GAMUT: A BENCHMARK FOR FACTUAL COMPLETENESS WITH TWO-LEVEL META-RUBRICS

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce GAMUT (*Grounded Assessment of Multimodal Factuality*), an *everyday deep research* benchmark for factual completeness, built from naturally phrased questions concerning everyday subjects, yet requiring multi-step research and multi-paragraph answers. Evaluating these answers exposes a fundamental tension in rubric-based evaluation. A faithful rubric must express the structure of the space of good answers: open-ended sets of acceptable options, ordered processes, and relative importance. Consistent automatic grading, however, favors narrow, independently checkable criteria. GAMUT addresses this tension with a **two-level meta-rubric** design: a structured meta-rubric captures answer requirements at authoring time, which are compiled by fixed mechanical rules into a flat binary checklist for evaluation. The benchmark comprises 1,813 questions grounded in real-world images across 10 domains, each paired with an evidence-backed rubric verified by expert human annotators. Evaluating 14 frontier and open-weight models, we find GAMUT challenging and highly discriminative, with scores ranging from 5.1% to 58.7%, and broadly consistent rankings across three LLM judges.[1]

1 INTRODUCTION

We introduce GAMUT (*Grounded Assessment of Multimodal Factuality*), a benchmark for factual completeness in long-form answers. Most work on evaluating the factuality of long-form generation has focused on *precision*: of the claims a response makes, what fraction are correct. The dominant decompose-search-verify (DSV) paradigm breaks a response into atomic claims and verifies each against retrieved web documents, as in FActScore (Min et al., 2023) and VeriScore (Song et al., 2024). Comparatively little attention has gone to *recall*: whether a response contains all the information a complete answer should, which a few recent works target directly (Liu et al., 2025). But both lines rest on a more questionable assumption: that the factual quality of an answer is captured by a list of independent boolean fact-checks. Reducing factuality to such a checklist is convenient for scoring, but it misrepresents how the quality of a knowledge-seeking answer is determined.

Several common situations break this model. Many questions are open-ended, with a large or unbounded set of acceptable facts: asked which traditional dishes use a given ingredient, a good answer names enough of them, but no single dish is required and two answers listing different valid dishes can be equally good, which a fixed list of individually required fact checks cannot express. Other questions turn on a process whose order matters, such as how an everyday object reached its modern form, where stating the milestones out of sequence is an error even when each is individually correct. Treating every fact as an independent, individually required checkbox discards these distinctions, along with the ability to ask whether an answer covers enough of an open set or respects a necessary order. Measuring factual completeness faithfully therefore calls for a richer representation: a *structured meta-rubric* of the facts a complete answer should contain, organized by importance and by the relations, orderings, and groupings among them.

Such structure, however, is not suited to reliable automatic evaluation, since judges are far more consistent scoring flat, binary, independently checkable items than making holistic judgments (Lee et al., 2025; Rao & Callison-Burch, 2026). GAMUT resolves this tension with a **two-level representation**: a structured meta-rubric captures the organization and importance of the required content, which is

[1]The GAMUT dataset is publicly available; the URL is withheld for double-blind review.

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

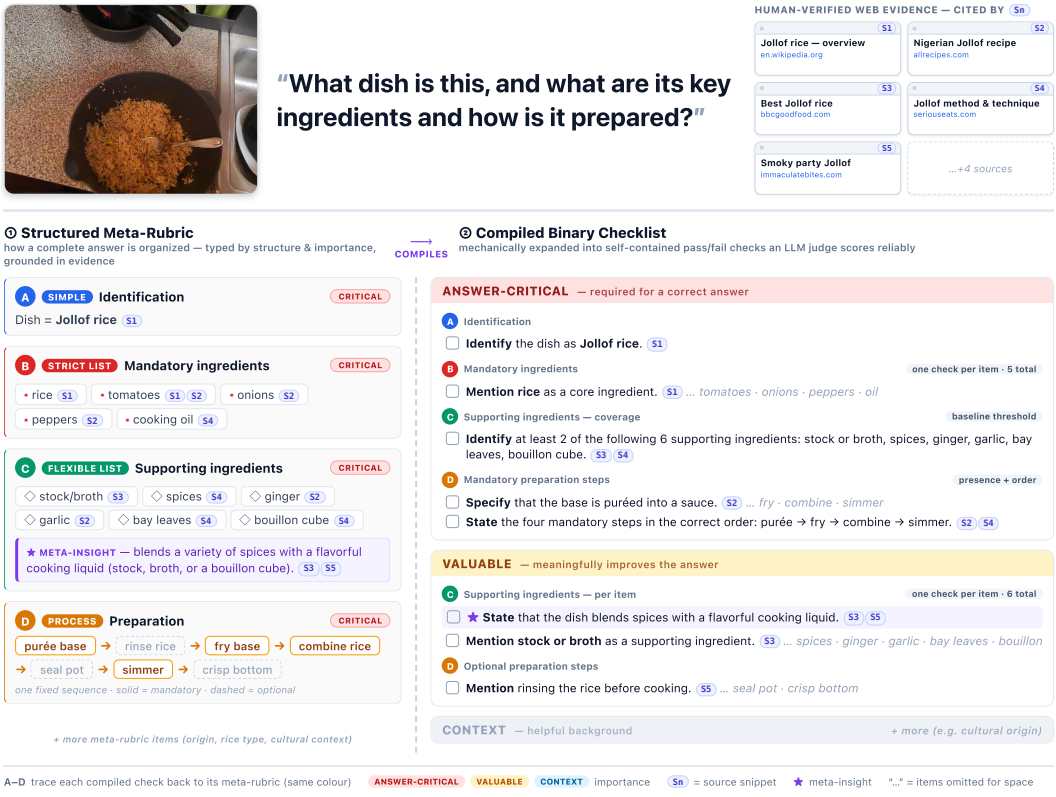


Figure 1: A representative example (jollof rice) from GAMUT (some rubrics not shown for brevity). Top: an *everyday deep-research* question and human-verified web evidence used to ground the rubrics. Left: the structured meta-rubric, with typed components (Simple Knowledge, Strict List, Flexible List with a meta-insight, Process), importance tiers, and citations. Right: the compiled binary checklist organized by importance tier, showing how each meta-rubric item mechanically expands into self-contained pass/fail checks. More details can be found in Section 3, and complete examples are given in Appendix C.

then mechanically compiled into a flat checklist of binary, machine-gradable rubric items used at evaluation time. The rubric is built and verified in the structured space while grading happens in the converted binary space, retaining the expressiveness of structured rubrics while inheriting the low variance of binary LLM-as-a-judge scoring.

GAMUT targets a setting we call *everyday deep research*: realistic, naturally phrased questions a person would ask about something in view, which nonetheless require multi-step research and multi-paragraph answers. Figure 1 shows one such question and illustrates how GAMUT represents what a complete answer should contain, from the structured meta-rubric to the compiled binary checklist. This is deep research in the sense of the research process, not academic depth; the questions cover explorative learning, troubleshooting, and shopping, rather than report-level questions on academic topics. Its 1,813 questions are grounded in real-world images drawn from CRAG-MM (Wang et al., 2025), spanning 10 domains. While GAMUT is multimodal-first, its evaluation framework is modality-agnostic, and we also release a text-only variant whose questions are made self-contained.

Constructing GAMUT involves two tasks, Question Creation and Rubric Creation, each pairing frontier LLMs with human annotators over multiple rounds, so that neither the questions nor the reference facts rest on unverified model output (Section 4). Evaluating 14 frontier and open-weight models reveals that GAMUT is genuinely challenging, with the strongest (Gemini 3.1 Pro) reaching a GAMUT score[2] of only 58.7%, highly discriminative, and robust in its model rankings to the choice of judge. The text-only variant further shows that removing visual identification raises every model's

[2]The GAMUT score is an importance-weighted average of how many of a rubric's binary checks a response satisfies, ranging from -100% to 100% with contradictions penalized below omissions (Section 3.3).

score by an amount largely independent of its multimodal score, so the multimodal ranking reflects differences in knowledge completeness rather than perception alone.

This work makes three contributions. First, it presents GAMUT, a benchmark of everyday deep-research questions for factual completeness, with evidence-backed, human-verified rubrics and a text-only variant. Second, it introduces a two-level rubric representation that reconciles the expressiveness needed for annotation with the reliability needed for automatic grading. Third, it benchmarks 14 frontier and open-weight models and shows that factual completeness is far from solved.

2 Related Work

2.1 Long-Form Factuality Evaluation

Most work on long-form factuality has measured *precision* through a decompose-search-verify (DSV) pipeline: a response is broken into atomic claims, each verified against retrieved web documents (Min et al., 2023; Wei et al., 2024; Song et al., 2024; Chern et al., 2025; Fatahi Bayat et al., 2025). A thinner thread targets *recall*, checking whether a response covers the reference facts a complete answer should contain (Liu et al., 2025; Sharifymoghaddam et al., 2025; Chen et al., 2025; Jeong et al., 2024). A few methods add event order (Zheng et al., 2025) or importance-weighted recall (Wanner et al., 2025; Jafari et al., 2026) to atomic-fact evaluation, but each remains a list of atomic facts with one added axis, far short of representing an answer's content as structured, gradable targets. This limitation, with DSV's reliance on noisy web search, motivates the structured meta-rubric of Section 3.

2.2 Rubric-Based Evaluation

Grading free-form responses with a strong LLM against stated criteria (Liu et al., 2023; Zheng et al., 2023) is now a standard alternative to reference-based metrics, but holistic Likert judgments are noisy, which has driven a move to rubric-based grading against an explicit list of criteria (Kim et al., 2024; Ye et al., 2024), and further to binary checklists, which lower score variance and raise agreement (Lee et al., 2025; Rao & Callison-Burch, 2026). Applied to factuality and long-form generation, such rubrics grade against expert-written criteria (Asai et al., 2026; Arora et al., 2025; Sharma et al., 2025; Ruan et al., 2025). Closest to GAMUT is FACTS Multimodal (Cheng et al., 2025; Jacovi et al., 2025), which scores image-grounded answers against a human-authored rubric of reference facts labeled essential or non-essential. The two differ on both ends: its questions elicit a few essential facts, whereas GAMUT targets richer questions whose complete answers span many facts; and it sorts those facts into a single essential-versus-non-essential split, without the ordering, grouping, or open-ended sets GAMUT represents. Its rubrics are authored and graded in the same space; GAMUT instead uses expressive structured authoring and mechanically compiled, narrow, self-contained grading checks.

2.3 Multimodal Knowledge-Seeking Benchmarks

GAMUT relates to multimodal retrieval and agentic research benchmarks (Mialon et al., 2024; Wei et al., 2025; Jiang et al., 2025; Li et al., 2025), which pair each question with a single short, verifiable answer, whereas GAMUT asks open-ended questions and scores factual completeness against a reference-fact rubric. Closest in setting are multimodal deep-research benchmarks (Ye et al., 2026; Huang et al., 2026), which grade the quality of a generated report rather than the factual completeness of an answer. MMDeepResearch-Bench scores reports with an adaptive LLM-as-a-judge, the holistic scoring GAMUT avoids. MiroEval instead grades against a rubric, but one of report-level process and outcome criteria rather than the a priori reference facts a complete answer must contain. We target *everyday deep research*: naturally phrased questions that require multi-source research yet stay everyday in topic, rather than academic or report-level tasks.

3 Two-Level Meta-Rubrics for Evaluating Open-Ended Generation

A faithful rubric must capture the coverage and ordering among facts that determine a good answer, but a judge asked to weigh that structure while scoring reverts to the holistic, high-variance judgment

that rubrics were meant to avoid. GAMUT resolves this tension with a **two-level representation**: a structured *meta-rubric* describes the space of good answers (Section 3.1), and fixed mechanical rules convert it into a flat *binary rubric* of pass-or-fail checks (Section 3.2) used for scoring (Section 3.3). The expressive structure thus lives where a good answer is described, and the reliable binary checks live where the judge operates. We develop it for long-form factuality, but it applies to rubric-based evaluation of open-ended generation more broadly.

3.1 META-RUBRICS

A meta-rubric describes answer quality the way a knowledgeable reviewer reasons about it: which points are essential, which sets of options are acceptable, what must be said in order, and what merely rounds out the answer. It is a set of *meta-rubric items*, each carrying a *type* and an *importance tier*. The three tiers are *Answer-Critical* for content a correct answer must contain, *Valuable* for content that meaningfully improves it, and *Context* for helpful background. An item takes one of the following five types:

Simple Knowledge A single discrete fact, such as the identity of an entity or a specific date.

Strict List A finite set in which every item is required, such as the core ingredients of a dish. Naming the set as a list, rather than as separate facts, makes a missing item easier to catch.

Flexible List A pool of valid options in which sufficient coverage is required but no specific item is, such as which dishes use a given ingredient. It specifies a *baseline threshold* of minimum acceptable coverage, for example naming at least two options. When a concept has both a required core and an optional remainder, it is split into a strict list and a flexible list so each stays internally uniform.

Process An *ordered* sequence of steps in which the order carries meaning, such as how an object reached its modern form. A process may have both required and optional steps.

Beyond the items, a list or process may carry *meta-insights*: a synthesis across items, such as an overarching trend or categorization, that no single item captures. For a dish prepared with different red meats, the specific meats form a flexible list, but the pattern that it essentially always uses some red meat is a meta-insight. Its importance is assigned case by case, since some meta-insights are as essential as the underlying facts while others are merely secondary.

Relationship A connection or contrast between entities, such as how one option compares to an alternative or how one property depends on another.

3.2 FROM META-RUBRICS TO BINARY RUBRICS

Asking a judge to weigh coverage, importance, and ordering at once would reintroduce the holistic judgment we set out to avoid, so GAMUT converts each meta-rubric into a flat *binary rubric* of pass-or-fail checks by fixed rules, making the conversion deterministic and auditable. Most of the design lies in how the flexible structures are handled.

Simple knowledge A simple-knowledge item becomes one check.

Strict lists A strict list becomes one check per item, each inheriting the list's importance tier.

Flexible lists A flexible list becomes two kinds of check: one or two baseline thresholds and additional credit for broader coverage. The baseline is a check of the form "mention at least N of the following: ..." with all options enumerated so the check is self-contained, inheriting the list's importance. The additional credit sits one tier lower and depends on pool size: a *short* list (≤ 7 items) produces one check per item, while a *long* list produces a few higher coverage thresholds instead, avoiding a flood of low-value checks.[3] For a short Answer-Critical list of five dishes, the conversion produces an Answer-Critical check "mention at least 2 of the following", enumerating all five, plus five Valuable per-dish checks, so naming a dish outside the five neither helps nor hurts. GAMUT thus measures how much of the acceptable space an answer covers, not whether it matched a fixed list.

[3] A Context-level short list, having no tier beneath it, is treated as long.

Processes A process produces one presence check per step, required steps at the list's tier and optional steps one lower. It adds a sequence check over the required steps at the list's tier, and, when optional steps are present, a second over the full ordering one tier lower.

A **meta-insight** (for a list or process) becomes a separate check at its assigned tier.

Relationships A relationship becomes a check identifying the related entity plus one check per aspect of the connection.

The compiled rubric is again a flat list of binary checks. But its threshold and sequence checks grade coverage of an open set and correct ordering, which a one-fact-per-check rubric cannot, and each check's tier records how much its content matters. Appendix C works two complete questions through end to end.

3.3 SCORING

Given the binary rubric and a response, we compute a GAMUT score in three steps: a verdict on each check, a score within each tier, and a weighted combination across tiers.

Per-check verdicts An LLM judge evaluates the response against each check independently and returns one of four verdicts. A check can fail in two ways a binary met-or-not verdict would collapse: the response can *miss* the required content or *contradict* it. Scoring these the same would treat a cautious answer and a confidently wrong one alike, so we separate them and penalize contradiction more heavily, which is essential for a benchmark whose purpose is factual reliability. We further distinguish *meets* from *partially meets* to credit a response that is directionally right but imprecise, such as the correct genus but not species.

Per-tier score Verdicts map to numbers: *meets* scores 1, *partially meets* scores λ, *missing* scores 0, and *contradicts* scores μ, with $0 < \lambda < 1$ and $\mu < 0$ so a falsehood costs more than an omission. Writing M, P, S, and C for the counts of each verdict, the tier score is

$$s = \frac{M + \lambda P + \mu C}{M + P + S + |\mu| C}, \tag{1}$$

which lies in $[-1, 1]$; we report scores as percentages. Because μ enters both numerator and denominator, a contradiction costs more than one check, so a cautious answer outranks a confident but wrong one. We use $\lambda = 0.5$ and $\mu = -2$.

Combining tiers The GAMUT score is a weighted average of the three tier scores with global weights w_{AC}, w_{V}, w_{C}, set so that covering the Answer-Critical content alone reaches roughly a passing score. The weights are global rather than implied by per-tier check counts, which vary widely across questions, so that a question with many minor Context checks cannot drown out its few Answer-Critical ones. We use $w_{\text{AC}} = 0.6$, $w_{\text{V}} = 0.3$, $w_{\text{C}} = 0.1$, and when a question has no checks in a tier its weight is redistributed over the rest.

4 DATASET CONSTRUCTION

GAMUT is built in two stages: we create *everyday deep research* questions grounded in real-world images from CRAG-MM (Section 4.1), then construct a structured meta-rubric for each and convert it into the binary rubric used for scoring (Section 4.2). Both stages pair a frontier LLM, which proposes candidates at scale, with human annotators who review and revise them over multiple rounds.

4.1 QUESTION CREATION

The goal is to elicit questions a person would naturally ask about something in view, yet require multi-step research to answer.

Source images We draw 1,938 real-world images and their ground-truth entities from CRAG-MM (Wang et al., 2025). Most images (roughly 80%) were taken by people wearing smart glasses; the object of interest is often small, rotated, truncated, occluded, or poorly lit. The entities span 10

domains of everyday life, from plants, food, and animals to vehicles, local places, everyday objects, and consumer products, and range across head, torso, and tail popularity.

Candidate generation For each image we prompt a frontier multimodal LLM[4] to propose open-ended candidate questions (Appendix B.1) that require multi-step web research and a multi-paragraph answer, are concise and natural as a user would actually speak, and refer to the entity only through image-dependent terms.

Multi-round human revision Annotators then *accept*, *revise*, or *discard* each candidate, with every question reviewed independently by two annotators and a third adjudicating disagreements. A question accepted by at least two advances, a revised one re-enters the next round, and when an image's candidates are all discarded, new ones are generated, repeating until every image has a question that passes both human review and the automatic pass below.

Automatic filtering, ranking, and diversification A second LLM pass rejects *leading* questions, those answerable only by someone who already recognized the entity, then ranks the rest and seeds diverse candidates to counter the templating of independent per-image generation (see Appendix B.2). The top question accepted by both the annotators and the LLM becomes the image's entry in GAMUT. The procedure yields 1,843 questions, one per image; we analyze their diversity in Section 5.

4.2 RUBRIC CREATION

For each question we construct the structured meta-rubric and the binary rubric it converts to, following Section 3. As with questions, an LLM produces the initial rubric at scale and expert annotators verify and revise it, but rubric creation is far more demanding: it requires gathering the facts a complete answer should contain, organizing them into the right structures, and checking the conversion into the binary rubric used for evaluation.

Generation We prompt the LLM (Appendix B.3) to first research the question on the web, then build a structured meta-rubric from the retrieved knowledge, citing the supporting snippets, and compile it into the binary rubric by the rules of Section 3. The **evidence snippets** are provided so the rubrics trace to verifiable sources.

Self-refinement After the initial generation, we prompt the LLM to refine its own rubric (Appendix B.4): an audit phase re-verifies every snippet, flags partially covered lists or processes, and an independent search phase re-researches the question for any missing snippets. A final phase rebuilds the meta-rubric and re-compiles the binary rubric. We did one round of self-refinement, since a second round changed little.

Multi-round human revision The refined rubrics are revised by a small in-house team working closely with the authors, since a general annotation pool could not improve on the LLM-generated rubrics, which demand both subject-matter judgment and fluency with the structured representation. Annotators are shown the meta-rubric but revise the binary rubric directly, because it is what is used for scoring, so revising it lets them check the conversion itself; they are trained to reason in the structured space, using the meta-rubric as a scaffold to spot a missing list item, a misordered step, or a miscategorized fact. The annotators also refine the web snippets, browsing each source, dropping unsupported ones, and adding missed evidence. After a first revision round we run another pass of LLM refinement followed by a second human round.

Entity and question corrections The annotators surfaced a few upstream errors, cases where the CRAG-MM ground-truth entity was wrong or a question was ambiguous about which object it referred to, which we corrected by hand in 12 examples. A small number of questions were dropped when no sound rubric could be produced, leaving 1,813 questions with complete, human-verified rubrics. Annotator training and auditing details for both stages are given in Appendix D.

[4]We use Gemini 3.1 Pro throughout question and rubric construction.

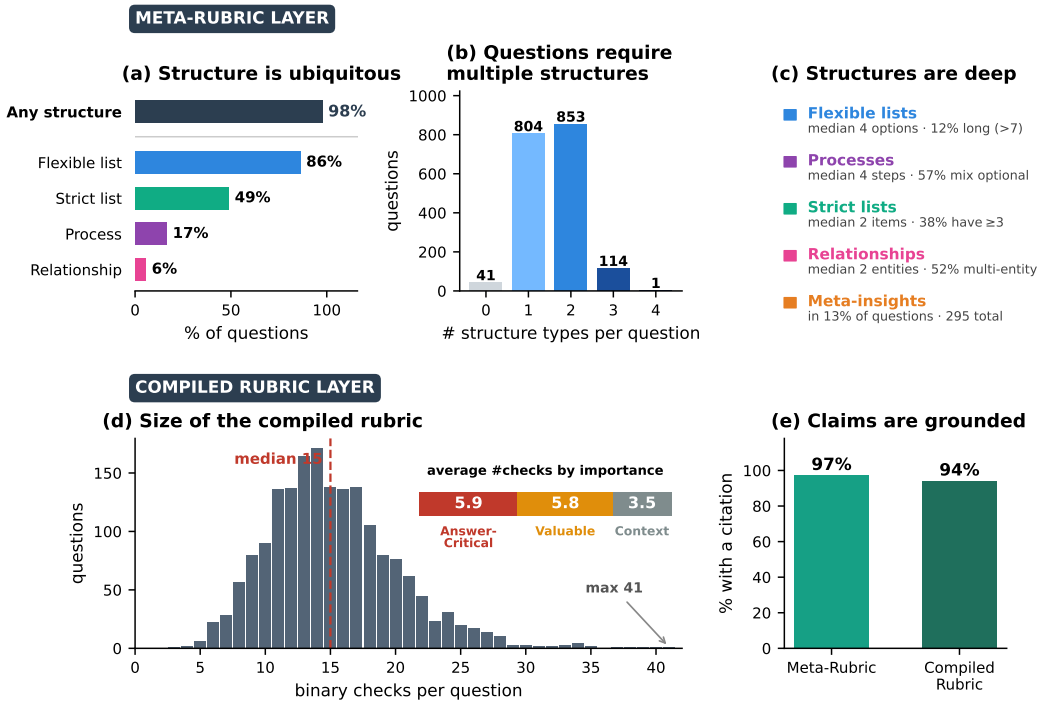


Figure 2: Rubric statistics for GAMUT ($n = 1{,}813$). The structure the two-level design captures is pervasive, not incidental: almost every question needs at least one component beyond plain facts, most need several, and those components are deep, so a one-fact-per-check rubric would misrepresent the large majority of questions. Panels: (a) prevalence of each meta-rubric type, (b) how many types co-occur per question, (c) depth of the structures, (d) size and importance-tier breakdown of the compiled checks, (e) evidence grounding at both levels: 97% of meta-rubric items and 94% of compiled checks cite a supporting web snippet.

5 Dataset Analysis

We analyze both the questions and the rubrics of GAMUT. The questions are diverse and complex while staying everyday (Section 5.1). The rubrics (Sections 5.2 and 5.3) showcase the meta-rubric structure at the core of our design: real, ubiquitous, and useful. Their depth also reflects how demanding the questions are, since a rubric enumerates what a complete answer requires.

5.1 Question Diversity

Image-grounded question sets risk collapsing into a few templates, such as "what is this and what is it used for" repeated across every image, which the pipeline of Section 4.1 counters in both surface form and content. On the surface the questions are short (median 23 words) yet not templated: the 25 most common opening phrases cover under 60% of them, and about one in six do not begin with a standard interrogative, opening instead with framings such as "Based on . . . " or "If I wanted . . . ". In content each domain spans several recurring topics rather than one dominant template (Figure 5 in Appendix E): questions about plants range over habitat and care, those about local places over history and architecture, those about vehicles over specifications and reliability, and those about food over ingredients and regional variation.

5.2 Rubric Statistics

Each question carries a rubric of 15.2 binary checks on average (median 15), split across the three tiers at roughly 5.9 Answer-Critical, 5.8 Valuable, and 3.5 Context checks (Figure 2). The rubrics are grounded in human-verified evidence: each example cites about 10 web snippets, and across the benchmark these cite over 9,400 distinct web pages.

Table 1: Model performance on GAMUT. GAMUT score on the 1,813 multimodal questions under the main judge (Gemini 3.1 Pro, which produces every other column) and two alternative judges (Claude Opus 4.8, Qwen3-VL 235B); per-tier scores are the unweighted scores within each importance tier; the verdict distribution is the percentage of checks receiving each verdict; *Text-only* is the GAMUT score on the text-only variant (1,806 questions, Section 6.2) and its gain over the multimodal score. Scores range from -100% to 100%; Δ is in percentage points (Section 3.3); the full judge and text-only tables are in Appendix E.

		GAMUT score by judge			Per-tier score			Verdict distribution (%)				Text-only	
	Model	Gemini	Claude	Qwen	AC	Val	Ctx	meets	partial	miss.	contra.	score	Δ
Proprietary	Gemini 3.1 Pro	**58.7**	58.7	67.0	63.9	51.9	45.1	55.2	11.5	29.2	4.1	73.7	+15.0
	Gemini 3 Flash	57.9	57.4	66.1	63.0	51.6	44.7	54.9	12.1	28.5	4.5	70.8	+12.9
	Claude Opus 4.8	53.1	52.8	60.1	59.4	43.9	40.4	45.3	13.5	38.6	2.6	59.9	+6.8
	Gemini 2.5 Pro	51.8	51.7	60.9	55.8	47.2	39.7	52.1	11.5	30.5	5.9	66.2	+14.4
	Claude Opus 4.6	50.5	50.1	59.1	54.6	45.7	38.8	49.2	12.9	32.6	5.2	62.1	+11.6
	GPT-5.4	42.6	41.3	51.0	46.8	37.8	29.0	39.5	14.0	42.0	4.5	58.5	+16.0
	Claude Sonnet 4.6	41.5	40.4	48.3	46.7	34.3	30.5	38.4	13.5	43.2	4.9	53.8	+12.2
	GPT-4o	34.2	33.6	38.8	39.1	28.0	21.5	29.7	12.6	54.3	3.4	56.5	+22.3
Open-weight	Qwen3-VL 235B	33.0	31.2	44.0	34.1	32.6	26.4	39.4	12.1	39.8	8.7	47.2	+14.2
	Llama 4 Maverick	18.3	16.7	22.1	19.5	17.3	12.3	22.9	11.7	57.9	7.6	36.5	+18.2
	Llama 4 Scout	14.1	12.8	18.6	14.4	14.3	10.5	21.1	11.3	59.3	8.3	27.8	+13.7
	Qwen3-VL 8B	13.7	12.6	23.0	12.3	17.0	11.5	27.3	10.6	50.7	11.5	23.4	+9.7
	Llama 3.2 90B	11.8	10.1	16.3	12.7	10.9	8.2	17.8	10.0	65.1	7.2	25.2	+13.4
	Llama 3.2 11B	5.1	3.3	9.6	3.8	7.8	4.3	14.2	9.2	68.3	8.3	17.7	+12.7

5.3 THE STRUCTURE BEHIND THE RUBRICS

The checks are compiled from a structured meta-rubric, and that structure is pervasive. Only a small fraction of questions are answerable by isolated facts alone: 98% have a rubric with at least one structured component beyond Simple Knowledge, and more than half require two or more structures. Flexible lists are the most common (86% of questions), reflecting how often a good answer must cover enough of an open-ended set; 49% also contain a strict list, 17% an ordered process, and 6% an explicit relationship.

The structures are also deep rather than nominal. Flexible-list pools range from a few options to several dozen (median four), and 357 are long enough that tiered coverage thresholds replace per-item checks; processes carry a median of four steps, and 182 of 318 mix optional with mandatory steps, so the ordering and partial-credit logic of Section 3 are exercised rather than hypothetical. Beyond item coverage, 13% of questions carry a meta-insight, an overarching pattern or end-state a complete answer should convey, which a per-fact checklist has no place to record.

A one-fact-per-check rubric would likely misrepresent these answer requirements: treating every option in a flexible list as required would penalize otherwise adequate answers, while checking only step presence would miss incorrect ordering.

6 EVALUATING MODELS ON GAMUT

We score every model with the procedure of Section 3.3, using Gemini 3.1 Pro as the LLM judge (Appendix B.5) and weights $w_{\text{AC}} = 0.6$, $w_{\text{V}} = 0.3$, $w_{\text{C}} = 0.1$ with $\lambda = 0.5$, $\mu = -2$. Each model is given the image and question and produces a free-form answer, which the judge grades against the binary rubric.

6.1 MAIN RESULTS

Table 1 reports the GAMUT score, per-tier scores, and verdict distribution for each model on the 1,813 multimodal questions. Three findings stand out. First, GAMUT is challenging: the best model reaches a GAMUT score of only 58.7%. Second, the metric is highly discriminative: newer and larger models outscore older and smaller ones, proprietary outscore open-weight, and known tendencies surface in

the verdicts, such as the two Qwen3-VL variants showing the highest contradiction rates and Claude Opus the lowest at 3%.

Judge agreement Third, model rankings are robust to the choice of judge: we re-score every model with two additional judges, Claude Opus 4.8 and Qwen3-VL 235B (Table 1; full table in Appendix E). Gemini and Claude produce an identical ranking, with per-model scores within 1.8 percentage points of each other, and neither favors its own answers: each scores itself within 0.3 points of how the other scores it. Qwen3-VL 235B, a smaller open-weight judge, is more lenient by 4 to 11 points but preserves the same broad ordering, with only local reorderings among closely spaced models; its two largest lifts go to the two Qwen3-VL models (+11.0 and +9.3 points, against 3.8 to 9.1 for the rest), each of which moves up in its ranking, the one sign of self-preference we observe.

Where models fall short Grading each check separately shows which parts of a complete answer models get wrong or omit (Table 1; breakdown by check kind in Table 4, Appendix E). Scores fall from Answer-Critical to Valuable to Context for 12 of 14 models: models cover the essentials first but miss the detail needed for a complete answer. The two exceptions are the weakest models, which fail even on essential content. The verdicts explain this pattern: the more essential the content, the more a model must commit to it, so failures appear as contradictions rather than omissions. A meaningful answer must name the pictured subject, and this Answer-Critical check is *missing* for only 1% to 25% of questions across models but *contradicted* for 11% to 38%. The remaining Answer-Critical checks are missing for 16% to 58% and contradicted for 3% to 16%; Valuable and Context checks are missing for 35% to 81% and contradicted for under 8%. For every one of the 14 models, the missing rate rises from each group to the next. Beyond the subject, 84% to 95% of failures are omissions of content the rubric marks as required or valuable, not misstatements of it.

6.2 A TEXT-ONLY VARIANT

To evaluate factual completeness without visual identification, we construct and release a *text-only* variant in which each question is rewritten to name its subject explicitly (Appendix B.6). The same conversion also removes the Answer-Critical rubric item that identifies the subject, which the converted question makes trivial. Of the 1,813 questions, 7 remain image-dependent even after conversion and are excluded, leaving 1,806 answerable from text alone.

Results The last two columns of Table 1 report text-only scores alongside the multimodal ones (full table in Appendix E). Every model scores higher without the image and the ordering is broadly preserved, so the difference between the two settings isolates the cost of visual identification. That cost ranges from 7 to 22 points (10 to 20 for 11 of the 14 models) and is largely independent of a model's multimodal score.

7 CONCLUSION

We introduced GAMUT, an everyday deep research benchmark for factual completeness, with 1,813 questions grounded in real-world images. Its two-level meta-rubrics resolve the tension between expressiveness and reliability. A structured meta-rubric captures the open-ended sets, ordered processes, relationships, and importance tiers a complete answer involves. Fixed rules convert it into a flat binary rubric an LLM judge scores reliably. This structure is not incidental: 98% of GAMUT's questions require a component a one-fact-per-check rubric would misrepresent.

The benchmark is far from solved, with the strongest model reaching a GAMUT score of only 58.7%, yet discriminates current systems with stable rankings that hold across judges. The text-only variant raises scores but leaves substantial differences between models, showing that the benchmark tests knowledge completeness rather than perception alone. While we develop it for long-form factuality, the two-level meta-rubric is a general recipe for rubric-based evaluation of open-ended generation where a flat list of criteria falls short. We release GAMUT, its text-only variant, and our evaluation scripts; limitations are discussed in Appendix A.

AI USE STATEMENT

Large language models are used throughout this project, in every case under human review. Among tasks with required disclosure, we used generative AI to generate candidate data: Gemini 3.1 Pro proposes the candidate questions and rubrics during dataset construction (Section 4), which human annotators then verify and revise over multiple rounds; it also serves as the LLM judge whose verdicts produce the reported scores (Section 3.3), and rewrites questions for the text-only variant (Section 6.2). Claude Opus 4.8 assisted in implementing the data pipeline and evaluation code; Claude and GPT models also assisted in interpreting the results. Among tasks with recommended disclosure, Claude Opus 4.8 assisted with drafting and revising this manuscript and producing its figures and charts, and both models were used for brainstorming and discussion. The authors have reviewed all AI-assisted work and take full responsibility for the final content of this paper, including all text, claims, and released artifacts.

ETHICS STATEMENT

GAMUT uses images and ground-truth entities from the publicly released CRAG-MM benchmark (Wang et al., 2025). We release no new images and do not redistribute CRAG-MM images; each is referenced by its CRAG-MM identifier and remains subject to the source release terms (CC BY-NC 4.0). Most source images are egocentric, but every question concerns the ground-truth entity in view, not the photographer. The few questions asking a person's identity concern public figures depicted in statues, monuments, murals, paintings, posters, and book covers. Annotators verified the rubric facts against public web sources and cited the supporting evidence. They received written guidelines, training sessions, and ongoing support for question and rubric creation (Appendix D); no personal data about annotators is released. GAMUT is released under CC BY-NC 4.0 for benchmarking. The LLM-generated questions and rubrics, despite human revised and verified, may still inherit biases from the generating model and the English-language web sources they cite, while domain coverage follows CRAG-MM rather than the full range of everyday information needs.

REPRODUCIBILITY STATEMENT

We release the GAMUT dataset (1,813 questions, each with its compiled binary rubric organized by importance tier and the cited web-evidence snippets that support each check), the text-only variant (1,806 questions), and the complete evaluation harness (LLM-judge client, judge prompt, scoring, and coverage validation). The meta-rubric types and the deterministic rules that compile them into binary checks are specified in Sections 3.1 and 3.2, and the scoring procedure with all constants ($\lambda = 0.5$, $\mu = -2$, $w_{\text{AC}} = 0.6$, $w_{\text{V}} = 0.3$, $w_{\text{C}} = 0.1$) is given in Section 3.3 and Section 6; the released harness implements exactly this procedure with these defaults and works against any OpenAI-compatible judge endpoint. Every prompt used in the pipeline (question generation and selection, rubric generation and refinement, the LLM judge, and text-only conversion) is reproduced verbatim in Appendix B; the construction and annotation protocol are described in Section 4 and Appendix D; complete worked examples appear in Appendix C; and per-model results under all three judges are reported in Table 1 and Appendix E.

REFERENCES

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, Johannes Heidecke, and Karan Singhal. HealthBench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*, 2025. URL https://arxiv.org/abs/2505.08775.

Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D'arcy, David Wadden, Matt Latzke, Minyang Tian, Pan Ji, Shengyan Liu, Hao Tong, Bohao Wu, Yanyu Xiong, Luke Zettlemoyer, Graham Neubig, Dan Weld, Doug Downey, Wen-tau Yih, Pang Wei Koh, and Hannaneh Hajishirzi. Synthesizing scientific literature with retrieval-augmented language models. *Nature*, 2026. doi: 10.1038/s41586-025-10072-4. URL https://www.nature.com/articles/s41586-025-10072-4.

Mingda Chen, Yang Li, Xilun Chen, Adina Williams, Gargi Ghosh, and Scott Yih. FACTORY: A challenging human-verified prompt set for long-form factuality. *arXiv preprint arXiv:2508.00109*, 2025. URL https://arxiv.org/abs/2508.00109.

Aileen Cheng, Alon Jacovi, Amir Globerson, Ben Golan, Charles Kwong, Chris Alberti, Connie Tao, Eyal Ben-David, Gaurav Singh Tomar, Lukas Haas, Yonatan Bitton, Adam Bloniarz, Aijun Bai, Andrew Wang, Anfal Siddiqui, Arturo Bajuelos Castillo, Aviel Atias, Chang Liu, Corey Fry, Daniel Balle, Deepanway Ghosal, Doron Kukliansky, Dror Marcus, Elena Gribovskaya, Eran Ofek, Honglei Zhuang, Itay Laish, Jan Ackermann, Lily Wang, Meg Risdal, Megan Barnes, Michael Fink, Mohamed Amin, Moran Ambar, Natan Potikha, Nikita Gupta, Nitzan Katz, Noam Velan, Ofir Roval, Ori Ram, Polina Zablotskaia, Prathamesh Bang, Priyanka Agrawal, Rakesh Ghiya, Sanjay Ganapathy, Simon Baumgartner, Sofia Erell, Sushant Prakash, Thibault Sellam, Vikram Rao, Xuanhui Wang, Yaroslav Akulov, Yulong Yang, Zhen Yang, Zhixin Lai, Zhongru Wu, Anca Dragan, Avinatan Hassidim, Fernando Pereira, Slav Petrov, Srinivasan Venkatachary, Tulsee Doshi, Yossi Matias, Sasha Goldshtein, and Dipanjan Das. The FACTS leaderboard: A comprehensive benchmark for large language model factuality, 2025. URL https://arxiv.org/abs/2512.10791.

Ethan Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, and Pengfei Liu. FacTool: Factuality detection in generative AI – a tool augmented framework for multi-task and multi-domain scenarios. In *Second Conference on Language Modeling (COLM)*, 2025. URL https://openreview.net/forum?id=hJkQL9VtWT.

Farima Fatahi Bayat, Lechen Zhang, Sheza Munir, and Lu Wang. FactBench: A dynamic benchmark for in-the-wild language model factuality evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.acl-long.1587/.

Peizhou Huang, Zixuan Zhong, Zhongwei Wan, Donghao Zhou, Samiul Alam, Xin Wang, Zexin Li, Zhihao Dou, Li Zhu, Jing Xiong, Chaofan Tao, Yan Xu, Dimitrios Dimitriadis, Tuo Zhang, and Mi Zhang. MMDeepResearch-Bench: A benchmark for multimodal deep research agents. *arXiv preprint arXiv:2601.12346*, 2026. URL https://arxiv.org/abs/2601.12346.

Alon Jacovi, Andrew Wang, Chris Alberti, Connie Tao, Jon Lipovetz, Kate Olszewska, Lukas Haas, et al. The FACTS grounding leaderboard: Benchmarking LLMs' ability to ground responses to long-form input. *arXiv preprint arXiv:2501.03200*, 2025. URL https://arxiv.org/abs/2501.03200.

Nazanin Jafari, James Allan, and Mohit Iyyer. Beyond precision: Importance-aware recall for factuality evaluation in long-form LLM generation. *arXiv preprint arXiv:2604.03141*, 2026. URL https://arxiv.org/abs/2604.03141.

Minbyul Jeong, Hyeon Hwang, Chanwoong Yoon, Taewhoo Lee, and Jaewoo Kang. OLAPH: Improving factuality in biomedical long-form question answering. *arXiv preprint arXiv:2405.12701*, 2024. URL https://arxiv.org/abs/2405.12701.

Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanmin Wu, Jiayi Lei, Pengshuo Qiu, Pan Lu, Zehui Chen, Guanglu Song, Peng Gao, Yu Liu, Chunyuan Li, and Hongsheng Li. MMSearch: Benchmarking the potential of large models as multi-modal search engines. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://iclr.cc/virtual/2025/poster/30134.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing fine-grained evaluation capability in language models. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=8euJaTveKw.

Yukyung Lee, JoongHoon Kim, Jaehee Kim, Hyowon Cho, Jaewook Kang, Pilsung Kang, and Najoung Kim. CheckEval: A reliable LLM-as-a-judge framework for evaluating text generation using checklists. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.emnlp-main.796/.

Shilong Li, Xingyuan Bu, Wenjie Wang, Jiaheng Liu, Jun Dong, Haoyang He, Hao Lu, Haozhe Zhang, Chenchen Jing, Zhen Li, et al. MM-BrowseComp: A comprehensive benchmark for multimodal browsing agents. *arXiv preprint arXiv:2508.13186*, 2025. URL https://arxiv.org/abs/2508.13186.

Xin Liu, Lechen Zhang, Sheza Munir, Yiyang Gu, and Lu Wang. VeriFact: Enhancing long-form factuality evaluation with refined fact extraction and reference facts. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.emnlp-main.905/.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.emnlp-main.153/.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: a benchmark for general AI assistants. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=fibxvahvs3.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of EMNLP*, 2023. URL https://aclanthology.org/2023.emnlp-main.741/.

Delip Rao and Chris Callison-Burch. Autorubric: A unified framework for rubric-based LLM evaluation. *arXiv preprint arXiv:2603.00077*, 2026. URL https://arxiv.org/abs/2603.00077.

Jie Ruan, Inderjeet Nair, Shuyang Cao, Amy Liu, Sheza Munir, Micah Pollens-Dempsey, Tiffany Chiang, Lucy Kates, Nicholas David, Sihan Chen, Ruxin Yang, Yuqian Yang, Jasmine Gump, Tessa Bialek, Vivek Sankaran, Margo Schlanger, and Lu Wang. ExpertLongBench: Benchmarking language models on expert-level long-form generation tasks with structured checklists. *arXiv preprint arXiv:2506.01241*, 2025. URL https://arxiv.org/abs/2506.01241.

Sahel Sharifymoghaddam, Shivani Upadhyay, Nandan Thakur, Ronak Pradeep, and Jimmy Lin. Chatbot arena meets nuggets: Towards explanations and diagnostics in the evaluation of LLM responses. *arXiv preprint arXiv:2504.20006*, 2025. URL https://arxiv.org/abs/2504.20006.

Manasi Sharma, Chen Bo Calvin Zhang, Chaithanya Bandi, Clinton Wang, Ankit Aich, Huy Nghiem, Tahseen Rabbani, Ye Htet, Brian Jang, Sumana Basu, Aishwarya Balwani, Denis Peskoff, Marcos Ayestaran, Sean M. Hendryx, Brad Kenstler, and Bing Liu. ResearchRubrics: A benchmark of prompts and rubrics for evaluating deep research agents. *arXiv preprint arXiv:2511.07685*, 2025. URL https://arxiv.org/abs/2511.07685.

Yixiao Song, Yekyung Kim, and Mohit Iyyer. VeriScore: Evaluating the factuality of verifiable claims in long-form text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.findings-emnlp.552/.

Jiaqi Wang, Xiao Yang, Kai Sun, Parth Suresh, Sanat Sharma, Adam Czyzewski, Derek Andersen, Surya Appini, Arkav Banerjee, Sajal Choudhary, Rohit Patel, Wen-tau Yih, Xin Luna Dong, et al. CRAG-MM: Multi-modal multi-turn comprehensive RAG benchmark. *arXiv preprint arXiv:2510.26160*, 2025. URL https://arxiv.org/abs/2510.26160.

Miriam Wanner, Leif Azzopardi, Paul Thomas, Soham Dan, Benjamin Van Durme, and Nick Craswell. All claims are equal, but some claims are more equal than others: Importance-sensitive factuality evaluation of LLM generations. *arXiv preprint arXiv:2510.07083*, 2025. URL https://arxiv.org/abs/2510.07083.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. BrowseComp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025. URL https://arxiv.org/abs/2504.12516.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. Long-form factuality in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/937ae0e83eb08d2cb8627fe1def8c751-Abstract-Conference.html.

Fangda Ye, Yuxin Hu, Pengxiang Zhu, Yibo Li, Ziqi Jin, Yao Xiao, Yibo Wang, Lei Wang, Zhen Zhang, Lu Wang, et al. MiroEval: Benchmarking multimodal deep research agents in process and outcome. *arXiv preprint arXiv:2603.28407*, 2026. URL https://arxiv.org/abs/2603.28407.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. FLASK: Fine-grained language model evaluation based on alignment skill sets. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=CYmF38ysDa.

Danna Zheng, Mirella Lapata, and Jeff Z. Pan. Long-form information alignment evaluation beyond atomic facts. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.emnlp-main.558/. arXiv:2505.15792.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

A LIMITATIONS

GAMUT measures factual *completeness*, and this focus is also its main limitation. The GAMUT score rewards covering the facts in a rubric and penalizes contradicting them, but ignores content outside the rubric. It therefore does not measure precision: the hallucination rate over all claims a model makes. Precision is an important complementary axis, and combining the two is future work. The decompose-search-verify methods of Section 2 target precision, but they share the flaw we identify in recall work: they decompose an answer into a flat list of independent facts. We expect a structured meta-rubric like ours could improve precision evaluation in the same way.

Despite extensive effort, a rubric may not be exhaustive. This leaves an open evaluation question: how to treat a fact a response states that the rubric does not cover. Such a fact may be a valid one the rubric missed or a hallucination, and its meta-rubric structure is informative. A fact added beyond a *strict list*, which is meant to be complete, is more likely a hallucination than one added beyond a *flexible list*, which is open by design.

Our evaluation covers one question collection and its text-only variant, not other open-ended generation tasks. GAMUT is static, inherits CRAG-MM's entity and domain coverage, and is relatively small because its two-level rubrics are expensive to construct.

B PROMPTS

This appendix collects the full text of the prompts used to construct GAMUT.

B.1 QUESTION GENERATION

The prompt used to generate candidate questions for each image (Section 4.1).

Question Generation Prompt

```
You will be given an image and a user question. The question is what the user asks its
↪  smart glasses while looking at the image.
Your task is to generate 10 new questions that are more open-ended that require at least a
↪  few paragraphs to answer. Below are the detailed requirements. Each question MUST
↪  satisfy ALL of these requirements.

1) The question is challenging and calls for in-depth research. Pose questions that
↪  typically require multi-step web research and careful reasoning, often involving
↪  reconciling information across multiple sources.
2) The ideal answer is more than 1-2 paragraphs (no factoid question that can be answered
↪  in a few sentences).
3) Concise, natural info-seeking questions a real user would likely to ask its smart
↪  assistant. Avoid lengthy questions or questions with too many subquestions. Remember
↪  this is what the user sees through its glasses, so avoid mentioning the image
↪  explicitly such as "in this image".
4) The questions do not need to be related to the given question, but think of them as
↪  additional questions the same user asks. Do NOT assume the user to have any additional
↪  knowledge about the entity in the image in addition to the what's in the original
↪  question.
5) The questions must rely on the image **as much as possible**. The question should be
↪  unanswerable without seeing the actual image. Do NOT ask leading question or provide
↪  excessive textual hints or contexts in your question.
6) The questions must be answerable, and the answer must be unlikely to change over time.
7) The questions must be about the same entity as the given question. Do not ask about
↪  other entities in the image. Refer to the entity in the same way as the provided
↪  question. Do not name the entity if the original question refers to it using a pronoun
↪  or generic term.

# Examples:

## Image:
an image of Space Needle

### Good Question:
This tower looks really futuristic --- have there been any major redesigns and safety
↪  upgrades, especially how they've affected its earthquake and wind resistance today?

### Bad Question:
Have there been any major redesigns and safety upgrades for the Space Needle over the years?
#### Rationale:
```

```
It assumes the user knows about the entity. In addition, the question can be answered
↪  without the image.


## Image:
an image of a vehicle (Subaru WRX)

### Good Question:
I really like this car. Could you tell me the make and model, a detailed performance spec,
↪  and any common maintenance issues for a perspective buyer?

### Bad Question:
What's the pros and cons of the boxer engine used on this car?
#### Rationale:
The question assumes the user knows about the car and what type of engine it uses. The
↪  question is answerable without the image.

Now it's your turn.

Question:
{question}

Output one question per line. Do NOT include any other text or explanation. Do NOT include
↪  the original question in the output.
```

B.2 Question Selection

The prompt used to filter, rank, and diversify the accepted questions (Section 4.1).

The initial per-image generation makes two recurring errors this pass corrects. First, many candidates are *leading*: they presuppose the entity is already recognized, making the image decorative. A *stranger test* rejects any question a stranger could not ask from the image alone. For example, "how does this engine's turbocharger improve on the previous generation?" assumes the car is known, while "what kind of car is this and how does it perform?" does not. Second, generating questions independently per image makes them cluster into a few templates. The pass ranks the survivors by concreteness and specificity, and adds fresh candidates for images whose questions are all boilerplate.

Question Selection Prompt

```
You are selecting the best question for a multimodal deep research benchmark. Each example
↪  has an image (described by entity_name and category) and one or more candidate
↪  questions. Your job is to either pick the single BEST question or REJECT ALL candidates
↪  if none are good enough. Then, propose 5 new questions that are better than the
↪  candidates.

=== HARD REQUIREMENTS (ALL must be met -- failing ANY ONE means the question is invalid) ===


1. RESEARCH DEPTH: The question must be challenging and require searching the web to form
↪  an accurate answer. It cannot be answered in one or two sentences -- the ideal answer
↪  should be more than 1-2 paragraphs.

2. NOT LEADING OR OVERLY SPECIFIC: The question must not presuppose the correct answer or
↪  include specialized hints that only someone who already knows the entity's identity
↪  would know.

   **Stranger Test**: Imagine a person who sees the image but does NOT know the
   ↪  ground-truth identity of the entity. Could that person plausibly ask this exact
   ↪  question? If the question can only be formulated by someone who already knows what
   ↪  the entity is, it is LEADING and fails this requirement.

   Examples of LEADING questions (BAD -- fail the Stranger Test):
   - "How does this laptop's cooling architecture differ from its predecessor?" -> Only
   ↪  someone who already identified the laptop model AND knows its predecessor had a
   ↪  different cooling design would ask this.
   - "Based on this hood design, can you tell me more about this car's engine?" -> The
   ↪  phrase "based on this hood design" implies the asker knows the hood design is
   ↪  distinctive/relevant, which requires prior knowledge of the car.
   - "What role did this building play during the civil rights movement?" -> Only someone
   ↪  who already knows the building's historical significance would ask this.
   - "How does the fermentation process for this cheese differ from standard aging
   ↪  techniques?" -> Only someone who already identified the cheese type and knows it
   ↪  uses unusual fermentation would ask this.

   Examples of NON-LEADING questions (GOOD -- pass the Stranger Test):
```

```
   - "What laptop is this, and how does it compare to others in its class for gaming?" ->
   ↪  Anyone seeing a laptop can ask this.
   - "I notice this car has an unusual scoop on the hood -- what is it for?" -> Grounded in
   ↪  a VISIBLE feature; anyone can observe the scoop.
   - "What is this building and what is its historical significance?" -> A natural question
   ↪  anyone could ask upon seeing an interesting building.
   - "What type of cheese is this, and how is it made?" -> Anyone seeing an unfamiliar
   ↪  cheese could ask this.

   The key difference: non-leading questions are grounded in what is VISIBLE in the image
   ↪  (appearance, text, labels, distinctive visual features). Leading questions smuggle in
   ↪  knowledge that goes beyond what the image shows.

3. IMAGE DEPENDENCY: The question must depend on the image. It must use pronouns or generic
↪  terms (e.g., "this/these/that," "this vehicle/fruit/device") so the subject is resolved
↪  by looking at the image rather than being named or fully described in text.

4. UNAMBIGUOUS REFERENCE: The entity reference must not be genuinely ambiguous to a
↪  reasonable viewer. When the question uses an unqualified generic reference (e.g., "the
↪  car" / "this car"), it must refer to the main salient instance in the image.

Rejection rule: REJECT ALL candidates if EVERY candidate fails at least one hard
↪  requirement. Also reject if the question does not make sense given the image.

=== BEST SELECTION CRITERIA (for ranking valid candidates) ===

When multiple candidates pass all hard requirements, select the BEST using these
↪  preferences (in priority order):

A. QUALITY SWEET SPOT: Strongly prefer questions that simultaneously excel at being (i)
↪  succinct and natural, (ii) genuinely challenging / research-intensive, and (iii)
↪  dependent on the image. A short, natural question that demands deep research and relies
↪  on the image is ideal.

B. INSTANCE-SPECIFICITY (diversity preference): Prefer questions that are specific to THIS
↪  particular entity -- questions that could NOT be asked about any other object in the
↪  same category. A question is "boilerplate" if you could swap the entity for any other
↪  in the same category and the question would still work identically.

   Known boilerplate patterns to DEPRIORITIZE for this category:
   {boilerplate_patterns}

   If the ONLY valid candidate is a boilerplate question, you may still select it -- but
   ↪  flag it as IS_BOILERPLATE: YES.

C. ORIGINALITY: Among equally instance-specific questions, prefer ones with a unique angle
↪  or asking about something surprising rather than the most obvious question.

=== TASK 2: PROPOSE 5 NEW QUESTIONS ===

After selection, propose 5 NEW questions for this entity/image. These must:
- Satisfy ALL 4 hard requirements above
- Avoid the boilerplate patterns listed above for this category
- Be as DIVERSE as possible from each other (different themes, angles, aspects)
- Be as DIVERSE as possible from the existing candidate questions
- Aim for questions you would rank ABOVE all the given candidates (more natural, more
↪  challenging, more instance-specific)
- Reference the entity using image-dependent terms (e.g., "this plant," "this building") --
↪  never by name

CRITICAL -- Stranger Test for new questions:
You know the entity's identity (given in "Entity in image"), but you must NOT let that
↪  knowledge leak into the questions as hidden hints. Every proposed question must pass
↪  the Stranger Test: it must be something a person could ask purely from LOOKING AT the
↪  image, without knowing what the entity is. Ground your questions in visible features
↪  (shape, color, text, labels, design elements, setting, context clues) -- not in facts
↪  you know about the entity.

   BAD (leaks entity knowledge): "How did this artist's mental illness influence the
   ↪  swirling brushwork seen here?" -> Only someone who already identified the painting
   ↪  and knows the artist's biography would connect mental illness to the visual style. A
   ↪  stranger sees swirling brushwork but has no reason to mention mental illness.
   GOOD (grounded in visual): "I notice the very distinctive swirling brushwork in this
   ↪  painting -- who is the artist, and what inspired this style?" -> Anyone seeing the
   ↪  painting can observe the brushwork and ask openly.

Context:
- Category: {category}
- Entity in image: {entity_name}
- Query type: {query_category}
```

```
Candidate Questions:
{questions_formatted}

First, check each candidate against the 4 hard requirements. Then, among valid candidates,
↪  select the best using criteria A-C. Finally, propose 5 new questions.

Output ONLY a single raw JSON object. No markdown fences, no comments, no extra text before
↪  or after the JSON. The JSON must have exactly these fields:
{{"reasoning": "<1-3 sentences: which candidates pass/fail requirements, and why the
↪  selected one is best>", "selected_question_id": "<question_id of the best question, or
↪  null if rejecting all>", "selected_question_text": "<full text of the selected
↪  question, or null if rejecting all>", "is_boilerplate": <true if the selected question
↪  is boilerplate, false otherwise, null if rejected>, "new_questions": ["<first proposed
↪  question>", "<second>", "<third>", "<fourth>", "<fifth>"]}}
```

B.3 Rubric Generation

The prompt used to generate the initial meta-rubric and binary rubric for each question (Section 4.2).

Rubric Generation Prompt

```
You will receive: (1) an image, (2) a user question about the image, and (3) the
↪  ground-truth entity name in the image.

Your task is to design an evaluation rubric--a compact set of elements ("ingredients") that
↪  a high-quality final answer should satisfy, mapped to relevant web snippets.

You will work in THREE phases. Each phase is a separate top-level key in your JSON
↪  response. You MUST complete each phase fully before starting the next.

================================
PHASE 1: KNOWLEDGE GATHERING
================================
In this phase, you will focus EXCLUSIVELY on gathering the factual information required to
↪  answer the question perfectly. You MUST actively use your web-browsing capabilities.

A. TARGET ALIGNMENT
   - State the specific subject targeted by the user's question (this may be the main focus
   ↪  of the image or a specific secondary object).
   - State what entity the provided Ground Truth name refers to.
   - Note: The Ground Truth Entity is *intended* to help ground your web searches. However,
   ↪  mismatches can occur (e.g., GT is the author, but the question is about the book).
   ↪  You MUST prioritize the question's actual target as your strict anchor for searches.

B. KNOWLEDGE SEARCH & SNIPPET CREATION
   - Issue web searches to find all critical, valuable, and contextual information needed
   ↪  to formulate a PERFECT answer to the user's specific question.
   - Create a pool of Snippets from your search results. For each snippet, you MUST provide
   ↪  the exact URL, the Page Title, and extract an exact substring (Text) from the page
   ↪  that contains the fact. The Text **should not exceed 40 words**.
   - Assign them unique IDs (e.g., "s1", "s2"). Prefer authoritative sources (Wikipedia,
   ↪  official sites).

================================
PHASE 2: EVALUATION BLUEPRINT (META-RUBRICS)
================================
Before writing binary rubrics, map out the required knowledge into higher-level
↪  "Meta-Rubrics."

A. META-RUBRICS & TARGET TYPES
Break down the user's question to identify all distinct parts of a perfect answer. Design
↪  as many Meta-Rubrics as necessary to cover this knowledge. A single part of the
↪  question may require multiple Meta-Rubrics of different Target Types.

For each Meta-Rubric, you must write a `description` explaining its specific focus, assign
↪  an overall `importance` (Critical, Valuable, Context), assign ONE of the following
↪  Target Types, and extract the relevant Knowledge Components (citing your Phase 1
↪  snippet IDs).

You MUST strictly adhere to these definitions and field mappings. Leave arrays for
↪  unselected types empty `[]`:
- **Simple Knowledge**: Discrete facts. (Populate `simple_fact`).
- **Strict List**: A finite set of items where *every single item* is individually
↪  required. (Populate `strict_list_items`).
- **Flexible List**: A pool of valid options where no specific item is strictly required,
↪  but a certain volume/number of items is needed. (Populate `flexible_list_items` and
↪  `threshold_tiers`).
```

```
  * **Mixed List Rule**: If a concept has both mandatory core items and a pool of valid
  ↪  secondary options, you MUST split it into TWO separate Meta-Rubrics: one Strict List
  ↪  for the core, and one Flexible List for the options.
- **Process**: A chronological sequence of steps. (Populate `ordered_steps`, flagging
↪  absolute milestones with `is_mandatory: true` and minor/optional steps with
↪  `is_mandatory: false`).
- **Relationship**: Evaluating how things connect. (Populate `entities` and
↪  `relationship_nature`).

B. HANDLING FLEXIBLE LISTS & THRESHOLDS
When dealing with a **Flexible List**, you must explicitly plan for both "Baseline" and
↪  "Bonus" partial credit using SHORT-HAND notation:
- **Baseline Thresholds (Always Required)**: You must design 1-2 baseline threshold tiers
↪  representing the minimum acceptable recall (e.g., ">= 1", ">= 3"). These will inherit
↪  the Flexible List's main category.
- **Short Flexible Lists (<= 7 items AND Meta-Rubric is Critical/Valuable)**: For the
↪  remaining bonus credit, map each list item to its own individual rubric element. Place
↪  these in a lower category than the Meta-Rubric. Do NOT use upper-tier thresholds.
- **Long Flexible Lists (> 7 items) OR Context Meta-Rubrics**: Do NOT use individual
↪  elements to prevent rubric bloat. Instead, design AT LEAST 2-3 Upper-Tier thresholds
↪  beyond the baselines. These must be placed in a lower category (unless the Meta-Rubric
↪  is Context). Crucially, the final upper-tier threshold MUST represent a perfect
↪  answer--usually requiring ALL items, unless the user's explicit question caps the
↪  requirement.
- Populate your tiers in the `threshold_tiers` array using strict short-hand (e.g.,
↪  `["Critical: >= 2", "Valuable: >= 5", "Valuable: ALL 12"]` or just baselines for short
↪  lists).

================================
PHASE 3: FINAL RUBRIC (Final JSON Output)
================================
Now, translate your Meta-Rubrics into strict, testable binary rubric elements.

A. TRANSLATING META-RUBRICS TO ELEMENTS
- **Simple Knowledge**: Map `simple_fact` to a single binary element.
- **Strict List**: Map EACH item to its own DEDICATED individual element. They inherit the
↪  Meta-Rubric's exact category.
- **Flexible List**:
  * Create your Baseline Threshold elements based on `threshold_tiers`. **CRITICAL RULE:
  ↪  While your blueprint used short-hand, the final threshold elements MUST fully expand
  ↪  and exhaustively list ALL acceptable options directly in the text** (e.g., "Identify
  ↪  at least 2 of: A, B, C, D..."). Do NOT use "etc." or cross-references.
  * **If Short Flexible List**: Map EACH list item to its own individual element in a lower
  ↪  category.
  * **If Long Flexible List / Context Meta-Rubric**: Create your Upper-Tier Threshold
  ↪  elements (in a lower category), expanding them to exhaustively list all options in
  ↪  the text.
- **Process**:
  * **Presence**: Map EACH step to its own individual element. Steps marked `is_mandatory:
  ↪  true` inherit the Meta-Rubric's category. Steps marked `is_mandatory: false` drop to a
  ↪  lower category. **Do NOT use thresholds for processes.**
  * **Sequence (Mandatory)**: Draft 1 element evaluating the relative order of the
  ↪  mandatory steps. **CRITICAL RULE: You MUST use this exact phrasing format:**
  ↪  "Evaluate the relative order of the matching mandatory steps against this master
  ↪  list: 1. [Step A], 2. [Step B], 3. [Step C]. Missing steps or extra unlisted steps do
  ↪  not fail this element. It passes as long as the included matching steps appear in the
  ↪  correct chronological sequence without reversals."
  * **Sequence (Overall)**: Draft 1 element evaluating the perfect chronological flow. Use
  ↪  the exact same phrasing format as above, but embed the full numbered master list of
  ↪  ALL steps (mandatory + optional).
- **Relationship**: Break into atomic elements (identify entity, and separate elements for
↪  aspects of the `relationship_nature`).

B. ASSIGNING CATEGORIES (CASCADING LOGIC)
The final category (Answer_Critical, Valuable, Context) of each individual rubric element
↪  depends on its parent Meta-Rubric's overall `importance`:
- If a Meta-Rubric is overall "Critical":
  * Strict List items, Mandatory Process steps/sequences, and **Baseline Thresholds** are
  ↪  `Answer_Critical`.
  * Optional Process steps, individual optional items (Short Flexible Lists), and
  ↪  **Upper-Tier Thresholds** (Long Flexible Lists) strictly drop to `Valuable`.
- If a Meta-Rubric is overall "Valuable":
  * Strict items, Mandatory steps, and Baselines are `Valuable`. Optional items/Upper-Tiers
  ↪  drop to `Context`.
- If a Meta-Rubric is overall "Context":
  * All resulting elements (Mandatory, Baselines, and Upper-Tiers) are `Context`. NONE can
  ↪  be Answer_Critical.

--------------------------------
OUTPUT FORMAT
--------------------------------
```

```
Return a single JSON object with EXACTLY these top-level keys (no markdown fences, no
↪  comments, no extra text):
{{
  "Phase_1_Knowledge_Gathering": {{
    "Target_Alignment": {{
      "targeted_subject": "<string>",
      "ground_truth": "<string>",
      "alignment_note": "<string>"
    }},
    "Search_Strategy": "<string>",
    "Draft_Snippets": [
      {{ "id": "<string>", "URL": "<string>", "Title": "<string>", "Text": "<string - no
      ↪  more than 40 words>" }}
    ]
  }},
  "Phase_2_Evaluation_Blueprint": {{
    "Meta_Rubrics": [
      {{
        "description": "<string>",
        "target_type": "<enum: Simple Knowledge | Strict List | Flexible List | Process |
        ↪  Relationship>",
        "importance": "<enum: Critical | Valuable | Context>",
        "Knowledge_Components": {{
          "simple_fact": [ {{ "text": "<string>", "citations": ["<snippet_id>"] }} ],
          "strict_list_items": [ {{ "item_text": "<string>", "citations": ["<snippet_id>"]
          ↪  }} ],
          "flexible_list_items": [ {{ "item_text": "<string>", "citations":
          ↪  ["<snippet_id>"] }} ],
          "ordered_steps": [ {{ "step_text": "<string>", "is_mandatory": true, "citations":
          ↪  ["<snippet_id>"] }} ],
          "entities": [ {{ "text": "<string>", "citations": ["<snippet_id>"] }} ],
          "relationship_nature": [ {{ "text": "<string>", "citations": ["<snippet_id>"] }}
          ↪  ],
          "meta_insights": [ {{ "text": "<string>", "citations": ["<snippet_id>"] }} ]
        }},
        "threshold_tiers": [ "<string - USE SHORT-HAND for Flexible Lists. e.g., 'Critical:
        ↪  >= 2', 'Valuable: >= 8', 'Valuable: ALL 15'>" ],
        "drafting_thoughts": "<string -- explain translation to Phase 3 via cascading
        ↪  logic.>"
      }}
    ],
    "Citation_Completeness_Check": "<string>"
  }},
  "Phase_3_Final_Rubric": {{
    "Question": "<string>",
    "Snippets": [
      {{ "id": "<string>", "URL": "<string>", "Title": "<string>", "Text": "<string>" }}
    ],
    "Answer_Critical": [
      {{ "Ingredient": "<string>", "Handle": "<string>", "Specifics": [ {{ "Text":
      ↪  "<string>", "Citation": "<snippet_id>" }} ] }}
    ],
    "Valuable": [
      {{ "Ingredient": "<string>", "Handle": "<string>", "Specifics": [ {{ "Text":
      ↪  "<string>", "Citation": "<snippet_id>" }} ] }}
    ],
    "Context": [
      {{ "Ingredient": "<string>", "Handle": "<string>", "Specifics": [ {{ "Text":
      ↪  "<string>", "Citation": "<snippet_id>" }} ] }}
    ]
  }}
}}

--------------------------------
ELEMENT REQUIREMENTS
--------------------------------
Each element is one checklist item the final answer should satisfy:
- **Ingredient**: Start with a strict, verifiable command verb (e.g., "State", "Specify",
↪  "Identify", "Cite", "Mention", "Warn", "Quantify").
  - Evaluate exactly one indivisible concept (unless it is a Threshold Element or a
  ↪  Sequence Element).
  - Formatted as strict binary conditions (pass/fail).
  - **Self-Contained**: The element MUST be fully understandable and gradable in isolation.
  ↪  Do not reference other elements or external context (e.g., avoid "from the list
  ↪  above" or "the aforementioned").
  - DO NOT use subjective verbs like "Explain," "Describe," or "Discuss."
  - Specify exact numerical thresholds, ranges, names, dates, or descriptions where
  ↪  applicable.
```

```
- **THRESHOLD ELEMENTS**: When grading Flexible Lists via Baseline or Upper-Tier
↪  Thresholds, YOU MUST create distinct aggregate threshold elements based on your
↪  `threshold_tiers` array. **These elements MUST fully expand the short-hand to
↪  exhaustively list ALL acceptable options directly in the text** (e.g., "Identify at
↪  least 4 of the following 12 secondary uses: A, B, C, D..."). DO NOT use shortcuts like
↪  "e.g." or "etc." The evaluator must have the complete universe of acceptable answers
↪  within this single sentence.
- **Handle**: Short (2-5 words, Title Case), general label that abstracts the ingredient.
- **Specifics**: Zero or more exact quotes from snippets:
  - "Text": an exact substring from a snippet's Text field (preserve punctuation/casing; do
  ↪  not paraphrase).
  - "Citation": the snippet's id exactly as it appears in the Snippets array.

---------------------------------
CATEGORY DEFINITIONS
---------------------------------
- **Answer_Critical**: Minimal essentials that directly determine correctness/completeness.
↪  Missing any one makes the answer effectively wrong or incomplete.
- **Valuable**: Substantial supports that meaningfully improve the depth, precision, or
↪  usefulness of the answer (e.g., specific examples, distinguishing features, practical
↪  implications, or expert-level elaborations).
- **Context**: Helpful background that aids understanding but is not required for
↪  correctness (e.g., basic definitions, general geographic origins, or orienting
↪  details).

---------------------------------
TARGET BUDGET (FLEXIBLE)
---------------------------------
- Aim for ~10 valid Snippets.
- Aim for AT LEAST ~5 Answer_Critical elements.
- Aim for AT LEAST 1-2 Valuable elements.
*Note: These are guidelines, not hard constraints. The strict rule that "all snippets must
↪  be cited" takes absolute priority.*

---------------------------------
RULES
---------------------------------
1) Complete Phase 1 entirely before Phase 2, and Phase 2 before Phase 3.
2) **Exactness**: Quotes in Phase 3 Specifics must be exact substrings from snippet text.
↪  No edits to punctuation, casing, or numbers.
3) **No Orphan Snippets**: Every snippet in the final Phase 3 output MUST be cited by at
↪  least one rubric element in its `Specifics` array.
4) **Unsupported essentials (rare)**: If an ingredient is clearly essential but no snippet
↪  supports it, include it with "Specifics": [] -- use sparingly and only with high
↪  confidence.
5) **Mandatory Entity Element**: You MUST include at least one Answer_Critical element that
↪  requires identifying the specific subject targeted by the question.
6) Raw JSON only. Your entire response must be parseable JSON -- no markdown fences,
↪  comments, or trailing commas.

---------------------------------
INPUT
---------------------------------
- Question: {question}
- Ground Truth Entity Name: {entity_name}
```

B.4 RUBRIC REFINEMENT

The prompt used for LLM self-refinement of the draft rubric (Section 4.2).

Rubric Refinement Prompt

```
You will receive: (1) an image, (2) a user question about the image, (3) the ground-truth
↪  entity name, and (4) a DRAFT rubric (JSON) generated by another model.

Your task is to review, restructure, and refine the draft rubric by fixing factual errors,
↪  filling knowledge gaps, applying strict threshold grading curves, and enforcing atomic,
↪  self-contained formatting.

You will work in FOUR phases. Each phase is a separate top-level key in your JSON response.
↪  You MUST complete each phase fully before starting the next.

================================
PHASE 1: DRAFT AUDIT (Snippets & Rubrics)
================================
In this phase, you will review the draft to verify its facts and understand its intended
↪  coverage, specifically looking for gaps and incomplete lists/processes.
```

```
A. TARGET ALIGNMENT
   - State the specific subject targeted by the user's question.
   - State what entity the provided Ground Truth name refers to.
   - Note: The Ground Truth Entity is *intended* to help ground your web searches. However,
   ↪  mismatches can occur. You MUST prioritize the question's actual target as your
   ↪  strict anchor.

B. SNIPPET AUDIT
   For EACH snippet in the draft, actively use your browsing tool to verify it.
   - Determine if the snippet is valid (URL exists, correctly matches the targeted subject,
   ↪  and quoted text is exactly matched).
   - Evaluate snippets that are NOT cited by the draft rubric. Do they contain missing
   ↪  information, or are they irrelevant "hard negatives"?
   - Ensure the Text field is an EXACT quote (max 40 words).
   - Decide the Action: KEEP, DELETE (if invalid/irrelevant), EDIT (to fix quotes), or
   ↪  SPLIT.

C. DRAFT RUBRIC AUDIT
   Review each draft rubric element for correctness, relevance, and formatting. Crucially,
   ↪  use the draft for inspiration to find missing siblings:
   - **Completeness Flagging**: If a draft rubric tests a single item from a broader
   ↪  category or a single step from a larger procedure (e.g., "Mention Affiliation A" or
   ↪  "State Step 2"), you MUST flag this as a partial list/process. You will use this
   ↪  flag in Phase 2 to search for the *exhaustive* list or complete sequence.
   - Determine if the rubric is factually correct, properly categorized, and formatted as a
   ↪  strict, self-contained binary check.

================================
PHASE 2: INDEPENDENT SEARCH & KNOWLEDGE EXPANSION
================================
To avoid "anchoring bias" on a flawed draft, you must now independently search for missing
↪  knowledge.

A. SEARCH STRATEGY
   - Formulate a search strategy to answer the question from scratch, ensuring you cover
   ↪  all aspects of a PERFECT answer.
   - Explicitly target the "Completeness Flags" raised in Phase 1C. If the draft had an
   ↪  incomplete list or sequence, you MUST search for the comprehensive, exhaustive set
   ↪  of items or steps.

B. NEW SNIPPET CREATION
   - Issue web searches to find this missing information and ADD new snippets.
   - Assign them new, unique IDs (e.g., "s_new_1", "s_new_2").

================================
PHASE 3: EVALUATION BLUEPRINT (META-RUBRICS)
================================
Now, synthesize your verified knowledge into a structural blueprint. Your blueprint MUST
↪  act as a correct, verified superset: encompassing the valid ideas from the draft rubric
↪  PLUS your newly discovered exhaustive knowledge.

A. META-RUBRICS & TARGET TYPES
Break down the complete required knowledge into distinct parts. Design as many Meta-Rubrics
↪  as necessary. For each Meta-Rubric, write a `description` explaining its focus, assign
↪  an overall `importance` (Critical, Valuable, Context), assign ONE Target Type, and
↪  extract the Knowledge Components (citing your Phase 1/Phase 2 snippet IDs).

You MUST strictly adhere to these definitions and field mappings. Leave arrays for
↪  unselected types empty `[]`:
- **Simple Knowledge**: Discrete facts. (Populate `simple_fact`).
- **Completeness Resolution**: If you resolved a Completeness Flag from Phase 1 for a set
↪  of valid entities, options, or sequential steps, you MUST use the `Strict List`,
↪  `Flexible List` or `Process` Target Type to exhaustively enumerate all of them. Do NOT
↪  collapse them into a generic `Simple Knowledge` wildcard.
- **Homogeneous Lists Only**: A Strict or Flexible List MUST only contain items of the
↪  exact same conceptual class (e.g., all ingredients, all locations, or all dates). **Do
↪  NOT merge distinct conceptual dimensions** (e.g., mixing geographical origins with
↪  historical dates) into a single "catch-all" list. If a question requires multiple
↪  distinct dimensions of knowledge, you must create separate Meta-Rubrics for each, or
↪  map standalone facts to `Simple Knowledge`.
- **Strict List**: A finite set of items where *every single item* is individually
↪  required. (Populate `strict_list_items`).
- **Flexible List**: A pool of valid options where no specific item is strictly required,
↪  but a certain volume/number of items is needed. (Populate `flexible_list_items` and
↪  `threshold_tiers`).
  * **Mixed List Rule**: If a concept has both mandatory core items and a pool of valid
  ↪  secondary options, you MUST split it into TWO separate Meta-Rubrics: one Strict List
  ↪  for the core, and one Flexible List for the options.
- **Process**: A chronological sequence of steps. (Populate `ordered_steps`, flagging
↪  absolute milestones with `is_mandatory: true` and minor/optional steps with
↪  `is_mandatory: false`).
```

```
- **Relationship**: Evaluating how things connect. (Populate `entities` and
↪  `relationship_nature`).

B. HANDLING FLEXIBLE LISTS & THRESHOLDS
When dealing with a **Flexible List**, you must explicitly plan for both "Baseline" and
↪  "Bonus" partial credit using SHORT-HAND notation:
- **Baseline Thresholds (Always Required)**: You must design 1-2 baseline threshold tiers
↪  representing the minimum acceptable recall (e.g., ">= 1", ">= 3"). These will inherit
↪  the Flexible List's main category.
- **Short Flexible Lists (<= 7 items AND Meta-Rubric is Critical/Valuable)**: For the
↪  remaining bonus credit, map each list item to its own individual rubric element. Place
↪  these in a lower category than the Meta-Rubric. Do NOT use upper-tier thresholds.
- **Long Flexible Lists (> 7 items) OR Context Meta-Rubrics**: Do NOT use individual
↪  elements to prevent rubric bloat. Instead, design AT LEAST 2-3 Upper-Tier thresholds
↪  beyond the baselines. These must be placed in a lower category (unless the Meta-Rubric
↪  is Context). Crucially, the final upper-tier threshold MUST represent a perfect
↪  answer--usually requiring ALL items, unless the user's explicit question caps the
↪  requirement.
- Populate your tiers in the `threshold_tiers` array using strict short-hand (e.g.,
↪  `["Critical: >= 2", "Valuable: >= 8", "Valuable: ALL 15"]` or just baselines for short
↪  lists).

================================
PHASE 4: FINAL REFINED RUBRIC (JSON Output)
================================
Do not attempt to output a 1:1 mapping of the old draft elements. Instead, translate your
↪  Phase 3 Meta-Rubrics directly into the final, perfect rubric elements.

A. TRANSLATING META-RUBRICS TO ELEMENTS
- **Simple Knowledge**: Map `simple_fact` to a single binary element.
- **Strict List**: Map EACH item to its own DEDICATED individual element. They inherit the
↪  Meta-Rubric's exact category.
- **Flexible List**:
  * Create your Baseline Threshold elements based on `threshold_tiers`. **CRITICAL RULE:
  ↪  While your blueprint used short-hand, the final threshold elements MUST fully expand
  ↪  and exhaustively list ALL acceptable options directly in the text** (e.g., "Identify
  ↪  at least 2 of: A, B, C, D..."). Do NOT use "etc." or cross-references.
  * **If Short Flexible List**: Map EACH list item to its own individual element in a lower
  ↪  category.
  * **If Long Flexible List / Context Meta-Rubric**: Create your Upper-Tier Threshold
  ↪  elements (in a lower category), expanding them to exhaustively list all options in
  ↪  the text.
- **Process**:
  * **Presence**: Map EACH step to its own individual element. Steps marked `is_mandatory:
  ↪  true` inherit the Meta-Rubric's category. Steps marked `is_mandatory: false` drop to a
  ↪  lower category. **Do NOT use thresholds for processes.**
  * **Sequence (Mandatory)**: Draft 1 element evaluating the relative order of the
  ↪  mandatory steps. **CRITICAL RULE: You MUST use this exact phrasing format:**
  ↪  "Evaluate the relative order of the matching mandatory steps against this master
  ↪  list: 1. [Step A], 2. [Step B], 3. [Step C]. Missing steps or extra unlisted steps do
  ↪  not fail this element. It passes as long as the included matching steps appear in the
  ↪  correct chronological sequence without reversals."
  * **Sequence (Overall)**: Draft 1 element evaluating the perfect chronological flow. Use
  ↪  the exact same phrasing format as above, but embed the full numbered master list of
  ↪  ALL steps (mandatory + optional).
- **Relationship**: Break into atomic elements (identify entity, and separate elements for
↪  aspects of the `relationship_nature`).

B. ASSIGNING CATEGORIES (CASCADING LOGIC)
The final category (Answer_Critical, Valuable, Context) of each individual rubric element
↪  depends on its parent Meta-Rubric's overall `importance`:
- If a Meta-Rubric is overall "Critical":
  * Strict List items, Mandatory Process steps/sequences, and **Baseline Thresholds** are
  ↪  `Answer_Critical`.
  * Optional Process steps, individual optional items (Short Flexible Lists), and
  ↪  **Upper-Tier Thresholds** (Long Flexible Lists) strictly drop to `Valuable`.
- If a Meta-Rubric is overall "Valuable":
  * Strict items, Mandatory steps, and Baselines are `Valuable`. Optional items/Upper-Tiers
  ↪  drop to `Context`.
- If a Meta-Rubric is overall "Context":
  * All resulting elements (Mandatory, Baselines, Upper-Tiers) are `Context`. NONE can be
  ↪  Answer_Critical or Valuable.

--------------------------------
OUTPUT FORMAT
--------------------------------
Return a single JSON object with EXACTLY these top-level keys (no markdown fences, no
↪  comments, no extra text):
{{
  "Phase_1_Draft_Audit": {{
    "Target_Alignment": {{
```

```
      "targeted_subject": "<string>",
      "ground_truth": "<string>",
      "alignment_note": "<string>"
    }},
    "Snippet_Audit": [
      {{ "id": "<string>", "finding": "<string>", "action": "KEEP | DELETE | EDIT | SPLIT"
      ↪  }}
    ],
    "Draft_Rubric_Audit": [
      {{ "handle": "<string>", "evaluation": "<string>", "completeness_flag": "<string or
      ↪  None>" }}
    ]
  }},
  "Phase_2_Independent_Search": {{
    "Search_Strategy": "<string>",
    "New_Snippets_Added": [
      {{ "id": "<string>", "reason_added": "<string>", "URL": "<string>", "Title":
      ↪  "<string>", "Text": "<string>" }}
    ]
  }},
  "Phase_3_Evaluation_Blueprint": {{
    "Meta_Rubrics": [
      {{
        "description": "<string>",
        "target_type": "<enum: Simple Knowledge | Strict List | Flexible List | Process |
        ↪  Relationship>",
        "importance": "<enum: Critical | Valuable | Context>",
        "Knowledge_Components": {{
          "simple_fact": [ {{ "text": "<string>", "citations": ["<snippet_id>"] }} ],
          "strict_list_items": [ {{ "item_text": "<string>", "citations": ["<snippet_id>"]
          ↪  }} ],
          "flexible_list_items": [ {{ "item_text": "<string>", "citations":
          ↪  ["<snippet_id>"] }} ],
          "ordered_steps": [ {{ "step_text": "<string>", "is_mandatory": true, "citations":
          ↪  ["<snippet_id>"] }} ],
          "entities": [ {{ "text": "<string>", "citations": ["<snippet_id>"] }} ],
          "relationship_nature": [ {{ "text": "<string>", "citations": ["<snippet_id>"] }}
          ↪  ],
          "meta_insights": [ {{ "text": "<string>", "citations": ["<snippet_id>"] }} ]
        }},
        "threshold_tiers": [ "<string - USE SHORT-HAND for Flexible Lists. e.g., 'Critical:
        ↪  >= 2', 'Valuable: >= 8', 'Valuable: ALL 15'>" ],
        "drafting_thoughts": "<string -- explain translation to Phase 4 via cascading
        ↪  logic.>"
      }}
    ],
    "Citation_Completeness_Check": "<string>"
  }},
  "Phase_4_Final_Refined_Rubric": {{
    "Question": "<string>",
    "Snippets": [
      {{ "id": "<string>", "URL": "<string>", "Title": "<string>", "Text": "<string>" }}
    ],
    "Answer_Critical": [
      {{ "Ingredient": "<string>", "Handle": "<string>", "Specifics": [ {{ "Text":
      ↪  "<string>", "Citation": "<snippet_id>" }} ] }}
    ],
    "Valuable": [
      {{ "Ingredient": "<string>", "Handle": "<string>", "Specifics": [ {{ "Text":
      ↪  "<string>", "Citation": "<snippet_id>" }} ] }}
    ],
    "Context": [
      {{ "Ingredient": "<string>", "Handle": "<string>", "Specifics": [ {{ "Text":
      ↪  "<string>", "Citation": "<snippet_id>" }} ] }}
    ]
  }}
}}

--------------------------------
ELEMENT REQUIREMENTS
--------------------------------
Each element is one checklist item the final answer should satisfy:
- **Ingredient**: Start with a strict, verifiable command verb (e.g., "State", "Specify",
↪  "Identify", "Cite", "Mention", "Warn", "Quantify").
  - Evaluate exactly one indivisible concept (unless it is a Threshold Element or a
  ↪  Sequence Element).
  - Formatted as strict binary conditions (pass/fail).
  - **Self-Contained**: The element MUST be fully understandable and gradable in isolation.
  ↪  Do not reference other elements or external context (e.g., avoid "from the list
  ↪  above" or "the aforementioned").
  - DO NOT use subjective verbs like "Explain," "Describe," or "Discuss."
```

```
  - Specify exact numerical thresholds, ranges, names, dates, or descriptions where
  ↪  applicable.
- **THRESHOLD ELEMENTS**: When grading Flexible Lists via Baseline or Upper-Tier
↪  Thresholds, YOU MUST create distinct aggregate threshold elements based on your
↪  `threshold_tiers` array. **These elements MUST fully expand the short-hand to
↪  exhaustively list ALL acceptable options directly in the text** (e.g., "Identify at
↪  least 4 of the following 12 secondary uses: A, B, C, D..."). DO NOT use shortcuts like
↪  "e.g." or "etc." The evaluator must have the complete universe of acceptable answers
↪  within this single sentence.
- **Handle**: Short (2-5 words, Title Case), general label that abstracts the ingredient.
- **Specifics**: Zero or more exact quotes from snippets:
  - "Text": an exact substring from a snippet's Text field (preserve punctuation/casing; do
  ↪  not paraphrase).
  - "Citation": the snippet's id exactly as it appears in the Snippets array.

---------------------------------
CATEGORY DEFINITIONS
---------------------------------
- **Answer_Critical**: Minimal essentials that directly determine correctness/completeness.
↪  Missing any one makes the answer effectively wrong or incomplete.
- **Valuable**: Substantial supports that meaningfully improve the depth, precision, or
↪  usefulness of the answer (e.g., specific examples, distinguishing features, practical
↪  implications, or expert-level elaborations).
- **Context**: Helpful background that aids understanding but is not required for
↪  correctness (e.g., basic definitions, general geographic origins, or orienting
↪  details).

---------------------------------
RULES
---------------------------------
1) Complete each phase sequentially.
2) **Do not renumber existing Snippet IDs.** Preserve the original IDs exactly as provided.
↪  Assign distinct IDs to new snippets and use those exact IDs in your Citations.
3) **Mandatory Tool Use for Verification**: You MUST actively use your browsing tool to
↪  navigate to URLs to verify quotes, validate sources, and search for missing knowledge.
4) **No Orphan Snippets**: Every snippet in the final Phase 4 output MUST be cited by at
↪  least one rubric element in its `Specifics` array.
5) All Specifics "Text" values must be exact substrings from the cited snippet's Text
↪  field. No paraphrasing, no punctuation/casing changes.
6) Raw JSON only. Your entire response must be parseable JSON -- no markdown fences,
↪  comments, or trailing commas.

---------------------------------
INPUT
---------------------------------
- Question: {question}
- Ground Truth Entity Name: {entity_name}
- Draft Rubric:
{draft_rubric}
```

B.5 LLM JUDGE

The system prompt and evaluation template used by the LLM judge to grade each binary rubric check (Section 3.3).

LLM Judge Prompt

```
SYSTEM PROMPT
=============
You are a rigorous and objective evaluator. Your task is to judge whether an answer
↪  adequately addresses specific rubric requirements.

You must evaluate holistically, not mechanistically. Your goal is to assess whether the
↪  answer would be genuinely useful and accurate to a reader asking this question -- not
↪  to perform literal string matching against rubric text.

You must evaluate each rubric element independently and provide a structured verdict.


EVALUATION PROMPT
=================
## Question
{question}

## Rubric Requirements
The following rubric elements define what a high-quality answer should contain. Each
↪  element has an "Ingredient" (the requirement) and optionally "Specifics" (supporting
↪  citations from reference materials).
```

```
{formatted_rubrics}

## Answer to Evaluate
{llm_answer}

## Task
For EACH rubric element listed above, determine whether the answer:
- **meets**: The answer substantively satisfies this rubric requirement. The relevant
↪  information is present and correct.
- **partially_meets**: The answer addresses this rubric requirement but with insufficient
↪  specificity, vague approximation, or only partial coverage. The reader gets
↪  directionally correct information but not the full or precise detail the rubric
↪  requires.
- **missing**: The answer does not address this rubric requirement at all. The relevant
↪  information is entirely absent.
- **contradicts**: The answer contains information that MATERIALLY contradicts this rubric
↪  requirement in a way that would mislead the reader.

**Guidance and examples:**

"meets" vs "partially_meets":
- Rubric requires species "Daviesia longifolia", answer says "Daviesia longifolia" ->
↪  meets. Answer says "a Daviesia species" -> partially_meets (right genus, missing
↪  species). Answer says "Bossiaea ensata" -> contradicts (wrong genus entirely).
- Rubric requires "269 cu in displacement", answer says "268.5 cu in" or "around 270" ->
↪  meets (trivial rounding, reader not misled). Answer says "between 250 and 300 cu in" ->
↪  partially_meets (correct range but too vague to be useful). Answer says "340 cu in" ->
↪  contradicts.
- Rubric requires "PhD focused on British and American naval history", answer says "PhD in
↪  History" -> partially_meets (right field, missing the specific focus area).

"missing" vs "partially_meets":
- If the answer says nothing at all about the topic -> missing.
- If the answer touches on the topic but with wrong specifics or insufficient detail ->
↪  partially_meets.

"contradicts" -- use ONLY for substantive factual errors:
- Wrong identification (e.g. Gala apple when it's Honeycrisp).
- Wrong dates, wrong people, fabricated claims.
- Do NOT use "contradicts" for trivial numerical rounding, minor unit conversions, or
↪  inconsequential precision differences.
- When in doubt, ask: "Would a knowledgeable reader consider this answer materially wrong
↪  on this point?" If no, it is not a contradiction.

Output your evaluation as a JSON object with the following structure:
```json
{{
  "evaluations": [
    {{
      "category": "<Answer Critical | Valuable | Context>",
      "handle": "<rubric handle>",
      "verdict": "<meets | partially_meets | missing | contradicts>",
      "justification": "<1-2 sentence explanation>"
    }}
  ]
}}
```

IMPORTANT:
- Evaluate EVERY rubric element listed above. Do not skip any.
- Output ONLY the JSON object, no other text before or after it.
- Evaluate the quality and factuality of the response holistically, not mechanistically.
```

B.6 TEXT-ONLY QUESTION CONVERSION

The prompt used to convert each multimodal question into a self-contained text-only question by resolving the entity from the rubrics, and to remove the identification rubric that the converted question makes trivial (Section 6.2).

Text-Only Conversion Prompt

```
You will receive: (1) an image, (2) a user question about the image, (3) the ground-truth
↪  entity name in the image, and (4) a set of grading rubrics for the question.
```

```
Your task is to convert the multimodal user question into an unambiguous, text-only question that can be perfectly understood and answered without ever seeing the image, AND to remove the identification rubric that becomes trivially satisfied once the subject is named in the converted question.

You will work in FOUR phases. Each phase is a separate top-level key in your JSON response. You MUST complete each phase fully before starting the next.

================================
PHASE 1: TARGET ALIGNMENT
================================
A. TARGET ALIGNMENT
   - State the specific subject targeted by the user's question (this may be the main focus of the image or a specific secondary object).
   - State what entity the provided Ground Truth name refers to.
   - Note: The Ground Truth Entity is *intended* to help identify the subject. However, mismatches can occur (e.g., GT is the author, but the question is about the book). You MUST prioritize the question's actual target as your strict anchor.

B. RESOLVE THE MOST-SPECIFIC TRUE ENTITY (use the rubrics)
   The provided Ground Truth Entity Name is frequently UNDER-SPECIFIED (e.g., a generic category like "changing table", "ATM", or "nutcracker statue") rather than the real, named entity. A question anchored to an under-specified entity is often meaningless or non-answerable as standalone text (e.g., "the manufacturer of the changing table" makes no sense without the image).
   - The rubrics contain the ground-truth-quality information humans used to grade the answer. Among the `Answer_Critical` rubric elements, there is almost always an **identification rubric** -- the element that pins down what the subject actually is. It is USUALLY the first `Answer_Critical` element, and its `Handle` typically reads like "Identify <X>", "<X> Identification", "Make And Model", "Scientific Name", "Identify Brand/Manufacturer/Species/Product", etc. Its `Ingredient` (and `Specifics`) name the real entity.
   - Read that identification rubric and extract the MOST-SPECIFIC real entity it establishes (e.g., a specific brand, manufacturer, model, species, product, or named work -- such as "Rubbermaid Commercial Products" rather than "changing table").
   - If multiple rubric elements add specificity (e.g., brand + model + generation), combine them into the single most-specific correct entity name.
   - If NO rubric element clearly performs identification, fall back to the most specific name available from the first `Answer_Critical` element, then the Ground Truth Entity Name.
   - Record this as the `resolved_entity`. This -- NOT the raw Ground Truth name -- is the strict anchor for Phase 2.

================================
PHASE 2: QUESTION CONVERSION
================================
Convert the original question into a standalone text question, anchored to the `resolved_entity` from Phase 1.
- **Drop image-only identification asks**: If the question asks to "identify", "name", "what is the make/model/brand/species of", or otherwise determine WHAT the subject is, REMOVE that sub-question. Identification is only meaningful when looking at the image; in a text-only question you already state the entity. Keep only the substantive, knowledge-seeking parts of the question.
   - Example: "Could you identify the manufacturer of the changing table and provide a detailed overview about their history in the commercial childcare market?" (resolved_entity = "Rubbermaid Commercial Products") -> "Could you provide a detailed overview of the history of Rubbermaid Commercial Products in the commercial childcare market?"
- **Substitution**: Replace deictic pronouns and image references (e.g., "this car", "the building in the picture", "it", "the changing table") with the exact `resolved_entity`.
- **Grammar & Flow**: Ensure the new question reads as a natural, meaningful, grammatical standalone question.
- **Preserve Intent**: Do not answer the question or add external facts beyond inserting the resolved entity; simply reformulate the question itself.

================================
PHASE 3: AMBIGUITY VERDICT
================================
Evaluate your converted text question. Can a person perfectly understand and answer this question using ONLY general knowledge, without needing to look at the original image?

Assess the converted question for remaining visual dependencies:
- **Spatial/Relational Dependencies**: Does it ask about something "next to" or "behind" the resolved entity without naming it? (e.g., "What is the red object next to the Honda CR-V?") -> AMBIGUOUS.
- **Specific Visual States**: Does it ask about a transient, image-specific state? (e.g., "Is the Honda CR-V in this picture damaged?" or "What color is the apple?") -> AMBIGUOUS.
- **OCR/Text in Image**: Does it ask to read unspecified text? (e.g., "What does the sign on the Honda CR-V say?") -> AMBIGUOUS.
```

```
- **General Knowledge**: Does it ask about factual history, specifications, processes, or general traits of the resolved entity? (e.g., "What year was the Honda CR-V first produced?") -> UNAMBIGUOUS.

Note: A question is NOT ambiguous merely because the original Ground Truth name was under-specified -- if you successfully resolved a specific entity in Phase 1 and removed image-only identification asks in Phase 2, the converted question should be UNAMBIGUOUS.

Determine the final verdict: `UNAMBIGUOUS` or `AMBIGUOUS`.

================================
PHASE 4: REMOVE THE IDENTIFICATION RUBRIC
================================
In the text-only question the subject is now named explicitly, so any `Answer_Critical` rubric element whose SOLE purpose is to identify/name that subject is trivially satisfied and must be removed before grading the text-only answer.
- Select the SINGLE `Answer_Critical` element (if any) whose only job is to establish WHAT the subject is -- the identification rubric you used in Phase 1B (Handle like "Identify <X>", "<X> Identification", "Make And Model", "Scientific Name", etc.). Its removal must follow directly from the entity now being named in the converted question.
- Do NOT remove an element that:
   - identifies a DIFFERENT or SECONDARY thing (a component, a material, one item among several, a background object) that the converted question does not name;
   - requires coverage, attributes, counts, comparison, ordering, or any substantive knowledge beyond naming the subject (e.g., "Identify at least 2 of the following nutrients", "Compare crush resistance").
- If no element qualifies, return -1.
- Report the 0-based INDEX of the removed element within the `Answer_Critical` list.

--------------------------------
OUTPUT FORMAT
--------------------------------
Return a single JSON object with EXACTLY these top-level keys (no markdown fences, no comments, no extra text):
{{
  "Phase_1_Target_Alignment": {{
    "targeted_subject": "<string>",
    "ground_truth": "<string>",
    "identification_rubric": "<string -- quote/summarize the Answer_Critical element used to identify the real entity, or state none was found>",
    "resolved_entity": "<string -- the most-specific true entity name; the strict anchor for conversion>",
    "alignment_note": "<string -- note any mismatch or under-specification and confirm the resolved target>"
  }},
  "Phase_2_Question_Conversion": {{
    "original_question": "<string>",
    "removed_identification_part": "<string -- the identification sub-question you dropped, or 'none'>",
    "converted_text_question": "<string>"
  }},
  "Phase_3_Ambiguity_Verdict": {{
    "reasoning": "<string -- explain whether the converted question still relies on unstated visual context, spatial relations, or transient visual states>",
    "verdict": "<enum: UNAMBIGUOUS | AMBIGUOUS>"
  }},
  "Phase_4_Remove_Identification_Rubric": {{
    "remove_index": "<int -- 0-based index into Answer_Critical, or -1 if none qualifies>",
    "removed_handle": "<string -- the Handle of the removed element, or empty>",
    "reason": "<string, <=15 words>"
  }}
}}

--------------------------------
INPUT
--------------------------------
- Original Question: {question}
- Ground Truth Entity Name: {entity_name}
- Grading Rubrics: {rubrics}
```

C FULL EXAMPLES

Figure 1 shows a partial example in the main text. Here we give two complete examples, reproduced verbatim from GAMUT, that together exercise every meta-rubric type and every conversion rule of Section 3. Each figure shows the question and its human-verified web evidence, the full structured meta-rubric (Section 3.1), and the complete binary checklist it compiles into (Section 3.2), with each check placed in its importance tier.

Table 2: Judge agreement: all 14 models scored by three different judges on all 1,813 questions. Gemini 3.1 Pro and Claude Opus 4.8 give an identical ranking; the weaker Qwen3-VL 235B is more lenient and preserves the same broad ordering. Scores are GAMUT score percentages.

Model	Gemini	Claude	Qwen3-VL 235B
Gemini 3.1 Pro	58.7	58.7	67.0
Gemini 3 Flash	57.9	57.4	66.1
Claude Opus 4.8	53.1	52.8	60.1
Gemini 2.5 Pro	51.8	51.7	60.9
Claude Opus 4.6	50.5	50.1	59.1
GPT-5.4	42.6	41.3	51.0
Claude Sonnet 4.6	41.5	40.4	48.3
GPT-4o	34.2	33.6	38.8
Qwen3-VL 235B	33.0	31.2	44.0
Llama 4 Maverick	18.3	16.7	22.1
Llama 4 Scout	14.1	12.8	18.6
Qwen3-VL 8B	13.7	12.6	23.0
Llama 3.2 90B	11.8	10.1	16.3
Llama 3.2 11B	5.1	3.3	9.6

Figure 3 is the complete version of the jollof rice example from Figure 1. Its Flexible List of supporting ingredients is Answer-Critical, so its baseline threshold check stays Answer-Critical while its per-item checks drop to Valuable, the one-level demotion of Section 3.2. Its Process has both mandatory and optional steps, so it produces one sequence check over the mandatory steps at Answer-Critical and a second over the full ordering at Valuable. The meta-insight attached to the Flexible List becomes a single Valuable check.

Figure 4 is a Tang *sancai* figure, and it exercises the one type the jollof example does not: a Relationship, connecting the lead-based glaze to why it runs during firing. It also places a meta-insight inside a Strict List rather than a Flexible List, and it uses all five meta-rubric types in a single question. Together the two examples cover Simple Knowledge, Strict List, Flexible List, Process, Relationship, and meta-insights, and show how each compiles into binary checks whose importance tier follows the rules of Section 3.2.

D Annotator Training and Auditing

Question creation Annotators were given guidelines describing the task, the criteria for a good question (naturalness, requires multi-step search, image-dependent, and so on), and worked examples. They then completed scored gold jobs, and those passing at 95% moved to the live task. Completed jobs were audited at 10% and accepted or rejected with feedback, and further rounds followed the two-way review with third-annotator adjudication described in Section 4.1.

Rubric creation Annotators watched a task-and-tool video, then took a guidelines quiz (10 two-part items, each a true/false paired with a related multiple-choice question); those passing at 90% advanced. After practice examples, they attended two 45-minute trainings, one on meta-rubrics and one on binary rubrics, each with a Q&A session that prompted further clarifications to the guidelines, and a chat channel connected annotators with the authors throughout. Completed jobs were audited at 10% and returned with comments and edge-case feedback for revision.

E Additional Results and Analysis

This appendix gives the full versions of the results condensed into Table 1, and the per-domain topic composition of the questions (Section 5.1). Table 2 reports every model's GAMUT score under each of the three judges (Section 6.1), and Table 3 the multimodal and text-only scores side by side (Section 6.2).

Table 4 breaks down the primary-judge verdicts by compiled-check kind (Section 6.1). The released rubrics contain compiled checks, which we classify by the forms produced by the conversion rules



Figure 3: A complete example from GAMUT (jollof rice), the full version of Figure 1. Left: the structured meta-rubric. Right: the compiled binary checklist, grouped by importance tier. Bottom: the human-verified web evidence, grouped by source page. All text is reproduced verbatim.

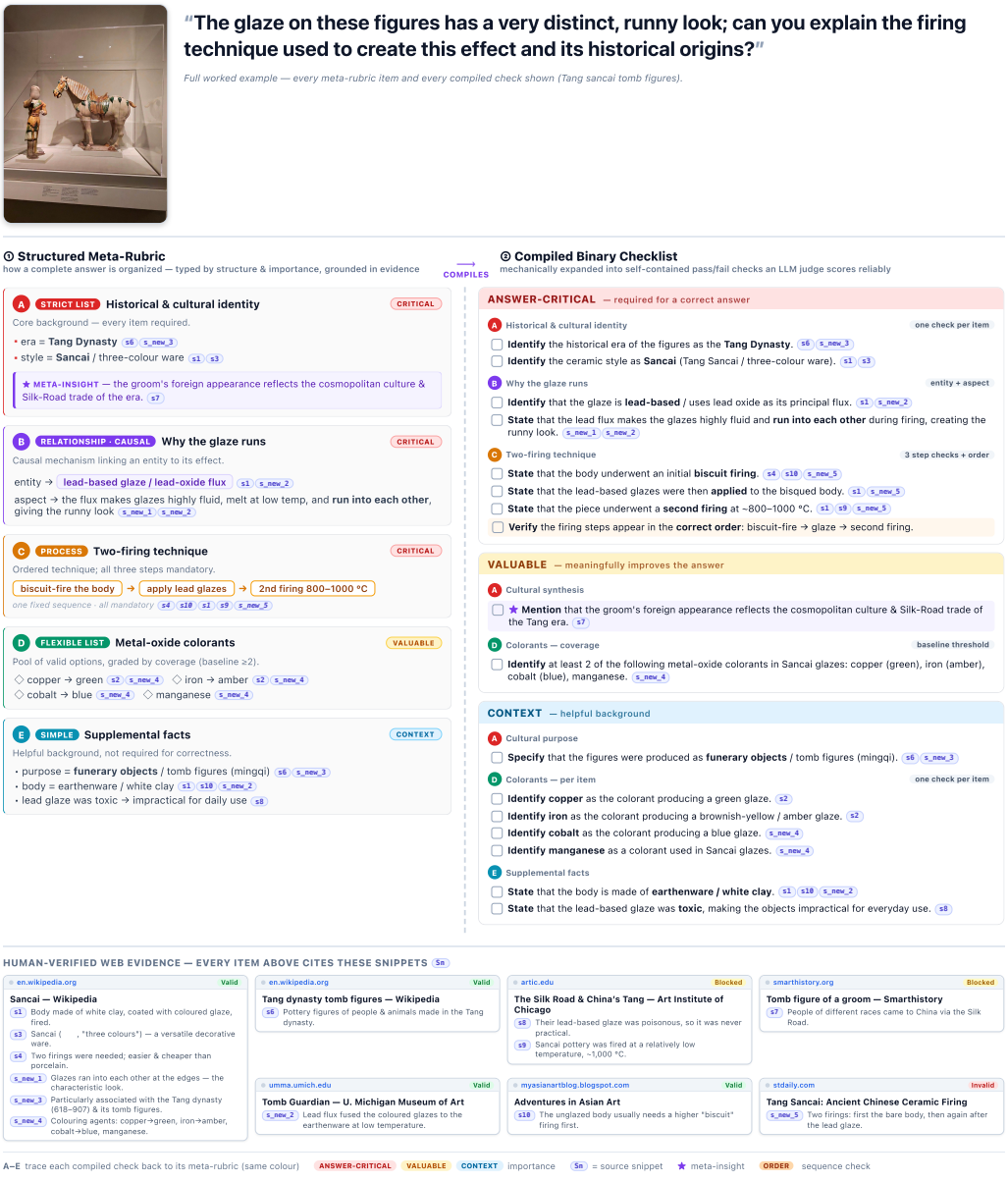


Figure 4: A complete example from GAMUT (a Tang *sancai* figure), exercising all five meta-rubric types including a Relationship. Layout as in Figure 3. All text is reproduced verbatim.

Table 3: Multimodal versus text-only GAMUT score on the same 1,806 questions. Scores are percentages; Δ is in percentage points.

Model	Multimodal	Text-only	Δ
Gemini 3.1 Pro	58.7	73.7	+15.0
Gemini 3 Flash	57.9	70.8	+12.9
Claude Opus 4.8	53.1	59.9	+6.8
Gemini 2.5 Pro	51.8	66.2	+14.4
Claude Opus 4.6	50.5	62.1	+11.6
GPT-5.4	42.6	58.5	+16.0
Claude Sonnet 4.6	41.5	53.8	+12.2
GPT-4o	34.2	56.5	+22.3
Qwen3-VL 235B	33.0	47.2	+14.2
Llama 4 Maverick	18.3	36.5	+18.2
Llama 3.2 90B	11.8	25.2	+13.4
Llama 4 Scout	14.1	27.8	+13.7
Qwen3-VL 8B	13.7	23.4	+9.7
Llama 3.2 11B	5.1	17.7	+12.7

Table 4: Verdicts by compiled-check kind under the primary judge (Gemini 3.1 Pro) on all 1,813 questions. *Identification* is the Answer-Critical check naming the pictured subject (1,525 checks); *threshold* checks grade flexible-list coverage (4,329); *sequence* checks grade process ordering (513); *other* includes all remaining checks (21,234). Scores are percentages computed with the tier formula in Section 3.3, pooling all checks of each kind; the last two columns report the percentage judged *contradicts*.

	Score by check kind				Contradicts (%)	
Model	Identification	Threshold	Sequence	Other	Identification	Other
Gemini 3.1 Pro	42.1	61.9	74.2	48.5	15.7	3.8
Gemini 3 Flash	38.4	63.6	76.7	47.4	17.4	4.2
Claude Opus 4.8	49.9	54.7	60.7	43.3	10.8	2.3
Gemini 2.5 Pro	27.1	59.0	68.8	41.1	21.6	5.7
Claude Opus 4.6	28.9	57.8	67.1	40.6	20.4	4.9
GPT-5.4	16.5	47.8	57.9	34.8	22.8	3.7
Claude Sonnet 4.6	25.0	45.7	57.9	31.4	20.9	4.4
GPT-4o	23.8	38.4	53.6	26.0	14.2	3.0
Qwen3-VL 235B	0.6	43.2	55.1	24.0	31.7	8.2
Llama 4 Maverick	-8.1	26.6	44.3	11.0	32.3	6.8
Llama 4 Scout	-13.3	24.0	42.0	7.8	34.4	7.5
Qwen3-VL 8B	-13.6	25.2	42.4	6.6	38.0	11.0
Llama 3.2 90B	-4.0	19.7	32.6	5.9	27.9	6.6
Llama 3.2 11B	-15.3	13.1	21.8	0.8	32.5	7.5

in Section 3.2. Coverage-threshold checks ("mention at least N of the following") and sequence checks ("evaluate the relative order of the matching steps") are identified by their templates. The identification check is the Answer-Critical check removed in the text-only variant (Section 6.2). All remaining checks, including per-item credit, strict-list items, step presence, relationships, and meta-insights, are grouped as *other*. All 14 models are scored on all 27,601 checks.

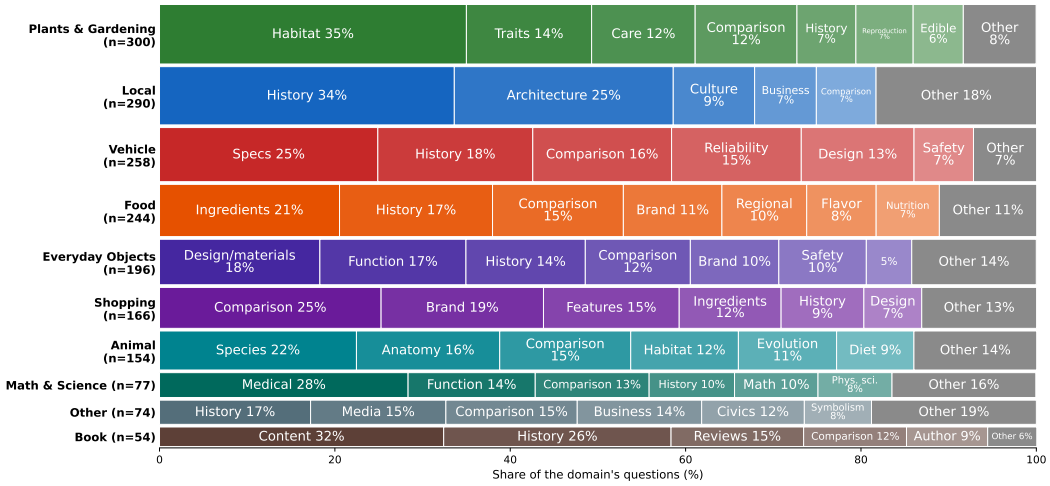


Figure 5: Topic composition of GAMUT questions within each of the 10 domains. Row height is proportional to the number of questions in the domain; horizontal segments give the share touching each topic, ranked by frequency, with shares below 5% folded into *Other*. The distinct per-domain profiles show questions are tailored to their subject rather than drawn from a few shared templates.