended to the conversation history. The interaction continues until all scheduled information has been delivered, even if the model answers correctly earlier. The assistant is told that the question and visual evidence may arrive across turns and that each layer is part of a decomposed scene. We use two user agents. The *scripted user* follows the fixed benchmark schedule. The *LLM user* selects a remaining question shard and rephrases it conversationally while preserving its information. It could reorder the layers, which would change the schedule under test. It uses the same model as the assistant at temperature 1.0 and forbids adding new facts, describing the image, or answering the question. The LLM user is used only for SHARDED-Q, RECAP-Q, and SNOWBALL-Q. Conditions with fragmented visual input use the scripted user to preserve the layer order and evidence schedule. Single-turn conditions use no user agent.

EVALUATION PROTOCOL: ANSWER ACCURACY: Answers are free-form, so completed conversations are evaluated by an LLM judge against the reference answer. We score the assistant's final response against the reference answer. Earlier correct answers do not count if the model later changes or withdraws them. This gives every condition one final prediction regardless of conversation length. Every completed conversation is independently graded by all eleven open models. Our primary metric is **Consensus Final-turn Accuracy**, obtained by majority vote across the eleven judges. RETAINED is a model's mean accuracy across a family's conditions, as a percentage of its own FULL score. On 400 conversations, the consensus matches humans 96.0% of the time ($\kappa = 0.92$), against 92.5% for the best single judge, more in Appendix A.8.

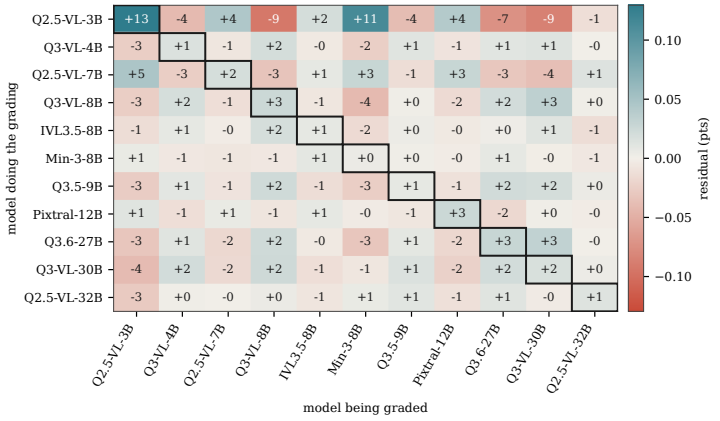

Figure 4: **Self-preference after cross-judging.** Boxed cells correspond to models judging their own outputs.

GROUNDING: In ALL-LAYERS, the model also reports the layers supporting its answer. We compare this set with the annotated evidence using precision, recall, F1, exact match, completeness, excess rate, distractor rate, and invalid rate. **Joint accuracy** requires both the answer and the predicted layer set to be correct. The **grounding gap** is the difference between answer accuracy and joint accuracy.

CROSS-JUDGING AND SELF-PREFERENCE: In the primary evaluation, each open model judges its own conversations. To quantify potential self-preference, we additionally re-grade every conversation with every open-model judge. The *self-preference residual* measures the remaining self-grading effect after accounting for assistant ability and judge leniency. This analysis is diagnostic, shown in Figure 4. Every model favours itself, by 2.6 points on average and by 12.9 for the smallest, so a single self-judge lifts exactly the models a benchmark needs to separate.

UNCERTAINTY: Every condition is run three times. Reported values are the mean ± standard deviation across these three runs. For the main paired comparisons, we additionally report 95% confidence intervals from a scene-level bootstrap, resampling scenes rather than individual questions.

Table 1: **What it costs to deliver the evidence across turns.** Consensus final-turn accuracy.

	BASE	QUESTION IN TEXT				QUESTION IN LAYERS				ALIGNMENT		ORDER			RETAINED %		
Model	FULL	CONCAT	SHARDED	RECAP	SNOWBALL	CONCAT	SHARDED	RECAP	SNOWBALL	ALIGNED	MISALIGNED	EV-FIRST	EV-LAST	SHUFFLED	Text	Visual	Layered
Qwen2.5-VL-3B	**.408** ±.02	.334 ±.01	.142 ±.01	.439 ±.01	.213 ±.05	.250 ±.02	.150 ±.02	.373 ±.01	.224 ±.03	.201 ±.03	.170 ±.01	.097 ±.02	.193 ±.03	.333 ±.03	69	61	41
Ministral-3-8B	**.463** ±.03	.433 ±.02	.199 ±.04	.402 ±.01	.200 ±.03	.247 ±.02	.180 ±.01	.523 ±.03	.272 ±.01	.143 ±.02	.176 ±.01	.113 ±.02	.298 ±.02	.454 ±.03	67	66	39
Qwen2.5-VL-7B	**.556** ±.02	.496 ±.02	.439 ±.03	.672 ±.01	.497 ±.01	.340 ±.02	.208 ±.03	.649 ±.03	.280 ±.01	.309 ±.01	.321 ±.03	.114 ±.01	.329 ±.01	.478 ±.04	95	66	48
Qwen2.5-VL-32B	**.559** ±.01	.549 ±.03	.533 ±.00	.711 ±.04	.630 ±.03	.426 ±.00	.327 ±.02	.708 ±.03	.433 ±.00	.337 ±.02	.367 ±.02	.274 ±.04	.432 ±.02	.557 ±.02	108	85	63
Pixtral-12B	**.572** ±.01	.556 ±.01	.393 ±.02	.577 ±.02	.471 ±.04	.360 ±.01	.233 ±.02	.579 ±.01	.392 ±.03	.271 ±.01	.299 ±.00	.154 ±.01	.271 ±.02	.502 ±.01	87	68	43
InternVL3.5-8B	**.577** ±.01	.524 ±.01	.343 ±.01	.696 ±.03	.376 ±.00	.386 ±.01	.300 ±.02	.618 ±.03	.349 ±.05	.410 ±.02	.424 ±.03	.284 ±.03	.362 ±.03	.481 ±.02	84	72	64
Qwen3.5-9B	**.656** ±.01	.612 ±.00	.542 ±.02	.762 ±.01	.638 ±.05	.443 ±.02	.398 ±.03	.751 ±.05	.492 ±.02	.332 ±.02	.330 ±.03	.329 ±.02	.504 ±.02	.607 ±.01	97	79	57
GPT-5.6-Luna†	**.693** ±.01	.694 ±.01	.654 ±.01	.715 ±.01	.681 ±.02	.642 ±.01	.621 ±.02	.689 ±.01	.638 ±.01	.563 ±.02	.585 ±.03	.621 ±.02	.619 ±.02	.688 ±.03	99	93	86
Qwen3.6-27B	**.697** ±.02	.679 ±.02	.533 ±.01	.783 ±.02	.650 ±.02	.502 ±.01	.460 ±.01	.762 ±.02	.563 ±.01	.370 ±.03	.391 ±.02	.417 ±.02	.529 ±.01	.683 ±.01	95	82	61
Qwen3-VL-4B	**.703** ±.01	.656 ±.00	.618 ±.02	.765 ±.01	.739 ±.03	.513 ±.01	.308 ±.00	.781 ±.00	.513 ±.04	.362 ±.04	.420 ±.02	.212 ±.01	.528 ±.01	.657 ±.00	99	75	54
Qwen3-VL-30B	**.722** ±.00	.676 ±.01	.600 ±.01	.760 ±.02	.716 ±.02	.567 ±.01	.476 ±.01	.742 ±.02	.501 ±.01	.511 ±.02	.556 ±.02	.479 ±.03	.501 ±.03	.664 ±.01	95	79	71
Qwen3-VL-8B	**.729** ±.01	.681 ±.00	.721 ±.01	.782 ±.03	.702 ±.03	.550 ±.01	.433 ±.01	.764 ±.02	.550 ±.00	.538 ±.02	.564 ±.02	.444 ±.03	.560 ±.03	.694 ±.00	99	79	72
Gemini-3.7-Flash†	**.862** ±.01	.858 ±.01	.835 ±.00	.860 ±.00	.844 ±.01	.850 ±.01	.836 ±.00	.860 ±.01	.835 ±.00	.805 ±.01	.783 ±.00	.842 ±.01	.847 ±.01	.862 ±.01	99	98	95
Mean, Open	**.604**	.563	.460	.668	.530	.417	.316	.659	.415	.344	.365	.265	.410	.555	90	74	56
Δ *vs* FULL	—	-.041	-.144	+.064	-.074	-.187	-.288	+.055	-.189	-.260	-.239	-.339	-.194	-.049			

4.1 Models and Implementation

We evaluate eleven open-weight VLMs from 3B to 32B parameters: Qwen2.5-VL-{3B, 7B, 32B} Bai et al. (2025b), Qwen3-VL-{4B, 8B, 30B-A3B} Bai et al. (2025a), InternVL3.5-8B Wang et al. (2025b), Ministral-3-8B Liu et al. (2026), Qwen3.5-9B Qwen Team (2026a), Pixtral-12B Agrawal et al. (2024), and Qwen3.6-27B Qwen Team (2026b). We additionally evaluate GPT-5.6-Luna and Gemini-3.7-Flash as proprietary assistants. Every model is evaluated on all sixteen conditions with three independent runs.

Open models are served with vLLM Kwon et al. (2023) on two AMD MI210 64GB GPUs. Models up to 27B use one GPU, while the 30B and 32B models use tensor parallelism across both. The proprietary models are accessed through OpenRouter. All generations use temperature 1.0, top-$p = 1.0$, no top-k truncation, and at most 1024 output tokens. The context window is 65,536 tokens where supported and otherwise the largest available value, always at least 32,768. Across all thirteen assistants, the evaluation contains 187,200 conversations: 13 models $\times$ 16 conditions $\times$ 300 questions $\times$ 3 runs. The 158,400 conversations generated by the 11 open models are each graded by all 11 open-model judges, giving $1.74M$ cross-judgments.

5 Results: Composition, not Conversation

We separate *composition* from *conversation*: whether the scene is whole or decomposed, and whether information arrives in one turn or across several. The question and source scene remain fixed; only presentation and delivery change.

Loss in composition: ***A scene is more than the layers it contains.*** Replacing the whole image with all of its layers in the same turn drops accuracy by $-.187$ across all models. No visual evidence is removed, only the composition is lost. Doing the same to the question costs just $-.041$. Text survives fragmentation much better than vision. ***Memory alone does not explain this gap.*** Spreading either question fragments or image layers across turns adds a similar penalty, $-.103$ and $-.101$. In SNOWBALL, every turn repeats all previously seen layers, so the model no longer needs to remember them. Performance returns almost exactly to the all-layers-in-one-turn condition, but not to the whole-image condition. The model can hold all the

parts without recovering the scene. ***Recomposition does.*** In RECAP, the same sharded conversation ends by showing the whole image once. Yet accuracy rises by $+.343$ (95% CI $[+.315, +.373]$), with every model improving. The recovered score even exceeds showing the whole image only at the start. What matters is not simply whether the model has seen the evidence, but whether that evidence is available as a composed scene when it answers. ***Visual evidence also has to arrive when it is useful.*** The same layers score .410 when useful evidence arrives late, but only .265 when it arrives before the question. Visual evidence is therefore substantially less useful when it arrives before the question. The model does not reliably hold it in reserve until the question gives it meaning. Even where alignment should help, it does not: relational questions score .313 in both conditions, and aligned pairing underperforms misaligned for 9/11 models. As Table 1 shows, the pattern is consistent: local alignment does not recover the scene, question fragmentation is relatively cheap, visual fragmentation is costly, and combining both is worse. ***Visual evidence is not additive.*** Having all the parts is not equivalent to seeing the scene. Models reason best when visual evidence is composed and available at the moment it is needed.

ORACLE INVERSION: ***Perfect retrieval makes the model worse.*** Table 2 presents the same visual evidence in three ways. RGB shows the whole image. ALL-LAYERS shows the image together with every layer. ORACLE keeps only the annotated layers needed to answer the question. Oracle removes irrelevant content while preserving everything required for the answer. Instead, accuracy falls by $-.148$ relative to the whole image, and this happens for all models. Giving the model exactly the annotated evidence is worse than giving it the scene. ***The layer format does not explain the loss.*** ALL-LAYERS contains the same layer renders used by ORACLE, but keeps the original image beside them. Its change from RGB is only $-.017$. The important difference is what ORACLE removes. It keeps the evidence directly tied to the answer, but removes the rest of the scene. ***Relevant evidence is not sufficient evidence.*** The model does not reason from isolated answer-bearing regions in the same way that it reasons from a complete scene. Background, surrounding objects, and global composition can help interpret evidence that looks sufficient on its own. A retrieval step can therefore become destructive even when it retrieves the correct regions perfectly. Better visual retrieval is not only about finding the right evidence. It is also about preserving the context that makes that evidence meaningful.

Table 2: **What models can do when evidence is handed to them.**

	ANSWER ACCURACY			EVIDENCE SELECTION					ERROR MODES			JOINT
Model	RGB	ALL-LYR	ORACLE	Precision	Recall	F1-score	Ex Match	Comp.	Exc.	Dist.	Im.	Acc.
Q2.5-VL-3B	**.408** ±.02	.371 ±.03	.303 ±.01	.561 ±.01	.522 ±.01	.503 ±.01	.101 ±.01	.279 ±.00	.387 ±.02	.520 ±.02	.137 ±.01	.048 ±.01
Min-3-8B	**.463** ±.03	.420 ±.02	.267 ±.02	.617 ±.01	.565 ±.01	.562 ±.01	.174 ±.00	.295 ±.01	.318 ±.01	.409 ±.02	.159 ±.02	.080 ±.01
Q2.5-VL-7B	**.556** ±.02	.463 ±.02	.372 ±.01	.469 ±.01	.364 ±.02	.385 ±.02	.105 ±.01	.154 ±.02	.395 ±.02	.327 ±.02	.251 ±.01	.059 ±.01
Q2.5-VL-32B	**.559** ±.01	.629 ±.01	.474 ±.02	.519 ±.01	.437 ±.02	.447 ±.01	.126 ±.02	.205 ±.03	.455 ±.01	.314 ±.02	.413 ±.03	.083 ±.02
Pixtral-12B	**.572** ±.01	.478 ±.02	.396 ±.03	.649 ±.03	.494 ±.01	.531 ±.02	.168 ±.02	.230 ±.01	.324 ±.03	.383 ±.02	.092 ±.02	.087 ±.02
IVL3.5-8B	**.577** ±.01	.568 ±.00	.410 ±.00	.563 ±.02	.365 ±.01	.416 ±.01	.102 ±.02	.134 ±.02	.433 ±.02	.333 ±.02	.254 ±.02	.056 ±.02
Q3.5-9B	**.656** ±.01	.634 ±.02	.489 ±.03	.609 ±.02	.531 ±.01	.533 ±.01	.129 ±.01	.270 ±.01	.375 ±.01	.421 ±.03	.229 ±.00	.085 ±.01
GPT-5.6-L†	**.693** ±.01	.719 ±.01	.634 ±.01	.927 ±.01	.729 ±.02	.785 ±.02	.468 ±.02	.491 ±.02	.073 ±.01	.122 ±.01	.000 ±.00	.348 ±.01
Q3.6-27B	**.697** ±.02	.736 ±.02	.539 ±.02	.722 ±.01	.658 ±.00	.654 ±.00	.247 ±.01	.406 ±.01	.275 ±.01	.404 ±.02	.116 ±.01	.191 ±.00
Q3-VL-4B	**.703** ±.01	.693 ±.01	.572 ±.01	.424 ±.00	.317 ±.00	.346 ±.00	.091 ±.01	.113 ±.00	.569 ±.01	.279 ±.02	.522 ±.01	.060 ±.01
Q3-VL-30B	**.722** ±.00	.717 ±.01	.607 ±.02	.562 ±.00	.517 ±.01	.510 ±.00	.160 ±.02	.281 ±.02	.438 ±.00	.371 ±.01	.424 ±.01	.120 ±.01
Q3-VL-8B	**.729** ±.01	.744 ±.03	.580 ±.01	.563 ±.01	.501 ±.01	.505 ±.01	.149 ±.01	.254 ±.01	.430 ±.01	.414 ±.02	.298 ±.00	.124 ±.02
Gem-3.7-F†	**.862** ±.01	.883 ±.00	.833 ±.01	.971 ±.00	.769 ±.01	.826 ±.01	.521 ±.01	.549 ±.01	.024 ±.00	.043 ±.00	.000 ±.00	.474 ±.01
Mean, Open	**.604**	.587	.455	.569	.479	.490	.141	.238	.400	.380	.263	.090

LOSS IN GROUNDING: ***A correct answer does not mean the model used the right evidence.*** In ALL-LAYERS, models answer with .587 accuracy, but answer and grounding are jointly correct only .090 of the time. Their cited evidence is often not merely incomplete. Models over-cite layers, select distractors, and sometimes cite layers that do not exist. A model can therefore reach the right answer while giving a poor account of where that answer came from. ***Answering and grounding are different abilities.*** The model ranking changes sharply when evidence is scored. Qwen3-VL-4B performs well, but poor grounding, answers .693 of questions with an evidence F1 of .346, while Qwen3.6-27B is strong in both, answers .736 with an F1 of .654. Answer accuracy alone hides this difference. Two models can look equally capable while reasoning from the image very differently. ***The conversation reveals an even stronger failure.*** In the sharded

setting, we know exactly when each supporting layer becomes available. If the model answers before its gold evidence arrives, that answer cannot be grounded in that evidence. We call such cases *ungrounded correct*. Across the visual conditions, .393 of correct answers arrive before the required evidence. Requiring the evidence to be present before the answer reduces accuracy from .420 to .270. Much of what standard accuracy rewards is therefore not supported by the visual evidence available at the time. ***This changes how visual robustness should be read.*** A model that can answer without seeing the required evidence may appear robust when the image is fragmented, delayed, or degraded. Its score survives partly because its answer does not depend on the manipulated evidence. ***Answer accuracy measures whether the model is right. Grounding measures whether it is right for the image.***

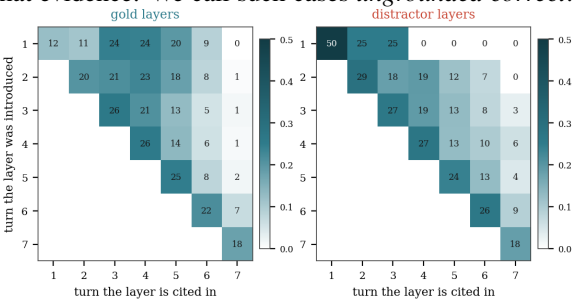

Figure 5: **Layer citation over time.**

HOW THE ANSWERS ARE PRODUCED: ***Answering early is worse than answering late.*** Accuracy by the point at which a model first commits runs .388, .458, .460, .473, .403 across fifths of the conversation, Table 4. Committing in the first fifth is the worst option, and all the models commit there most often. ***Longer replies are worse replies.*** Within each model's own length distribution, the longest fifth of responses is the least accurate, in both single-turn and multi-turn conditions, Table 5. The drop from shortest to longest is negative for all models and reaches $-.193$. ***Being right is not the same as being grounded.*** Scoring only the answers that arrived after their supporting layer takes the field from .420 to .270, Table 6. More than a third of correct answers were produced before the evidence for them existed. ***A layer is cited near the turn it arrives, and less often as the conversation moves on.*** Figure 5 plots where each layer is cited against the turn it arrived in. Two fifths of all citations of a layer fall in the turn that delivered it, and two turns later the share is under a fifth. A layer delivered in the sixth turn is cited in that turn 76% of the time, against 12% for one delivered first. There is simply no conversation left to carry it, which is the mechanism RECAP-V exploits. The scene is never new information; it is information the model was never given room to use. Appendices A.2, A.3, A.4, A.5, A.6, A.7 cover answer length, consistency, early commitment, verbosity, ungrounded answers, and judge self-preference.

6 CONCLUSION

We built Layered-VQA to separate two things that multi-turn evaluation normally changes at once. Composition is whether the model sees the scene whole or in pieces. Conversation is whether those pieces arrive in one turn or across many. Sixteen delivery conditions hold the question content and the source pixels fixed, which isolates the effect of presentation more closely than standard multi-turn benchmarks. Conversation is cheaper: splitting the question costs $-.041$, while splitting the scene costs $-.187$ even when all layers arrive in one turn. Recomposing the scene recovers $+.343$ without adding new pixels: the loss comes from fragmentation, not missing evidence. Two consequences follow, and neither is visible to a benchmark that shows one image and one question. Even oracle-selected sufficient layers trail the whole scene by $-.148$, showing that correct evidence selection can still hurt performance. Separately, a right answer is not a grounded one, since nearly three in ten correct answers arrive before the evidence supporting them does. Both failures concentrate on questions about relations between objects, which lose roughly three times more than questions about a single object. Fragmenting a scene keeps the objects and destroys what holds between them. VLMs are increasingly deployed on evidence that is cropped, retrieved, or revealed over time. Our results say that pipeline design carries a cost that answer accuracy alone will not report. Preserve the composed view, recompose before asking, and score grounding beside accuracy. ***The right evidence is not enough: VLMs need the scene it came from.***

ETHICS STATEMENT

The scenes come from LAION and LVIS through the MuLAN layer decomposition, all publicly released datasets. Many contain people, photographed in public settings by the original collectors. We add no new imagery, collect no personal data, and identify no one. The questions and reference answers were written by the authors, and the layer annotations were produced and checked by the authors. No crowd workers or external annotators were employed, so there are no labour concerns to report. The benchmark measures a failure mode rather than proposing a deployment. One result invites misreading and we state it plainly: an oracle that returns exactly the right evidence performs worse than the whole scene. That is an argument for keeping context, not an argument against retrieval. The proprietary models were accessed through a commercial API under its terms of service, and we report their behaviour without claiming access to their weights or training data.

REPRODUCIBILITY STATEMENT

The supplementary material contains the full code and the benchmark itself. The 16 delivery conditions are shipped as JSON, one file per condition, with the question shards, the layer schedules, the gold and minimal sufficient layer sets, and the distractor layers. Everything is included: the simulator, the user agents, the judges, the scoring, the annotation pipeline, and the scripts that produce every table and figure in this paper from the raw logs. The settings needed to repeat a run are in Section 4, and every prompt is reproduced in Appendix E. Each model runs three times under identical settings, and we report the run-to-run spread beside every number. Confidence intervals resample scenes rather than questions, since questions on one scene share an image. We will release the code, the benchmark, and the images publicly with the paper.

AI USE STATEMENT

We used an AI assistant while writing the code and while drafting parts of the text. The research itself was done by the authors: the idea, the experimental design, what each condition isolates, the analyses, the claims, and what the paper argues. The assistant wrote code alongside us and helped tighten prose after the structure and the content of a section were decided. Every line of code and every sentence in this paper was read and approved by the authors before it was included. We take full responsibility for the paper, including anything the assistant contributed. Vision-language models are also the object of study here rather than a tool. Their use as assistants, as simulated users, and as judges is described in Section 4.

REFERENCES

Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Baptiste Bout, Devendra Chaplot, Jessica Chudnovsky, Diogo Costa, Baudouin De Monicault, Saurabh Garg, Theophile Gervet, Soham Ghosh, Amélie Héliou, Paul Jacob, Albert Q. Jiang, Kartik Khandelwal, Timothée Lacroix, Guillaume Lample, Diego Las Casas, Thibaut Lavril, Teven Le Scao, Andy Lo, William Marshall, Louis Martin, Arthur Mensch, Pavankumar Muddireddy, Valera Nemychnikova, Marie Pellat, Patrick Von Platen, Nikhil Raghuraman, Baptiste Rozière, Alexandre Sablayrolles, Lucile Saulnier, Romain Sauvestre, Wendy Shang, Roman Soletskyi, Lawrence Stewart, Pierre Stock, Joachim Studnia, Sandeep Subramanian, Sagar Vaze, Thomas Wang, and Sophia Yang. Pixtral 12b, 2024. URL https://arxiv.org/abs/2410.07073.

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

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

Declan Campbell, Sunayana Rane, Tyler Giallanza, Nicolò De Sabbata, Kia Ghods, Amogh Joshi, Alexander Ku, Steven M Frankland, Thomas L Griffiths, Jonathan D Cohen, et al. Understanding the limits of vision language models through the lens of the binding problem. *Advances in Neural Information Processing Systems*, 37:113436–113460, 2024.

Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William Cohen. Murag: Multimodal retrieval-augmented generator for open question answering over images and text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5558–5570, 2022.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. Visual dialog. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 326–335, 2017.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. Eraser: A benchmark to evaluate rationalized nlp models. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4443–4458, 2020.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

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

Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14953–14962. IEEE, 2023.

Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality. *Advances in neural information processing systems*, 36:31096–31116, 2023.

Ziniu Hu, Ahmet Iscen, Chen Sun, Kai-Wei Chang, Yizhou Sun, David Ross, Cordelia Schmid, and Alireza Fathi. Avis: Autonomous visual information seeking with large language model agent. *Advances in Neural Information Processing Systems*, 36:867–878, 2023.

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

Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4198–4205, 2020.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2901–2910, 2017.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *2023 IEEE/CVF international conference on computer vision (ICCV)*, pp. 3992–4003. IEEE, 2023.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 881–905, 2024.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. Llms get lost in multi-turn conversation. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 54738–54778, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/59f6421e64707225fdf5b28840679a07-Paper-Conference.pdf.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, Baptiste Rozière, Baudouin De Monicault, Clémence Lanfranchi, Corentin Barreau, Cyprien Courtot, Daniele Grattarola, Darius Dabert, Diego de las Casas, Elliot Chane-Sane, Faruk Ahmed, Gabrielle Berrada, Gaëtan Ecrepont, Gauthier Guinet, Georgii Novikov, Guillaume Kunsch, Guillaume Lample, Guillaume Martin, Gunshi Gupta, Jan Ludziejewski, Jason Rute, Joachim Studnia, Jonas Amar, Joséphine Delas, Josselin Somerville Roberts, Karmesh Yadav, Khyathi Chandu, Kush Jain, Laurence Aitchison, Laurent Fainsin, Léonard Blier, Lingxiao Zhao, Louis Martin, Lucile Saulnier, Luyu Gao,

Maarten Buyl, Margaret Jennings, Marie Pellat, Mark Prins, Mathieu Poirée, Mathilde Guillaumin, Matthieu Dinot, Matthieu Futeral, Maxime Darrin, Maximilian Augustin, Mia Chiquier, Michel Schimpf, Nathan Grinsztajn, Neha Gupta, Nikhil Raghuraman, Olivier Bousquet, Olivier Duchenne, Patricia Wang, Patrick von Platen, Paul Jacob, Paul Wambergue, Paula Kurylowicz, Pavankumar Reddy Muddireddy, Philomène Chagniot, Pierre Stock, Pravesh Agrawal, Quentin Torroba, Romain Sauvestre, Roman Soletskyi, Rupert Menneer, Sagar Vaze, Samuel Barry, Sanchit Gandhi, Siddhant Waghjale, Siddharth Gandhi, Soham Ghosh, Srijan Mishra, Sumukh Aithal, Szymon Antoniak, Teven Le Scao, Théo Cachet, Theo Simon Sorg, Thibaut Lavril, Thiziri Nait Saada, Thomas Chabal, Thomas Foubert, Thomas Robert, Thomas Wang, Tim Lawson, Tom Bewley, Tom Bewley, Tom Edwards, Umar Jamil, Umberto Tomasini, Valeriia Nemychnikova, Van Phung, Vincent Maladière, Virgile Richard, Wassim Bouaziz, Wen-Ding Li, William Marshall, Xinghui Li, Xinyu Yang, Yassine El Ouahidi, Yihan Wang, Yunhao Tang, and Zaccharie Ramzi. Ministral 3, 2026. URL https://arxiv.org/abs/2601.08584.

Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, et al. Llava-plus: Learning to use tools for creating multimodal agents. In *European conference on computer vision*, pp. 126–142. Springer, 2024a.

Shuo Liu, Kaining Ying, Hao Zhang, Yue Yang, Yuqi Lin, Tianle Zhang, Chuanhao Li, Yu Qiao, Ping Luo, Wenqi Shao, et al. Convbench: A multi-turn conversation evaluation benchmark with hierarchical ablation capability for large vision-language models. *Advances in Neural Information Processing Systems*, 37:100734–100782, 2024b.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024c.

Ziyu Liu, Tao Chu, Yuhang Zang, Xilin Wei, Xiaoyi Dong, Pan Zhang, Zijian Liang, Yuanjun Xiong, Yu Qiao, Dahua Lin, et al. Mmdu: A multi-turn multi-image dialog understanding benchmark and instruction-tuning dataset for lvlms. *Advances in Neural Information Processing Systems*, 37:8698–8733, 2024d.

Ege Ozguroglu, Ruoshi Liu, Dídac Surís, Dian Chen, Achal Dave, Pavel Tokmakov, and Carl Vondrick. pix2gestalt: Amodal segmentation by synthesizing wholes. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3931–3940. IEEE Computer Society, 2024.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026a. URL https://qwen.ai/blog?id=qwen3.5.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, April 2026b. URL https://qwen.ai/blog?id=qwen3.6-27b.

Daniel Reich and Tanja Schultz. Uncovering the full potential of visual grounding methods in vqa. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4406–4419, 2024.

Dingjie Song, Shunian Chen, Guiming Hardy Chen, Fei Yu, Xiang Wan, and Benyou Wang. Milebench: Benchmarking mllms in long context. *arXiv preprint arXiv:2404.18532*, 2024.

Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11854–11864. IEEE, 2023.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5228–5238. IEEE, 2022.

Petru-Daniel Tudosiu, Yongxin Yang, Shifeng Zhang, Fei Chen, Steven McDonagh, Gerasimos Lampouras, Ignacio Iacobacci, and Sarah Parisot. Mulan: A multi layer annotated dataset for controllable text-to-image generation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22413–22422. IEEE, 2024.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965, 2023.

Hengyi Wang, Haizhou Shi, Shiwei Tan, Weiyi Qin, Wenyuan Wang, Tunyu Zhang, Akshay Nambi, Tanuja Ganu, and Hao Wang. Multimodal needle in a haystack: Benchmarking long-context capability of multimodal large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3221–3241, 2025a.

Weiyun Wang, Shuibo Zhang, Yiming Ren, Yuchen Duan, Tiantong Li, Shuo Liu, Mengkang Hu, Zhe Chen, Kaipeng Zhang, Lewei Lu, et al. Needle in a multimodal haystack. *Advances in Neural Information Processing Systems*, 37:20540–20565, 2024.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, Guanzhou Chen, Zichen Ding, Changyao Tian, Zhenyu Wu, Jingjing Xie, Zehao Li, Bowen Yang, Yuchen Duan, Xuehui Wang, Zhi Hou, Haoran Hao, Tianyi Zhang, Songze Li, Xiangyu Zhao, Haodong Duan, Nianchen Deng, Bin Fu, Yinan He, Yi Wang, Conghui He, Botian Shi, Junjun He, Yingtong Xiong, Han Lv, Lijun Wu, Wenqi Shao, Kaipeng Zhang, Huipeng Deng, Biqing Qi, Jiaye Ge, Qipeng Guo, Wenwei Zhang, Songyang Zhang, Maosong Cao, Junyao Lin, Kexian Tang, Jianfei Gao, Haian Huang, Yuzhe Gu, Chengqi Lyu, Huanze Tang, Rui Wang, Haijun Lv, Wanli Ouyang, Limin Wang, Min Dou, Xizhou Zhu, Tong Lu, Dahua Lin, Jifeng Dai, Weijie Su, Bowen Zhou, Kai Chen, Yu Qiao, Wenhai Wang, and Gen Luo. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency, 2025b. URL https://arxiv.org/abs/2508.18265.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13084–13094. IEEE, 2024.

Tsung-Han Wu, Giscard Biamby, Jerome Quenum, Ritwik Gupta, Joseph E Gonzalez, David Chan, et al. Visual haystacks: A vision-centric needle-in-a-haystack benchmark. In *International Conference on Learning Representations*, volume 2025, pp. 64592–64615, 2025.

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

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? *arXiv preprint arXiv:2210.01936*, 2022.

Lvmin Zhang and Maneesh Agrawala. Transparent image layer diffusion using latent transparency. *ACM Trans. Graph.*, 43(4), July 2024. ISSN 0730-0301. doi: 10.1145/3658150. URL https://doi.org/10.1145/3658150.

A Anatomy of the loss

A.1 Where the loss falls

Table 3 splits every headline condition by question type. It is the reason the paper claims a mechanism rather than a list of drops. Questions that ask about one object on its own lose −.081 under ORACLE and −.126 under CONCAT-V. Questions that ask about a relation between objects lose −.228 and −.259. The two losses fall on the same questions, in the same order, and they are roughly three times larger on relational questions than on single-object ones. An oracle crop and a layer decomposition are both ways of fragmenting a scene, and fragmenting a scene keeps the objects and destroys what holds between them. Rescheduling the question does not do this, which is why CONCAT-Q costs about −.03 to −.05 regardless of type.

Table 3: **Accuracy by question type and delivery condition.**

Question type	n	FULL	CONCAT-Q	SHARDED-Q	CONCAT-V	SHARDED-V	RECAP-V	ALL-LYR	ORACLE
Direct ask	83	**.663**	.632	.513	.538	.376	.693	.650	.573
Attribute binding	65	**.601**	.565	.434	.507	.398	.682	.604	.549
Occlusion	17	**.594**	.590	.412	.346	.248	.624	.560	.443
Counting	44	**.642**	.539	.492	.298	.235	.641	.604	.342
Spatial	54	**.524**	.513	.527	.280	.252	.658	.496	.305
Multi-hop	37	**.550**	.495	.293	.372	.282	.615	.546	.394
Single-object	165	**.632**	-.030	-.160	-.126	-.260	+.050	-.009	-.081
Relational	135	**.570**	-.053	-.118	-.259	-.315	+.071	-.025	-.228

A.2 Response length across attempts

Figure 6 plots the length of each successive answer attempt against how often that attempt was correct. Lost-in-Conversation Laban et al. (2026) reports that answers grow as a conversation drags on. Ours shrink. In the text conditions the first attempt averages 86 characters and the tenth averages 57. The visual and layered conditions do the same. The bloat appears immediately. The same question answered in a single turn gets 18 characters. Every multi-turn first attempt is three to five times longer than that. A model given part of a question pads immediately rather than gradually. Accuracy of an attempt peaks mid-conversation. In the text conditions it climbs from .34 on the first attempt to .51 on the sixth, then falls to .26 by the tenth. A model still revising after six attempts is not converging on an answer, it is drifting away from one.

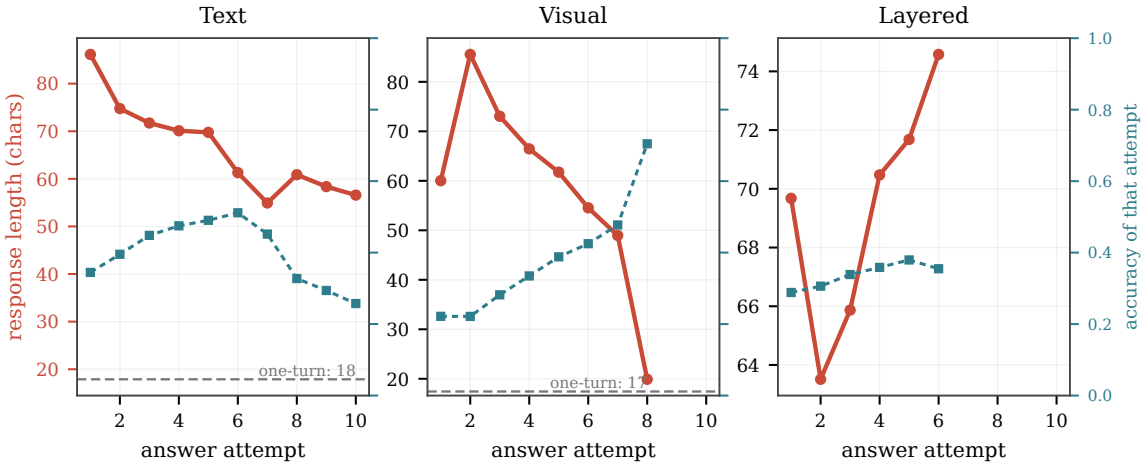

Figure 6: **Response length and accuracy across successive answer attempts.** The dashed line is the single-turn baseline for the same questions. Attempts get terser, and accuracy peaks mid-conversation.

A.3 APTITUDE AND UNRELIABILITY

Figure 7 separates how well a model does from how consistently it does it. Aptitude is accuracy. Unreliability is the share of questions the model decides both ways across the three replicate runs, on the same question under identical settings. Three exact replicates make this a direct measurement rather than a spread over resampled simulations. Breaking the question into shards costs .142 of accuracy and adds .147 of unreliability, taking the field from .290 to .437. Ten of the eleven models get less reliable. Breaking the scene into layers costs .099 and adds .076, and nine of eleven get less reliable. Almost every arrow points left and up. No model trades accuracy for consistency. Sharding does not make a model cautious, it makes it inconsistent.

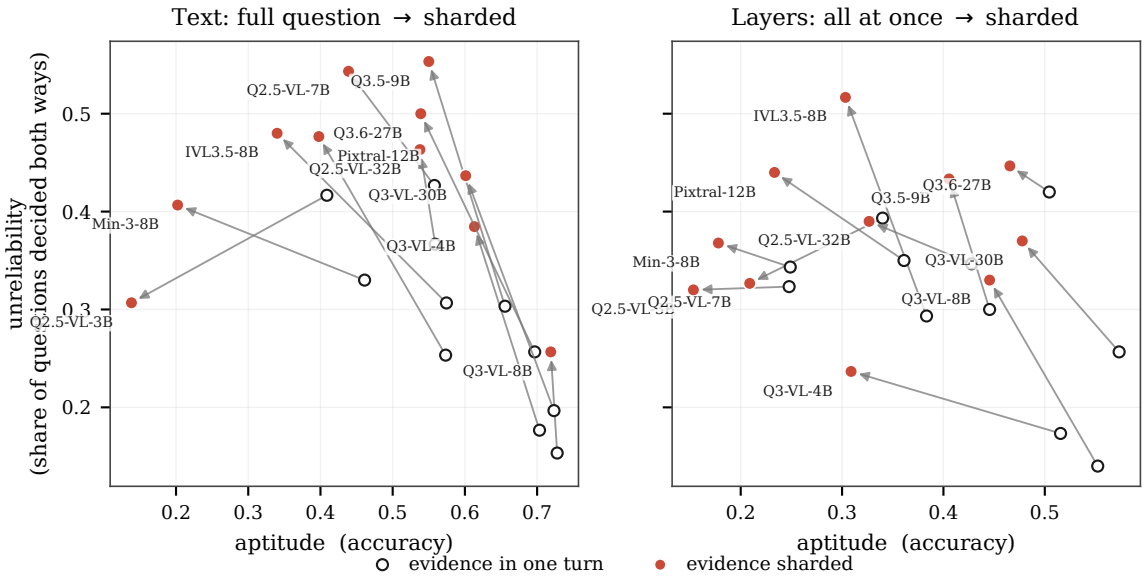

Figure 7: **Aptitude against unreliability.** Hollow markers deliver the evidence in one turn, filled markers shard it. An arrow joins the two states of one model.

A.4 PREMATURE COMMITMENT

Table 4 bins every conversation by how far through it the model first committed to an answer, then reports accuracy in each fifth. ***The best moment to answer is not the last one.*** Accuracy runs .388, .458, .460, .473, .403. It rises, peaks in the fourth fifth, and then falls again. The text-only result this parallels is monotone: waiting longer is always better there. Ours is not, and the difference is the point. A text conversation withholds constraints, so a later answer is simply a better informed one. Ours withholds a scene, and a scene has to be assembled. A model that has still not committed at four fifths is usually one that could not assemble it, working from a transcript full of its own earlier hedging. ***Models commit before the evidence justifies it.*** 62% of first commitments land in the first two fifths, and the modal bin is the second fifth for ten of the eleven open models. The first fifth is also the least accurate bin for nine of them. Models are answering where they do worst, and they are doing it because our instructions ask every turn for either an answer or one clarifying question. That is the same contract a deployed assistant operates under. Nothing in the prompt says an answer is due, and they give one anyway.

What the ideal looks like. A model should hold a hypothesis without stating it and commit when the last supporting layer lands, not before and not long after. It should be able to say that it has seen a constraint and is still missing the evidence for it. ***What would get us there.*** Make abstention cheap. Instruction tuning rewards a helpful answer on every turn, which is exactly the behaviour that costs accuracy here. A model that could emit an explicit *evidence not yet sufficient* state, and not be penalised for it, would move its commitments out of the worst bin. This bin is chosen by the model rather than assigned by us, so the table is observational. The intervention that would make it causal is to force commitment at a fixed turn, which we leave to future work.

Table 4: **When models first attempt an answer.**

Model	FIRST ANSWER ATTEMPT					COMMITS
	earliest 0–20%	early 20–40%	midway 40–60%	late 60–80%	latest 80–100%	share
Q2.5-VL-3B	.206	.220	.232	.261	.203	51%
Q3-VL-4B	.417	.561	.545	.514	.500	63%
Q2.5-VL-7B	.332	.393	.398	.390	.333	53%
Q3-VL-8B	.502	.626	.657	.646	.574	45%
IVL3.5-8B	.357	.442	.440	.405	.410	45%
Min-3-8B	.200	.250	.267	.302	.224	47%
Q3.5-9B	.439	.530	.530	.548	.603	58%
Pixtral-12B	.341	.337	.388	.384	.326	41%
Q3.6-27B	.498	.577	.550	.556	.508	44%
Q3-VL-30B	.521	.598	.626	.608	.549	54%
Q2.5-VL-32B	.188	.457	.504	.533	.378	42%
GPT-5.6-L†	.587	.622	.655	.645	.655	36%
Gemini-3.7-F†	.865	.829	.828	.837	.857	34%
Mean, Open	.388	.458	.460	.473	.403	

A.5 VERBOSITY

Table 5 sorts each model's responses into length quintiles computed within that model, so a bin compares a model against itself rather than against a chattier one. ***Length is only bad once the scene is gone.*** When the evidence arrives in one turn, accuracy runs .558, .577, .589, .544, .408. It rises before it falls, and the median-length fifth is the best. When the same evidence arrives across turns, it runs .493, .484, .465, .447, .410, falling from the first bin onward. The prevailing advice from text-only work, that shorter responses are better, holds only in the second case. In the first, some length is reasoning and it pays. ***Why the peak disappears.*** Length in a single turn buys the model something to do: it can describe the scene, bind objects, and check a relation. Length across turns buys nothing, because the scene it would reason over has not been assembled. What remains is hedging, and hedging starts on the first reply rather than growing into it. The drop from shortest to longest is negative for all eleven models, reaching $-.193$ across turns and $-.297$ within one. ***What the ideal looks like.*** Response length should track how hard the question is, not how the evidence was delivered. The same question should not produce a longer answer because the picture arrived in pieces. ***What would get us there.*** Do not optimise length globally. A brevity penalty applied everywhere would remove the reasoning that helps in the single-turn case along with the hedging that hurts in the multi-turn one. Length is more useful as a signal than as a target: a reply that runs long under fragmentation, against the same model's own baseline on the same question, is a cheap runtime indicator that the model is under-evidenced and should be handed the composed scene.

A.6 UNGROUNDED CORRECTNESS

Table 6 asks, of the answers we score as correct, how many were already given before the layer supporting them had been delivered. It covers only the seven conditions where the question is defined: multi-turn, and carrying a layer schedule. ***Answer accuracy is inflated by a prior.*** Across those conditions accuracy is .401, and .287 once ungrounded answers are removed. Nearly three in ten correct answers were produced before their evidence existed. The model is not reading the image at that moment. It is answering from what scenes of this kind usually contain, and our scenes are natural enough that this often works. ***Pairing evidence with the constraint that needs it helps.*** The share is .352 where layers arrive on a fixed schedule unrelated to the question, .282 where the order of the layers is manipulated, and .153 where every turn carries both a piece of the question and a layer. Only the last of these puts an image in front of the model at the moment a constraint is stated. It does not raise accuracy much, and it changes what accuracy means.

Table 5: **Verbosity: Accuracy by evidence length and delivery mode.** Models are grouped by the length of the evidence they receive, either in one turn or spread across multiple turns.

	EVIDENCE IN ONE TURN					EVIDENCE ACROSS TURNS				
Model	SHORTEST	SHORT	MEDIAN	LONG	LONGEST	SHORTEST	SHORT	MEDIAN	LONG	LONGEST
Q2.5-VL-3B	.311	.365	.349	.369	.253	.270	.249	.227	.236	.218
Q3-VL-4B	.649	.706	.715	.647	.451	.592	.569	.564	.534	.474
Q2.5-VL-7B	.478	.565	.554	.403	.336	.412	.412	.415	.381	.327
Q3-VL-8B	.696	.676	.732	.694	.521	.649	.657	.623	.622	.557
IVL3.5-8B	.524	.468	.512	.546	.407	.458	.431	.385	.426	.448
Min-3-8B	.526	.519	.446	.276	.229	.297	.291	.300	.242	.224
Q3.5-9B	.617	.576	.600	.631	.479	.563	.562	.531	.506	.462
Pixtral-12B	.453	.547	.510	.575	.396	.401	.383	.379	.375	.341
Q3.6-27B	.639	.625	.704	.710	.526	.609	.598	.578	.548	.499
Q3-VL-30B	.667	.678	.764	.651	.539	.584	.644	.624	.577	.566
Q2.5-VL-32B	.579	.618	.590	.485	.347	.589	.526	.488	.467	.396
GPT-5.6-L†	.758	.739	.629	.649	.622	.697	.684	.644	.624	.575
Gemini-3.7-F†	.815	.875	.873	.898	.829	.858	.864	.828	.826	.816
Mean, Open	.558	.577	.589	.544	.408	.493	.484	.465	.447	.410

The weakest models guess the most. Ungrounded share tracks capability inversely, from .421 for the smallest model down to .202 for the strongest open one. It does not vanish with scale. Gemini-3.7-Flash answers correctly .830 of the time and grounded .588 of the time, the largest absolute grounding gap in the study, because a model that is right more often has more correct answers available to be ungrounded. ***What the ideal looks like.*** Scored accuracy and grounded accuracy should coincide. A model should decline to answer until the layer carrying the answer has arrived, and should be able to name it. ***What would get us there.*** Report the gap. It is a hallucination rate measured on a benchmark where the ground truth for *which evidence was required* is annotated, which is rare, and it costs nothing beyond what a normal evaluation already computes. Training against it is the harder half: a model rewarded only for the final answer has no reason to wait, since a lucky prior scores the same as a grounded reading. A citation requirement of the kind ALL-LAYERS imposes turns that into something optimisable.

Table 6: **Ungrounded answers remain common across delivery settings.** ***Left***: reports the share of responses produced before the required evidence was available; ***Right***: compares standard accuracy with accuracy restricted to grounded answers.

	UNGROUNDED SHARE			ACCURACY	
Model	Visual	Layered	Order	as scored	grounded only
Q2.5-VL-3B	.603	.556	.606	.223	**.090**
Q3-VL-4B	.397	.369	.274	.482	**.312**
Q2.5-VL-7B	.454	.404	.376	.338	**.196**
Q3-VL-8B	.326	.291	.221	.573	**.411**
IVL3.5-8B	.478	.260	.377	.403	**.243**
Min-3-8B	.344	.443	.269	.270	**.181**
Q3.5-9B	.362	.413	.257	.473	**.315**
Pixtral-12B	.443	.445	.398	.339	**.193**
Q3.6-27B	.399	.382	.306	.524	**.333**
Q3-VL-30B	.443	.326	.330	.556	**.344**
Q2.5-VL-32B	.192	.325	.093	.434	**.354**
GPT-5.6-L†	.347	.371	.198	.630	**.440**
Gemini-3.7-F†	.273	.369	.116	.835	**.635**
Mean, Open	.393	.362	.298	.420	**.270**

A.7 SELF-PREFERENCE AMONG JUDGES

In the primary runs each model grades its own conversations. That is convenient and it is also a conflict of interest. To measure it we re-graded every conversation with all eleven models and built the 11×11 matrix in Figure 4. A raw cell mixes three things: the assistant was good, the judge was generous, or the judge favoured that assistant. Only the third is interesting, so we remove the other two. For each cell we subtract

the assistant's mean score across all judges and the judge's mean score across all assistants, then add the grand mean back. What is left is the residual, and it is what the figure plots. ***Judges disagree more than models do.*** Leniency runs from $.404$ to $.720$ across judges, a spread of $.32$. Ability runs from $.334$ to $.629$ across assistants, a spread of $.30$. Picking a judge therefore moves a score about as much as picking a model. A benchmark graded by one judge is reporting that judge as much as the field. ***Every model favours itself.*** All eleven diagonal residuals are positive. The average model gains 2.6 points from grading its own work, and off-diagonal residuals average zero. Nothing else in the matrix has a sign that holds eleven times out of eleven. ***The bias is largest where it does the most damage.*** Qwen2.5-VL-3B gains 12.9 points on itself, the largest value in the matrix, and it is the weakest model in the study. Ministral-3-8B gains 0.5 points, the smallest. Self-grading does not add noise evenly. It lifts the bottom of the ranking, which is exactly where a benchmark needs to discriminate. This is why every number for the open models is a consensus verdict rather than a self-graded one. The self-graded results are only for the closed-models. A model still grades itself in one cell of eleven, and ten other judges outvote it.

A.8 Human validation of the consensus judge

Every open model accuracy is a consensus verdict, so the consensus itself needs a check against people. We drew 400 conversations, 25 from each of the sixteen conditions, cycling through the eleven models inside each condition so that every model is represented. The draw is not tilted towards hard cases. It does not need to be: the eleven judges agree unanimously on only 27.8% of conversations, so disagreement is already the common case. Two annotators labelled every conversation independently. Each saw the question, the reference answer, and the full conversation with its images, but never the judges' votes. The instruction was a single decision: does the model's final answer mean the same as the reference answer? A partial answer counts as wrong, an answer the model later withdrew does not count, and a conversation with no answer is wrong. The annotators agreed on 97.5% of conversations ($\kappa = 0.95$), and the remaining cases were settled by discussion. ***The consensus agrees with people.*** Against the adjudicated human verdict, the consensus is correct on 96.0% of conversations ($\kappa = 0.92$). The best single judge reaches 92.5%. ***Voting is what earns the gap.*** A single judge carries its own leniency and its own preference for itself, Appendix A.7. A majority of eleven cancels most of both. Agreement is 99.1% on conversations where the judges were unanimous and 94.8% where they split, so the remaining errors sit where the judges themselves were unsure.

B Ruling out the alternatives

The scene loss has three obvious rivals. It could be missing context, because a layer shows less of the world than the composite. It could be distribution shift, because a lone object on a blank canvas is a strange input. It could be an artifact of how finely we cut a scene. Each subsection takes one and tests it against the data. The last asks whether the relational gap survives uncertainty measured at the scene level.

B.1 Is the oracle loss about missing context, or missing composition?

ORACLE removes two things at once: the surrounding scene, and the composed view. ALL-LAYERS keeps the composite and adds every layer beside it. It costs $-.017$. ORACLE takes the composite away and keeps only the gold layers. It costs $-.148$. Adding layers is nearly free when the composite is there, so the loss tracks the composite, not the layer count. The second comparison uses the annotation. For 97 questions the background is part of the gold set, so ORACLE delivers scene context. For the other 203 it does not. Context does not rescue the model. Questions whose oracle includes the background lose $-.171$, and questions whose oracle excludes it lose $-.136$. These are different questions, so this is a split rather than a control, but it points the wrong way for the context explanation. What ORACLE destroys is the arrangement. The layers preserve every object at its original position, and the model still cannot read the relations between them.

B.2 Is the loss just distribution shift from layer images?

A layer is an unusual input. It is one object on a blank canvas, and models rarely see that in training. If the loss were distribution shift, then feeding layers at all would hurt. ALL-LAYERS feeds exactly the same layer renders, in the same number, with the composite alongside. It costs $-.017$ against $-.187$ for the same layers without the composite. The renders are therefore not the problem. Whether the composed view is present is.

B.3 Does the result depend on how finely a scene is decomposed?

Our scenes decompose into 2 to 6 layers, so the benchmark contains its own sensitivity analysis. FULL is flat at about .60 across depth, so deeper scenes are not harder scenes. Both losses grow steadily with depth, from $-.050$ to $-.269$. The effect is not an artifact of one decomposition setting. It is a dose response: the more pieces a scene is cut into, the more of it is lost.

Table 7: **Loss by scene depth.** Both losses grow with the number of layers, and neither depends on a particular granularity.

Layers	Questions	FULL	Δ CONCAT-V	Δ ORACLE
2	11	.606	$-.050$	$-.003$
3	100	.597	$-.149$	$-.128$
4	110	.611	$-.193$	$-.155$
5	51	.594	$-.224$	$-.156$
6	28	.616	$-.269$	$-.229$

B.4 Are the relational losses significant at the scene level?

Both headline losses are about three times larger on questions about relations between objects than on questions about one object. We test the difference directly, with a scene-level cluster bootstrap over 2000 draws. Each draw resamples the 93 scenes, keeps all of a scene's questions, and recomputes both losses and their difference inside the replicate. Both intervals exclude zero. Fragmenting a scene and cropping to an oracle are different operations, and they cost the same kind of question the same extra amount. That is the mechanism: both keep the objects and destroy what holds between them.

Table 8: **Relational questions lose more under visual decomposition.** Extra loss is the difference between relational and single-object degradation.

Condition	Relational	Single-object	Extra loss	95% CI
CONCAT-V	$-.259$	$-.126$	$-.133$	$[-.189, -.074]$
ORACLE	$-.228$	$-.081$	$-.147$	$[-.202, -.089]$

B.5 Where the loss falls, and what putting it back buys

Two questions the headline averages cannot answer. Is the loss spread across the benchmark or concentrated in part of it? And when recomposition recovers accuracy, what exactly is it recovering?

A minority of questions carries the loss. Figure 8 ranks questions by how much they lose and plots the running total. An even loss would trace the diagonal. The measured curve does not come close. The worst tenth of questions carries 34% of the total loss from fragmenting the scene, the worst fifth carries 57%, and 78 of the 300 questions lose nothing at all. The oracle is more concentrated still, at 38%. Fragmentation does not make every question slightly harder. It destroys some questions and leaves the rest alone, which is what a compositional bottleneck looks like and not what uniform information loss looks like. ***Each extra***

layer costs more. The right panel splits the same two losses by how finely the scene decomposes. FULL is flat across depth at about .60, so deeper scenes are not harder scenes. Both losses grow monotonically, from $-.050$ at two layers to $-.269$ at six. The cost is not a fixed penalty for using layers at all. It scales with how much of the scene was taken apart.

Recomposition pays twice. Figure 9 decomposes what a final restoring turn is worth. Fragmenting the question costs $-.141$ and restoring it recovers $+.209$. Fragmenting the scene costs twice as much at $-.285$, and restoring it recovers $+.344$. Both end above the FULL baseline, and by almost the same margin: $+.068$ and $+.059$. The right panel shows the two overshoots against each other, one point per model, sitting on the diagonal. So the recovery has two parts. A final turn that restates everything is worth about $+.06$ whatever was fragmented, which is the value of a clean summary at the end of a conversation. The rest is undoing the fragmentation, and that part is twice as large for the scene as for the question. This also settles a smaller puzzle: RECAP-V scoring above FULL is not a quirk of the visual stream, since RECAP-Q does the same thing by the same amount.

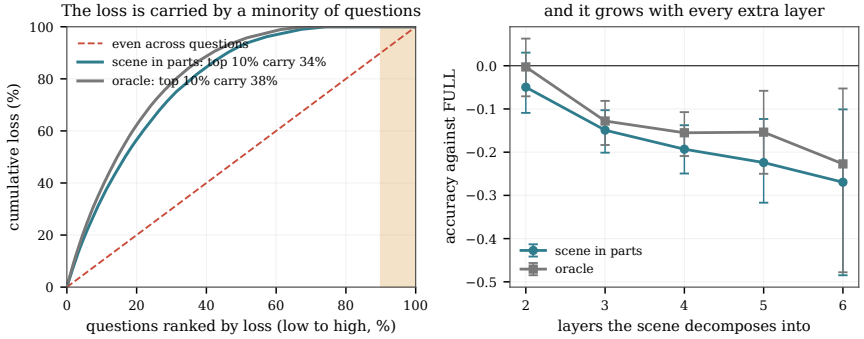

Figure 8: **The loss is concentrated, and it scales with depth.** *Left:* questions ranked by loss against the running total of that loss. *Right:* both losses by the number of layers a scene decomposes into, with 95% scene-level bootstrap intervals.

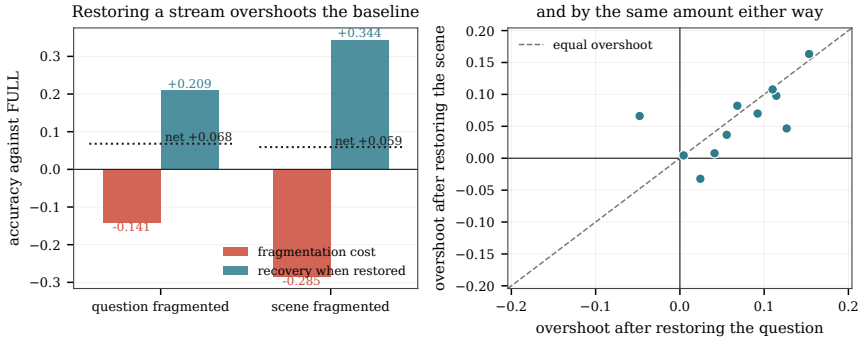

Figure 9: **Restoring a fragmented stream overshoots the baseline.** *Left:* the cost of fragmenting each stream and the recovery when it is restored, against FULL. *Right:* the two overshoots per model. Points on the diagonal mean the final turn is worth the same either way.

C ONE QUESTION, END TO END

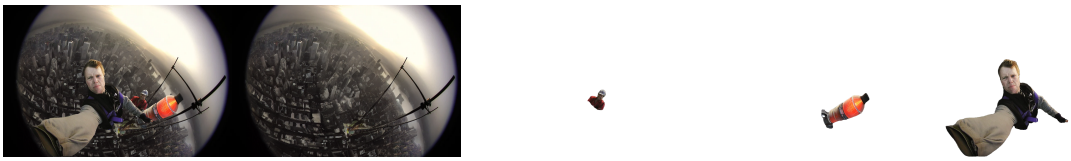

Figure 10: **The scene and its four layers.** Left to right: the composed scene, then layer 0 (background, not needed), layer 1 (the worker, the reference landmark), layer 2 (the bottle, the answer), and layer 3 (the man taking a selfie, the relational bridge). Every layer render is used exactly as shown.

Q&A. *What object is positioned to the right of the man taking a selfie and directly in front of the worker wearing a white helmet and an orange-red shirt on top of the skyscraper? – a red bottle with a black cap.*

The question in parts. The sharded conditions deliver the question as six pieces, one per turn. The first piece states the task and carries no content. Each later piece adds one constraint, and two of them exist only to disambiguate an earlier one. Shard 3 is the one that makes this question hard. It refers to *the man*, and the scene contains two men. The disambiguation arrives one turn later, in shard 4. A model that commits after shard 3 is guessing which man was meant, and Section 5 shows that models commit early far more often than they should. The layers in Figure 10, and shards in Table 9.

Table 9: **The six question shards, and the layers each one depends on.** A shard with no supporting layer is asking about the question rather than the image.

Shard	Text	Supported by
1	can you help me identify a particular thing in this image?	—
2	The answer should name an object.	2
3	The target object is positioned to the right of the man.	2, 3
4	The referenced man is taking a selfie.	3
5	The target object is directly in front of the worker.	1, 2
6	The referenced worker is wearing a white helmet and an orange-red shirt and is on top of the skyscraper.	1

D THE SAME QUESTION IN EVERY CONDITION

Section C introduced one question, its scene and its four layers. Now, that question through all sixteen conditions, Table 10. Nothing about the question or the image changes. Only the schedule does. Read Q as the whole question, q_k as its k-th shard, I as the composed scene, and L_k as layer k. The first six schedules break the question and keep the scene whole. The next four break the scene and keep the question whole. The next four vary the order or bind the two streams together. The last two vary the evidence rather than the schedule. CONCAT-V against FULL holds the pixels and the turn count fixed and takes away only the composition. RECAP-V against SHARDED-V adds nothing the model has not already seen and puts the composition back. ORACLE against FULL gives the model exactly the three layers it needs and takes away everything else. Note what ALIGNED-VQ and MISALIGNED-VQ do to shard 3, the ambiguous one. Under alignment it arrives with layer 3, the man it refers to. Under misalignment it arrives with layer 1, the other man. The pairing is the only difference between the two conditions, and the models score the same under both.

Table 10: **Sixteen delivery schedules for one question.**

Condition	Turns	What the model receives
The question arrives in parts, the scene stays whole		
FULL	1	Q with I
CONCAT-Q	1	$q_1 q_2 q_3 q_4 q_5 q_6$ with I
SHARDED-Q	6	q_1 with I; then q_2, q_3, q_4, q_5, q_6, one per turn
RECAP-Q	7	As SHARDED-Q, then a final turn repeating Q with I
SNOWBALL-Q	6	q_1 with I; then $q_1 q_2$; then $q_1 q_2 q_3$; and so on
SHUFFLED-CONCAT	1	$q_1 q_6 q_4 q_3 q_2 q_5$ with I
The scene arrives in parts, the question stays whole		
CONCAT-V	1	Q with L_0 L_1 L_2 L_3, no I
SHARDED-V	5	Q; then L_0, L_1, L_2, L_3, one per turn, no I
RECAP-V	6	As SHARDED-V, then a final turn carrying I
SNOWBALL-V	5	Q with L_0; then $L_0 L_1$; then $L_0 L_1 L_2$; and so on
Order, and binding the two streams		
EVIDENCE-FIRST	5	Q; then L_3, L_1, L_2, then the unneeded L_0 last
EVIDENCE-LAST	5	Q; then L_0 first, then L_3, L_1, L_2
ALIGNED-VQ	6	q_2 with L_2; q_3 with L_3; q_5 with L_1; other shards alone
MISALIGNED-VQ	6	q_1 with L_2; q_2 with L_3; q_3 with L_1; other shards alone
Changing the evidence		
ALL-LAYERS	1	Q with I and all four layers $L_0 L_1 L_2 L_3$, plus a request to name the layers used
ORACLE	1	Q with only the three gold layers $L_3 L_1 L_2$, and no I

Table 11: **Scene depth across the benchmark.** One scene per depth, from a single object over a background up to five. Depth is a property of the scene and shard count is a property of the question.

2
1 + bg







BG
background

L1
barn

DIRECT ASK — 2 layers — 5 shards

What natural landscape is visible rising behind the barns in the distance?

Answer **A mountain range.**

1 can you help me figure out what is being identified in this image?
2 The answer should name a natural landscape. BG
3 The target is the natural landscape that is visible and rising.
4 It is behind the barns.
5 The barns are in the distance.

3
2 + bg









BG
background

L1
bird

L2
bird

SPATIAL — 3 layers — 6 shards

Where is the brown hawk positioned relative to the white bird with the yellow-and-black beak on the rocky shoreline?

Answer **The brown hawk is to the left of the white bird.**

1 can you help me figure out a spatial property of a particular thing?
2 The answer should report where the thing is located.
3 The target thing is the brown hawk. L2
4 Determine the brown hawk's position relative to a comparison entity. L1
5 Use the white bird with the yellow-and-black beak as the comparison entity.
6 The white bird with the yellow-and-black beak is on the rocky shoreline. BG

Table 11, continued from the previous page.

4
3 + bg





BG background | L1 dog | L2 dog | L3 dog

COUNTING — 4 layers — 6 shards

How many dogs in the fenced dirt area are facing generally toward the camera or frontward, and how many are turned backward or away?

Answer **Two dogs are facing frontward, and one dog is facing backward.**

1 can you help me figure out the counts for two groups of subjects? L1 L2 L3
2 Each requested result is a count.
3 The subjects to count are dogs.
4 The dogs are in the fenced dirt area. BG
5 Count the dogs that are facing generally toward the camera or frontward.
6 Include a count of the dogs that are turned backward or away.

5
4 + bg





BG background | L1 symbol | L2 woman | L3 toy | L4 man

MULTI-HOP REFERENCE — 5 layers — 5 shards

What object is being held by the person standing to the right of the man who is pointing toward her?

Answer **A teddy bear.**

1 can you help me identify a particular thing in this image?
2 The answer should name an object. L3
3 The target object is being held by a person. L2
4 The person holding the target object is standing to the right of a man. L4
5 The referenced man is pointing toward the person holding the target object.

6
5 + bg





BG background | L1 lion | L2 lion | L3 lion | L4 lion | L5 lion

COUNTING — 6 layers — 5 shards

How many lions are running across the grassy field in this scene?

Answer **There are five lions visible in the field…**

1 can you help me figure out a quantity for particular animals? L1 L2 L3 L4 L5
2 The requested quantity is a count.
3 The animals to count are lions.
4 Count lions that are running across the grassy field. BG
5 Restrict the count to the lions in this scene.

E PROMPTS

Placeholders in double brackets are filled at run time. `[[IMAGE_MANIFEST]]` becomes one labelled line per image attached to that turn, and a layer is always named by its own number rather than by its position in the list.

E.1 QUESTION SHARDING PROMPT

The first of the three annotation calls. It sees only the canonical question, never the image, so it cannot leak the answer or add a visual fact. A per-category file is appended to it with rules specific to *direct ask*, *attribute binding*, *counting*, *spatial*, *occlusion*, and *multi-hop reference*.

```
You are constructing controlled multi-turn question shards for a
visual-question-answering benchmark.

Transform only the supplied canonical question. You cannot see the image. Do
not answer the question, infer the answer, use world knowledge to add image
facts, or make a vague phrase more specific than the source permits. You may
minimally repair grammar so each shard reads naturally, but the repair must
preserve the source meaning exactly.

Produce atomic_units first, then 3-6 conversational shards. Follow these
requirements:

1. Information preservation: all shards together must select the same answer as
   the canonical question. Preserve every answer-relevant constraint and no
   extra visual fact.
2. Clear initial intent: shard 1 must have role "intent" and state the requested
   operation while withholding the answer type, target, or at least one
   answer-selecting constraint so that the turn is genuinely underspecified.
3. Order independence: after shard 1, each shard must be understandable as an
   additional constraint without relying on the order of other follow-up shards.
   Repeat a compact noun phrase when needed; avoid dangling "it", "they", "that
   one", or "the other one" references.
4. Atomicity: each follow-up shard should introduce one meaningful constraint.
   Split compound spatial or referential chains when each part independently
   narrows the target.
5. Minimal transformation: stay close to the canonical wording. Generic
   conversational framing such as "the target object" or "in the image" is
   allowed; new colors, identities, counts, locations, relations, or object
   types are not.
6. No answer leakage: never state or imply the answer. A shard describes what to
   find, not what the result is.
7. Required granularity: always produce 3-6 shards. For a short question, use
   exactly three distinct functions: (1) the generic operation or request,
   (2) the expected answer type or property, and (3) the target or scope. For
   example, split "What color is the bulb?" into the intent to determine a
   property, the fact that the requested property is color, and the bulb as
   target. For a longer question, add separate answer-selecting attributes,
   relations, references, scopes, or conditions up to six shards. Do not
   fabricate constraints or repeat a shard merely to reach three.
8. Exact provenance: every atomic unit's source_span must be a contiguous
   excerpt of the canonical question. Intent and answer type may use separate
   subspans of an interrogative phrase, such as "What" and "color" or "How" and
   "many". Number atomic units and shards consecutively from 1. Every atomic
   unit must be referenced by at least one shard.
9. Natural delivery: write each shard as a normal conversational instruction or
   clarification. Never mention the "canonical question," "source phrase,"
   atomic units, annotations, shards, or benchmark machinery in shard text.

Set information_preserved to "yes" and added_visual_facts to "no" only after
checking the final plan against the canonical question.
```

E.2 EVIDENCE ANNOTATION PROMPT

The second annotation call. Unlike the sharding call, it sees the composite, every layer, and the reference answer. A per-category file is appended with rules for each of the six question types.

```
You are annotating visual evidence for a layered visual-question-answering
benchmark. You receive a full composite image, a numbered background layer (0),
numbered isolated object layers, human-reviewed captions, a canonical question
and answer, and question shards.

The full composite gives global context, but it is not a selectable evidence ID.
Select only IDs listed in candidate_layer_ids. Layer 0 is selectable when
background or scene context is genuinely necessary. Isolated RGBA layers are
shown on a synthetic gray checkerboard; the checkerboard is never scene
evidence.

Use pixels and human-reviewed captions as the primary evidence. Auxiliary labels
and geometry relations can help locate a layer, but labels may be noisy and
cannot override clear pixels or reviewed captions.

Apply these rules strictly:

1. Minimal sufficiency is causal, not merely topical. A sufficient set is the
   smallest set of retained numbered layers that lets a careful viewer both
   establish the question's complete answer-selecting reference path and justify
   the human gold answer. The full composite may help you adjudicate the
   annotation, but it cannot substitute for a missing numbered target, owner,
   landmark, bridge, or occluder in the selected set. Remove any member that is
   not necessary for either reference resolution or answer extraction.
2. gold_layer_ids is the canonical primary minimal sufficient set, in reasoning
   order when an order matters. It must exactly equal the first entry of
   minimal_sufficient_layer_sets.
3. Add another minimal set only when it is a genuinely different, independently
   sufficient, inclusion-minimal alternative. Do not list supersets or cosmetic
   variations.
4. layer_roles must contain exactly one assignment record for each layer in the
   union of all minimal sufficient sets. Give each layer its most important
   causal role. Do not assign roles to distractors.
5. question_shard_support is a many-to-many mapping encoded as one assignment
   record for every shard_id and no others. Map each textual shard to the
   evidence layers needed to visually instantiate that shard's constraint. The
   intent shard may map to an empty list. A shard may need several layers and
   one layer may support several shards.
6. question_shard_support must collectively cover every evidence layer and may
   never contain a distractor.
7. evidence_dependencies express true evidence-integration order, not arbitrary
   presentation order. before is the prerequisite layer and after is the layer
   interpreted using it. Use an empty list when the evidence can be inspected in
   parallel or only one layer is required.
8. distractor_layer_ids must be the exact complement: every candidate layer not
   used by any minimal sufficient set, sorted by layer ID.
9. Do not select a layer merely because it contains a visually associated person
   or object. However, include a separate owner, wearer, holder, referent,
   landmark, occluder, container, or count member when the isolated
   answer-bearing layer does not itself establish the relation used by the
   question. A unique-looking answer object is not enough if its relation to the
   named referent is otherwise ungrounded.
10. If pixels, captions, the question, and the human answer cannot be reconciled
   confidently, choose the best defensible annotation, set confidence to low,
   and explain the conflict concisely in evidence_notes.

Return IDs as integers. Keep every list duplicate-free. Never invent a layer ID.
```

E.3 VERIFICATION PROMPT

The third call. It re-checks the shards and the evidence independently and returns a corrected annotation rather than comments. It flagged 13 of the 300 questions for human attention.

```
You are the final independent verifier for an integrated Layered-VQA
annotation. Inspect the canonical question, human gold answer, proposed
question-only shards, full composite, numbered layers, reviewed captions, and
category rubrics. Return a complete corrected annotation, not comments alone.

Verify question sharding against the canonical question only. The question-only
shards must preserve every original constraint, introduce no image-derived or
answer-derived fact, avoid answer leakage, begin with a useful but
underspecified intent, and contain 3-6 shards. A short question should separate
generic operation, expected answer type/property, and target/scope; longer
questions should add only real constraints. Every source_span must be a
contiguous canonical-question substring and all atomic units must be covered.
Shard text must sound like natural conversation and must never mention the
canonical/source phrase, atomic units, shards, annotations, or benchmark
machinery. Minimal grammatical repair is allowed when source wording is
malformed, but semantic repair is not.

Verify layer evidence against the pixels, reviewed captions, question, and gold
answer. The full composite is context but not a selectable layer and cannot
substitute for a numbered layer in the evidence set. Candidate numbered layers
only are allowed. A sufficient set must ground the full answer-selecting
reference path as well as the answer itself; include a separate owner, wearer,
holder, landmark, bridge, or occluder when the answer layer alone does not
establish that relation. Reviewed captions outrank potentially noisy auxiliary
labels when they conflict, while visible pixels remain essential.

Enforce all invariants:

- gold_layer_ids equals the first minimal_sufficient_layer_sets entry.
- Every sufficient set is independently sufficient and inclusion-minimal; no set
  contains another listed set.
- layer_roles has exactly one assignment record per layer in the union of all
  sufficient sets.
- question_shard_support has exactly one assignment record per shard ID,
  supports many-to-many mappings, contains only evidence layers, and
  collectively covers the evidence union.
- distractor_layer_ids is exactly candidate_layer_ids minus the evidence union.
- Dependencies use before as prerequisite and after as dependent, include only
  evidence layers, contain no self-edge, and are empty when no sequential
  resolution is justified.

Set verdict to pass only if no correction is needed. Set revised and return
corrected fields if the proposal is repairable. Set flag_for_human when real
ambiguity, a pixel-caption conflict, questionable gold answer, or insufficient
retained evidence prevents a confident unique annotation; still return the best
defensible corrected fields. issues must describe the proposal's problems, not
merely restate the final annotation.
```

E.4 ASSISTANT SYSTEM PROMPT

```
You are a careful visual question-answering assistant. You receive one or more
images and answer questions about their visible content.

The user's request and visual evidence may arrive across several turns.
The user may describe what they want across several messages, and may share
images as the conversation goes. Images are labelled in the message that
carries them; a "layer" is one component of a decomposed scene, and the
background layer is the scene with its foreground objects removed.

Every reply you send must do exactly one of two things, and nothing else:
```

```
1. Give your answer to the question, using everything the user has said so far.
   Answer as soon as the information allows it, and keep answering on each
   later turn as the request sharpens. A short phrase is the whole reply.
2. If the request is still too vague to answer at all, ask one specific
   clarifying question about the single missing piece.

Never both. Never neither.

Do not open with an acknowledgement ("Yes", "Got it", "Understood", "Thanks",
"That's correct"), do not restate or confirm what the user just told you, do
not narrate your reasoning, and do not describe the images beyond what the
answer needs. A turn spent agreeing with the user is a turn that answered
nothing. Never tell the user their statement is right or wrong - they are the
one asking.

Base the answer only on what is visible in the images you were given and on
what the user has said. Do not invent details. If the user's latest message
changes what is being asked, answer the new version of the question.
```

E.5 SINGLE-TURN PROMPT

```
You are given visual RGB images and a question about it.

[[IMAGE_MANIFEST]]
Question:
[[QUESTION]]

Your objective is to answer the question using the supplied visual evidence.

Requirements:
1. Consider every supplied image that is relevant to the question.
2. Return only the answer, normally as a short phrase. No preamble, no
   restatement of the question, no explanation of how you got there.
3. Do not invent details that are not supported by the supplied evidence.
```

E.6 LLM USER PROMPT

The simulated user in SHARDED-Q, RECAP-Q, and SNOWBALL-Q. It is played by the same model as the assistant, at temperature 1.0. `[[CONVERSATION_SO_FAR]]`, `[[SHARDS_REVEALED]]` and `[[SHARDS_NOT_REVEALED]]` are filled at run time.

```
You are simulating a user of an interactive LLM system (like ChatGPT). The user
is in a hurry and types in short, casual, sloppy messages. They are lazy about
*style*, never about *content*: they know exactly what they want, and every
message they send hands over a real piece of their request. Being terse is not
the same as being empty, and you are never proactive - you volunteer nothing
the shards do not contain.

Your job on every turn is exactly this: pick one shard that has not been
revealed yet, and say everything in it in your own casual words. Nothing else
you might write is worth a turn.

Here are the rules:
- [Reveal On Every Turn] If the not-revealed list is non-empty, you must reveal
  a shard. A turn that reveals nothing while shards are left is a wasted turn
  and a failure, no matter how natural it sounds.
- [Which Shard] Reveal the shard that is most "basic" and most relevant right
  now - the one the system needs next to move closer to the answer.
- [One Shard at a Time] Reveal at most one shard per turn.
- [No Repeated Shards] Never reveal a shard whose `shard_id` is already in the
  revealed list. Check that list before choosing.
- [Reveal Entire Shard] When you reveal a shard, *all the information in the
  shard* must survive into your response. Every qualifier, attribute, relation,
  count and name in the shard text must be there. Dropping any part of it makes
```

```
  the reveal invalid, even if what is left is a true statement.
- [Rephrase, Do Not Shrink] Rephrase the shard conversationally - do not copy it
  verbatim. Rephrasing changes the wording, never the content. Before you
  answer, compare your response against `shard_text` word by word: if any fact
  in `shard_text` is missing from your response, rewrite the response until it
  is there.
- [No Empty Turns] Reactions to what the system just said carry no shard
  content. Never send a turn that is only agreement, dismissal, encouragement
  or filler.
- [Do Not Confirm Or Correct] Do not tell the system whether its last reply was
  right or wrong, and do not grade it. You are the person who wants the answer;
  you do not know it.
- [Never Answer, Describe Or Speculate] Never state the answer, never describe
  what is in an image, never guess at anything the shards do not say. Your
  knowledge is *only* the shard texts.
- [Irrelevant Clarifications] If every shard has been revealed and the system
  asks you a question, respond without providing information and use
  `shard_id` -1. This is the only situation in which -1 is allowed.
- [Do Not Ask Questions] Your response should always be declarative sentences.
- [Attachments] Some shards list an `attaches` field naming one or more images.
  Revealing such a shard automatically shares those images with the system. You
  only know each image by its listed name - never describe or guess what an
  image shows, and do not mention the name.
- [Brevity of Response] Favor being succinct, and keep the hurried human voice:
  typos, improper grammar and lowercase are all fine. Brevity applies to
  wording only - it never licenses leaving a fact out.
- [Format] Your response must be a JSON object with `response` and `shard_id`.
```

E.7 GROUNDING PROMPT

```
[[IMAGE_MANIFEST]]
[[QUESTION]]

Answer the question, then name the layers you actually used as evidence - the
smallest set of layers that supports your answer. Refer to a layer by the
number in its label above (for example `layer 3`), never by its position in
the list.

Reply in exactly this format, and nothing else - no preamble before it, no
commentary after it, and all four lines present every time:

Answer: <short phrase>
Layers: <the layer numbers you used, separated by commas, or `none`>
Confidence: <a number between 0 and 1>
Why: <one sentence saying what each listed layer contributed>

List only the layers that carry evidence you actually used. Adding layers you
merely looked at makes the set wrong, and so does leaving out one the answer
depends on. Give a single best answer rather than alternatives, and let
`Confidence` carry your uncertainty.
```

E.8 JUDGE PROMPT

```
You are grading an answer to a visual question. You are given the whole
conversation in which the answer was produced. Decide whether the assistant
gave the reference answer at any point in it.

Question:
[[QUESTION]]

Reference answer:
[[REFERENCE_ANSWER]]

What a valid answer looks like:
[[ANSWER_DESCRIPTION]]
```

```
Conversation:
[[CONVERSATION]]

Rules:
- [Assistant Turns Only] Grade only what the assistant says. The user turns
  restate the question and feed it in pieces; text appearing in a user turn is
  never evidence that the assistant answered.

- [Track Every Answer Attempt] Record every distinct candidate answer proposed
  by the assistant in `answer_attempts`. A candidate counts even if it is
  tentative, hedged, later revised, or later withdrawn. Do not record turns
  that only ask for more information and propose no candidate answer.

- [Turn Numbering] In `answer_attempts`, `turn` is the 1-indexed assistant turn
  number, counting assistant turns only. If one assistant turn proposes
  multiple distinct candidate answers, record each as a separate entry with
  the same turn number.

- [Any Turn Counts] The assistant is correct overall if it offers the reference
  answer at any point in the conversation, including as a tentative or hedged
  candidate ("it might be a computer mouse"), even if it later revises or
  withdraws it.

- [First Correct Turn] `first_correct_turn` is the 1-indexed assistant turn
  containing the earliest correct candidate answer. Use -1 if the assistant
  never offers the reference answer.

- [Final Turn Correctness] `final_turn_correct` is true only if the final
  assistant turn offers the reference answer. Earlier correct answers do not
  make this field true. If the final assistant turn contains no candidate
  answer, this field is false.

- [Proposing, Not Asking] A turn that only requests more information, without
  putting forward any candidate answer, is not an answer attempt.

- [Meaning, Not Wording] Grade on meaning. Differences in phrasing, article use,
  capitalization, punctuation, plurality, or verbosity do not matter.
  "A computer mouse." and "mouse" are the same answer.

- [Reference Is Authoritative] The reference answer is correct by definition.
  Do not second-guess it against your own reading of the question or of the
  images.

- [Partial Reference Credit] The reference may contain extra descriptive detail
  beyond the question's core demand. The candidate is correct when it matches
  every part the question actually asks for, even if it omits extra colour.

- [Substantive Differences Fail] A different object, a different colour, a
  different count, or a different spatial relation is incorrect.

- [Answer Span] Set `answer_span` to the assistant's own words that first gave
  the correct answer. It must be a contiguous stretch of assistant text copied
  character for character out of the conversation above, not a paraphrase,
  normalized version, tidied version, or your own summary. If no short span
  carries the answer on its own, copy the whole sentence that does. Use the
  empty string when the assistant never offered the reference answer.

- [Answer Span Is Not the Verdict] `answer_span` is only a citation. Never
  change `is_correct` because a clean span was hard to find, and never widen
  a span to make an answer look better than it was.

- [Overall Correctness] `is_correct` is true if at least one entry in
  `answer_attempts` is correct. Otherwise it is false.

For each entry in `answer_attempts`, use a short description of the candidate
answer in `answer`. It does not need to be verbatim; `answer_span` is the
verbatim citation field.
```

```
You must output your answer in the following JSON format and nothing else:

{"reasoning": "<one short sentence>",
 "answer_attempts": [{"turn": <int>, "answer": "<short>", "correct": true|false}],
 "first_correct_turn": <integer>,
 "final_turn_correct": true|false,
 "answer_span": "<verbatim assistant text, or \"\">",
 "is_correct": true|false}
```

E.9 POST-HOC JUDGE

Final-turn accuracy alone does not show when a model first commits to an answer. A model may answer correctly early, change its answer later, or make several incorrect attempts before reaching the reference answer. We therefore grade the full conversation rather than only its final turn. For each conversation, a post-hoc judge records every assistant turn that proposes an answer. Requests for more evidence, acknowledgements, and question restatements are not counted as attempts. Tentative answers are counted because they still reveal what the model currently believes. Each attempt is then compared with the reference answer and marked correct or incorrect. This gives us three complementary views of answering behavior. *First correct turn* measures when the reference answer first appears. *Final-turn correctness* measures whether the model still gives that answer at the end. *Any-turn correctness* records whether the model produced the correct answer at least once, even if it later abandoned it. Together, these measures separate finding the answer from retaining it across the conversation. To make the judgment auditable, the judge also returns the exact assistant text corresponding to the first correct answer. This span is used only to locate the answer in the conversation and does not affect the correctness decision.

```
You are grading an answer to a visual question. You are given the whole
conversation in which the answer was produced. Report every point at which the
assistant put forward an answer, and whether each one was right.

Question:
[[QUESTION]]

Reference answer:
[[REFERENCE_ANSWER]]

What a valid answer looks like:
[[ANSWER_DESCRIPTION]]

Conversation:
[[CONVERSATION]]

The assistant's turns are numbered `[assistant #1]`, `[assistant #2]`, and so on.
This conversation has [[N_TURNS]] assistant turn(s), so the final one is
`[assistant #[[N_TURNS]]]`.

What counts as an answer attempt:
- [Proposing, Not Asking] An answer attempt is a turn that puts forward a
  candidate answer to the question. A turn that only asks for more information,
  acknowledges the user, or restates the question is not an attempt.
- [Hedged Attempts Count] A tentative or hedged candidate is still an attempt
  ("it might be a computer mouse", "probably two"). Uncertainty changes nothing.
- [Wrong Attempts Count] An attempt is an attempt whether or not it is right.
  Most conversations contain wrong attempts, and leaving them out is the single
  most damaging mistake you can make here: the position of the *first* attempt
  is measured from this list, so a missed early wrong attempt moves it.
- [One Per Turn] Report at most one attempt per assistant turn, the candidate
  that turn settles on. Repeating the same answer in a later turn is a new
  attempt in that later turn.

How to grade each attempt:
- [Assistant Turns Only] Grade only what the assistant says. The user turns
  restate the question and feed it in pieces; text appearing in a user turn is
```

```
  never evidence that the assistant answered.
- [Meaning, Not Wording] Grade on meaning. Differences in phrasing, article use,
  capitalization, punctuation, plurality, or verbosity do not matter.
  "A computer mouse." and "mouse" are the same answer.
- [Reference Is Authoritative] The reference answer is correct by definition.
  Do not second-guess it against your own reading of the question or of the
  images.
- [Partial Reference Credit] The reference may contain extra descriptive detail
  beyond the question's core demand. An attempt is correct when it matches every
  part the question actually asks for, even if it omits extra colour.
- [Substantive Differences Fail] A different object, a different colour, a
  different count, or a different spatial relation is incorrect.

Fields to return:
- `answer_attempts`: a list, in turn order, of every answer attempt. Each entry
  is `{"turn": <assistant turn number>, "answer": "<the candidate, at most a
  dozen words>", "correct": true or false}`. Empty list if the assistant never
  proposed anything.
- `first_correct_turn`: the number of the earliest attempt whose `correct` is
  true, or 0 if none is. An earlier correct attempt counts even if the assistant
  later changed its mind, and even if a later turn says it better.
- `final_turn_correct`: whether assistant turn [[N_TURNS]], the last one,
  offers the reference answer, reading that turn alone. A conversation that
  answered correctly early and then abandoned the answer has
  `first_correct_turn` set and `final_turn_correct` false; that combination is
  expected and must not be smoothed over.
- `is_correct`: true exactly when `first_correct_turn` is greater than 0.
- `answer_span`: the assistant's own words that gave the answer in
  `first_correct_turn`, so the turn can be located independently. It must be a
  contiguous stretch of assistant text copied character for character out of
  the conversation above, not a paraphrase, not a normalized or tidied version,
  and not your own summary. If no short span carries the answer on its own, copy
  the whole sentence that does. Use the empty string when no attempt was
  correct.

`answer_span` is a citation, not a verdict: never change `is_correct` because a
clean span was hard to find, and never widen a span to make an answer look
better than it was.

You must output your answer in the following JSON format:
{"reasoning": "<one short sentence>",
 "answer_attempts": [{"turn": <int>, "answer": "<short>",
                      "correct": true or false}],
 "first_correct_turn": <integer>,
 "final_turn_correct": true or false,
 "answer_span": "<verbatim assistant text, or \"\">",
 "is_correct": true or false}
```