GLASSSEARCH-BENCH: TASK-ADAPTIVE EVALUATION AND CONTROLLED RUBRIC EVOLUTION FOR SMART-GLASSES SEARCH AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Smart-glasses search requests require different evaluation criteria: identifying an object, finding a suitable destination, and giving directions impose different obligations. Task-specific rubrics express these obligations, but regenerating criteria for each evaluation can change the checks applied to an unchanged task. We present GlassSearch-Bench, which constructs task-specific rubrics from versioned, reusable atoms to constrain this source of variation. An answer-blind planner extracts requirements; a constructor selects, parameterizes, and narrows applicable atoms, adding local criteria for uncovered requirements. Repeated product comparisons reuse the resulting frozen rubric. Recurring gaps supply candidate library revisions, with validation and human review governing new versions. In an 18-task development study, candidate expansion reduces local criteria from 32 to 18 relative to the eligibility-repaired library and matches all 15 controlled completion targets from five probes. A 90-case replay study documents changes in requirement records and atom selection across repeated construction. A 500-task archive with four system response sets demonstrates the diagnostic value of task-specific criteria: on 432 tasks assessable for all four systems under both reference-answer conditions, every system scores higher on object identification than on either place discovery or wayfinding. GlassSearch-Bench links task-specific evaluation, explicit control of criterion reconstruction, and cumulative library development to actionable product profiles.

1 INTRODUCTION

Identifying a building, finding a restaurant that meets a preference, and providing a usable route require different checks. A shared dimension such as helpfulness does not specify these obligations. Task-specific rubrics translate a request, its constraints, and available evidence into explicit criteria, consistent with instruction-dependent evaluation in FLASK (Ye et al., 2024).

Task specificity introduces a second problem: regenerating a rubric can change the checks applied to the same task. Requirement decompositions, selected criteria, and bindings may differ even when the input is unchanged. Recent work separates rubric generation from application (Siro et al., 2026) and compares independent with shared rubrics (Song et al., 2026). Our 90-case replay likewise records construction changes under an unchanged candidate pool. How can evaluation adapt to different tasks while retaining a consistent scoring basis for repeated comparisons?

GlassSearch-Bench separates three operations. An answer-blind planner identifies requirements. A constructor selects versioned rubric atoms, fills task-specific parameters, and explicitly narrows extra obligations; uncovered requirements receive local criteria. The final task rubric is frozen for product comparisons. Recurring gaps then nominate library revisions, evaluated separately before human release review. Reusable atoms anchor scoring content, frozen rubrics preserve the comparison, and reviewed revisions carry evaluation experience into later tasks.

The contributions are: (1) requirement-linked atomic construction that records reuse, adaptation, and new criteria; (2) development and replay studies that make criterion revision and reconstruction

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.

inspectable; and (3) a task taxonomy and four-system product analysis connecting scores to diagnostic needs. Candidate expansion reduces local criteria from 32 to 18 on 18 development tasks. Separately, a 500-task archive supplies 4,000 successful judgements: on 432 common tasks, every system in both reference conditions scores higher on object identification than on either discovery or wayfinding. The benchmark thus supports both current capability diagnosis and cumulative criterion development.

2 Related work

Situated tasks and evidence. SUPERGLASSES studies smart-glasses image questions, EgoBench combines egocentric video with interactive tool use, MC-Search annotates multimodal search, and JIR-Arena evaluates timely information recommendation (Jiang et al., 2026; Liu et al., 2026; Ning et al., 2026; Yang et al., 2025). We examine criteria for recorded answers to contextual requests. FLASK and IFEval make instruction-specific requirements explicit (Ye et al., 2024; Zhou et al., 2023); FActScore checks answer-derived atomic facts (Min et al., 2023). Our atoms instead represent input-derived obligations established before seeing the response.

Rubric construction and reuse. Prometheus uses supplied rubrics and references; LLM-Rubric calibrates rubric predictions against human judgements; RubricBench provides expert-annotated atomic criteria (Kim et al., 2024; Hashemi et al., 2024; Zhang et al., 2026). Support Vector Rubrics learns a rubric bank and selector, while RubricRAG retrieves related rubrics to condition generation (Sun et al., 2026; Dhole & Agichtein, 2026). Our construction exposes requirement bindings, template parameters, applicability, evidence boundaries, and narrowing operations, distinguishing unchanged reuse from adaptation and local generation.

Reliability and evolution. G-Eval, MT-Bench/Chatbot Arena, and LLMBar examine model judging, agreement, and instruction following (Liu et al., 2023; Zheng et al., 2023; Zeng et al., 2024). GER-Eval and shared-rubric research study criterion construction and evaluator agreement; Autorubric makes judging and reliability diagnostics configurable (Siro et al., 2026; Song et al., 2026; Rao & Callison-Burch, 2026). We separate construction replay from judge execution. Dynabench, LiveBench, and Benchmark Self-Evolving develop or refresh evaluation instances (Kiela et al., 2021; White et al., 2025; Wang et al., 2025); Rubrics on Trial evolves query-specific criteria using synthetic pairs (Yang et al., 2026). Our update unit is a reusable criterion and its applicability boundary, with frozen task rubrics retaining earlier comparisons.

3 Benchmark design

Units and collections. An input is $x = (o, q, c, t, e)$: observation, query, context, tool contract, and supplied evidence. Candidate response y and observed execution traces are separate. Responses are judgement units; source tasks are the minimum statistical grouping unit, keeping their criteria and perturbations together. Evidence snapshots retain verification material and temporal scope: visible relations may be supported by images, whereas walking reachability needs route evidence. Unsupported claims remain unresolved.

The 500-task archive contains 2,000 recorded responses from Li Auto Livis (hereafter Livis), Quark, Rokid, and RayNeo, with their collected interfaces and tool traces. Separate studies use 18 development tasks with 33 responses, a 90-case pool with 30 image and 60 text requests, and a 15-question subset with ten perturbed responses. These collections serve different roles and are not pooled; Appendix D records their overlap and evidence assets. [E1–E7]

Task taxonomy. Following the diagnostic perspective of CheckList and HELM (Ribeiro et al., 2020; Liang et al., 2022), each archived task receives one primary category: object identity, place identification, situated knowledge, place discovery, wayfinding, or long-tail decisions. AI-assisted curation used query texts and archived metadata, excluding answers, scores, and image-send status. The mapping was frozen before category-score computation, although overall archive results were already known. Appendix D gives category definitions and counts; Appendix C audits 23 uncertain classifications and their sensitivity. Overlapping demand tags retain location context, spatial reference, and explicit constraints. [E9]

The categories describe the user's requested outcome, not a universal rubric for every member of a category. Two place-discovery questions may differ in radius, price, accessibility, or whether a recommendation needs current opening information. Conversely, a single question may combine identification with a route request. We assign one primary category for disjoint reporting and retain overlapping demand tags for criterion construction. The complete archive contains 68 object-identity, 190 place-identification, 41 situated-knowledge, 171 discovery, 28 wayfinding, and two long-tail tasks. These counts expose the evaluation mixture: the archive supports a relatively well-populated identification analysis, whereas wayfinding estimates rest on fewer tasks and the long-tail category is descriptive only. Appendix D reports the comparison-eligible counts separately. [E9, E10]

4 Task-adaptive evaluation and controlled rubric evolution

Figure 1 separates the answer-blind construction path from answer-aware judging and from the later library-development path. The same task input can therefore yield a task-specific standard without giving the constructor access to the response being scored. The feedback path proposes future library changes; it does not edit the rubric already used for a product comparison.

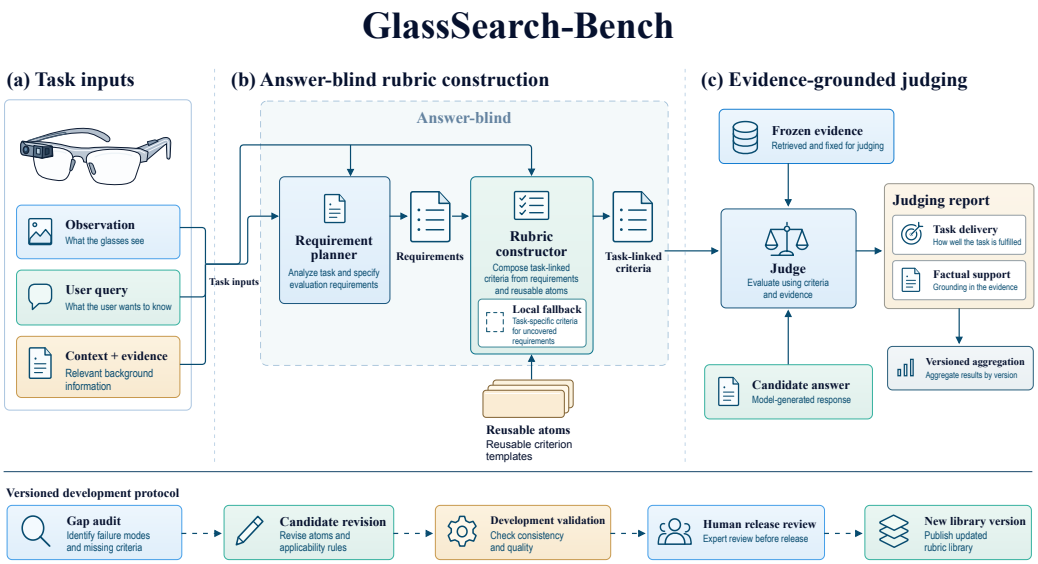

Figure 1: GlassSearch-Bench workflow. Observations and requests determine requirements and a frozen task rubric before candidate answers enter judging. The lower path describes a reviewed release protocol for future atom versions, not an autonomous improvement result established by the present experiments.

4.1 Answer-blind planning and atomic construction

A planner builds requirement set $R = P(x)$ before receiving candidate answers. Each requirement records an identifier, goal-or-constraint type, description, input origin and anchors, and evidence status. Constructor C binds criteria to these identifiers using library $\mathcal{A}_v$:

$$R = P(x), \qquad U = C(x, R, \mathcal{A}_v), \qquad y \notin \text{inputs}(P, C).$$

An atom stores an identity, version, applicability condition, parameter schema, criterion template, and evidence boundary. The constructor selects applicable atoms and fills parameters. For unchanged reuse, the effective criterion must match the parameter-substituted template exactly. When a template exceeds the current request, construction may only remove or narrow those obligations,

recording the effective criterion and reason. Uncovered requirements receive local criteria. For example, a coordinate-verifiable radius constraint need not require the answer to repeat its threshold. This removes an extra presentation obligation while retaining the requested spatial constraint. [E3]

Consider an illustrative request for a destination within a specified walking radius and with a stated preference. The planner records finding a suitable destination as a goal, the radius and preference as constraints, and the available image, location, and tool material as evidence sources. Construction may instantiate a distance-related atom with the requested threshold, select a preference-matching criterion, and add a local criterion if the library lacks a check for a request-specific detail. It must not turn an absent tool result into a verified walking route. The resulting rubric distinguishes naming a destination, satisfying its constraints, and supporting the claims used to justify it. This example illustrates the representation and is not an additional scored case.

The 18-task protocol allows at most six specialized criteria and validates duplicate atoms, unresolved parameters, and requirement bindings. It stores requirements, selected atoms, parameters, bindings, and effective criteria together, distinguishing unchanged reuse, narrowed reuse, and local generation. Repeated product comparisons use this frozen task rubric, rather than rebuilding it for every answer. The 90-case pipeline also stores temporary atom proposals separately from active library entries. A local criterion serves the current task; a proposal nominates reusable content for later validation. [E3, E4]

This distinction creates two version boundaries. An atom-library version determines what can be selected for a new task; a task-rubric version records what was actually selected, bound, narrowed, or locally written for that task. Updating the library does not silently rescore an earlier answer. A new comparison can either reuse the earlier frozen rubric to isolate product changes or declare a new rubric version and report the scoring-policy change. The stored requirement links and effective criterion text make those choices inspectable rather than implicit.

4.2 Judgement and versioned aggregation

The judge receives the frozen rubric, answer, and evidence. The revised development protocol separates task completion, factual support and hard constraints, missing-information handling, and style. Thus, appropriate clarification need not imply successful navigation. Deterministic v2.1 aggregation blocks full completion when an explicit hard constraint fails; satisfied constraints do not themselves count as completed goals. Partial, contradicted, and unverifiable delivery remain distinct. The archive uses its saved v13 scoring policy; the later response pilot uses pass/partial/fail/not-judgeable outcomes. Results from these contracts are reported separately. Appendix A records protocol changes and consistency checks. [E1, E3, E7]

4.3 Controlled library evolution

Repeated evaluation can expose reusable-representation gaps, applicability errors, missing evidence, or judge-execution errors. A missing check motivates a new atom; an overbroad check motivates a narrower rule. Candidate records retain supporting requirements, counterexamples, applicability boundaries, and evidence needs, with an initial recurrence threshold of three independent cases. Evidence and execution problems follow their own validation steps.

The release protocol separates a frozen core, an evolution pool, development validation, and an unused-source test, grouping shared sources and perturbation families. A selected candidate is frozen before transfer and core non-regression evaluation. Human review governs release; earlier results retain their rubric version. Implemented studies cover construction, proposals, candidate-library comparison, and replay. The further release evaluation is specified in Appendix E; it is not an observed autonomous improvement cycle.

5 Experiments

We examine product needs revealed by task-specific scoring, reusable coverage supplied by atoms, and changes under repeated construction. The 500-task product analysis uses saved v13 rubrics; the 18-task and 90-case studies examine separate development protocols.

Table 1: Category-level mean rubric scores in the merged result set. Each row uses identical tasks across systems, and both panels use identical tasks within each category. The comparison reflects recorded products with their collected interfaces and tool routing. Categories contain different task sets; cross-category differences are descriptive. [E9, E10]

Task category	n	Livis	Quark	Rokid	RayNeo
Judge without reference answers					
Object identity / attributes	67	0.569	0.572	0.685	0.523
Place / landmark identification	183	0.665	0.497	0.485	0.302
Situated knowledge / explanation	37	0.537	0.530	0.526	0.290
Place discovery / recommendation	123	0.275	0.084	0.248	0.225
Distance / wayfinding	20	0.255	0.029	0.185	0.192
Situated decisions (long tail)	2	0.021	0.098	0.000	0.000
All common tasks	432	0.506	0.370	0.436	0.307
Judge with reference answers					
Object identity / attributes	67	0.431	0.413	0.526	0.388
Place / landmark identification	183	0.357	0.355	0.325	0.252
Situated knowledge / explanation	37	0.356	0.315	0.398	0.206
Place discovery / recommendation	123	0.185	0.047	0.155	0.157
Distance / wayfinding	20	0.149	0.015	0.158	0.095
Situated decisions (long tail)	2	0.015	0.002	0.000	0.000
All common tasks	432	0.308	0.256	0.305	0.234

5.1 Product capabilities by task category

Table 1 uses the same 432 tasks assessable for all four systems under both reference conditions. Scores are [0,1] rubric aggregates; unassessable values remain null, not zero. Recorded product answers are fixed. Section 5.2 defines the merged result set and evidence conditions.

Identification exceeds location-based assistance. Every system in both conditions scores higher on object identity than on either discovery or wayfinding. With references, their ranges are 0.388–0.526, 0.047–0.185, and 0.015–0.158, respectively. Discovery and wayfinding need not have a fixed ordering: Rokid scores 0.155 and 0.158. The shared gap identifies location handling, destination constraints, and usable travel information as priorities for diagnosis.

Profiles reveal more than an aggregate rank. Rokid has the highest observed object-identity mean in both conditions. Livis leads place identification, but its with-reference 0.357 is close to Quark's 0.355. Overall with-reference means are 0.3081, 0.2556, 0.3047, and 0.2336 for Livis, Quark, Rokid, and RayNeo. The Livis–Rokid difference is 0.0034, with a task-bootstrap 95% interval of $[-0.0312, 0.0373]$. Category profiles expose distinct strengths without treating this small aggregate gap as a clear lead. The common subset excludes 48 discovery and eight wayfinding tasks; removing the 23 flagged classifications changes the highest-mean system for place identification and wayfinding (Appendix C).

Routing needs task context. RayNeo records 212 tasks without triggered image transmission, including 165 discovery requests. Within place identification, its with-reference means are 0.270 for 171 image-sent tasks and 0.068 for 19 untriggered tasks. The ASR-matched comparison is 0.269 versus 0.048. These are associations between different task groups, not causal effects; Appendix C reports both conditions and denominators. The taxonomy helps identify which traces warrant inspection.

5.2 Reference sensitivity on 500 archived tasks

The same 2,000 recorded answers are judged with and without references. The refreshed references were prepared from original images, questions, and recoverable public-source evidence, then frozen.

Table 2: Reference-condition comparison in the 4,000-record merged result set, on a [0,1] score scale. Columns n_- and n_+ count assessable tasks out of 500 without and with reference answers; n_{pair} is the system-specific paired sample used for both means. The cross-system common subset contains 432 tasks; excluding task 391 for provenance sensitivity leaves 431. [E10]

Response set	n_-	n_+	n_{pair}	$\bar{z}_-$	$\bar{z}_+$	Δ
Livis	469	483	456	0.4910	0.2994	-0.1916
Quark	483	485	476	0.3463	0.2360	-0.1102
Rokid	477	486	470	0.4270	0.2957	-0.1312
RayNeo	468	486	464	0.2977	0.2272	-0.0704

Model-review stages exclude product answers; targeted source amendments also use prior diagnostic context. This AI-assisted review supplies evaluation evidence, with independent human adjudication remaining separate. [E10]

The result set contains 4,000 successful execution records: 3,993 from the current run and seven reused historical records for task 391. Current records take precedence; all seven reused records enter the main statistics. The current run retains original image bytes, queries, responses, v13 rubrics, judge prompt, model profile, and aggregation, varying reference fields. It uses GPT-5.6 Luna with server-default sampling. Historical image/reference differences and three rubric-ID recoveries are documented in Appendix A. There are 3,837 numeric scores and 163 unassessable verdicts; each system has 500 successful records per condition.

For system s, I_s contains tasks with numeric scores in both arms:

$$\Delta_s = |I_s|^{-1} \sum_{i \in I_s} \left(z_{is}^{+\text{ref}} - z_{is}^{\text{no-ref}} \right).$$

Across 2,000 system–task pairs, 74 become assessable and 31 become unassessable, a net gain of 43. With-reference assessability is 96.6%, 97.0%, 97.2%, and 97.2%. Table 2 compares scores on identical paired tasks within each row, separating coverage changes from changed judgements. Removing task 391 changes each with-reference common-subset mean by less than 0.0006 and preserves its ordering. Task-bootstrap intervals use 10,000 paired resamples (seed 20260922), conditional on observed scores; they do not measure repeated-judge variability. Current-versus-historical no-reference differences and reference-recovery counts are retained in Appendix A. [E10]

Reference answers are therefore an evaluation condition, not an unconditional source of extra coverage. A reference can supply missing context for one judgement while exposing a conflict or an unverifiable historical claim in another. The 31 transitions toward unassessability prevent interpreting the net gain as a per-case guarantee. We report the transition counts alongside paired-score changes because conditioning only on numeric scores answers a different question from whether a score can be assigned at all. Neither comparison establishes that the refreshed references are independent human Gold; the source and review boundary remains part of the result.

5.3 Task-adaptive reuse and candidate-library revision

The 18-task study includes 13 historical questions and five probes built from historical images or coordinates. Eighteen generated answers plus three controlled answers per probe yield 33 responses. The 15 controlled targets comprise five task-correct and ten erroneous/omitted answers, clustered in five probes. A uses the original 21-atom library; B repairs two eligibility rules; C adds three candidate atoms to B; D generates freeform criteria. A–C allow local fallback. All conditions use GPT-5.6 Luna, the same answers, and a common construction protocol within each version. Equal criterion limits do not imply equal token budgets. [E3]

Under v2, C reduces local criteria from 32 to 18 relative to B, with 14 unchanged and 11 narrowed reuses. B and C share eligibility rules, isolating candidate expansion within the revised protocol. C matches 15/15 controlled completion targets; A, B, and D match 14, 14, and 13. This demonstrates increased reusable coverage on the development tasks, not held-out transfer. The v1 columns retain

Table 3: Library comparison on the 18-task development set. Columns v1 and v2 count completion-target matches among 15 controlled responses from five probes. FA and FR count v2 false accepts among 10 negative responses and false rejects among five positive responses. Reuse, Narrow, and Local count v2 unchanged reuse, narrowed reuse, and local criteria across all 18 tasks. Cross-condition comparisons use a common protocol within each version; the versions themselves include judgement-contract revisions. [E3]

Condition	v1	v2	FA	FR	Reuse	Narrow	Local
A: original library	15	14	0	1	8	0	37
B: eligibility repairs	14	14	1	0	4	8	32
C: candidate expansion	14	15	0	0	14	11	18
D: freeform	15	13	1	1	0	0	49

Table 4: Replay completion and reconstruction agreement. Exact fingerprints measure representation equality. Strict frozen-selector agreement additionally includes decisions, atom-to-requirement associations, and proposal outputs. Semantic equivalence requires a separate assessment. [E5]

Measurement	Full-pipeline replay	Requirements-frozen selector replay
Completed cases	90/90 in both runs	90/90
Candidate-pool fingerprint equality	90/90	Fixed by construction
Requirement fingerprint equality	1/90	Fixed by construction
Selector decision agreement	61/90	76/90
Selected atom-set agreement	67/90	76/90
Strict selector agreement	Not directly comparable across regenerated plans	25/90

the earlier protocol's results. All 162 logical jobs have usable final artifacts; Appendix A reports repairs and resource use.

5.4 Repeated construction on 90 cases

The selector uses `qwen3.5-omni-plus`, excluding historical answers and Gold from frozen inputs. Its archive contains 212 selected atom instances and 78 proposals; proposals are not active library entries. Two full replays share the manifest and candidate library. A further replay fixes the first requirement plan, candidate pool, and selector contract. [E4, E5]

All runs complete 90 cases. Full reconstruction preserves candidate pools but matches requirements in 1/90 cases, decisions in 61/90, and atom sets in 67/90. Fixing requirements gives 76/90 decision and atom-set agreement, with 25/90 strict agreement including bindings and proposals. Thus, a fixed library is not a fixed task rubric. Atoms anchor templates; freezing constructed requirements and rubrics preserves the actual checks. Exact fingerprints measure representation equality, not semantic equivalence or downstream score stability.

The two replay regimes expose different boundaries of a repeated evaluation. Rebuilding the requirement plan tests the complete path from a task to its checks; holding that plan fixed focuses on selection and binding. The increase from 61/90 to 76/90 decision agreement under the fixed-plan protocol is consistent with upstream planning contributing to reconstruction changes, but the regimes are not an estimate of a causal fraction. The lower 25/90 strict agreement further shows why a decision label alone does not fully specify a rubric: requirement associations and proposed-but-inactive atoms can differ while the coarse decision matches. For an unchanged product comparison, the unit to preserve is the effective task rubric, not merely the candidate atom pool.

Separate response-label audit. A 15-question pilot and ten perturbations complete under three judge conditions, but all 25 stored labels are pass despite intended negative perturbations. This

unresolved mismatch excludes the pilot from judge-accuracy comparisons; its protocol and recorded outcomes remain in Appendix B.

6 INDUSTRY FINDINGS AND DESIGN IMPLICATIONS

The shared gap between object identification and the two location-based task categories makes destination discovery and route delivery priorities for these recorded products. Evaluation should check entity identity, destination suitability, requested constraints, and travel information separately. An identification-focused demonstration does not summarize this capability profile.

These distinctions change what a product team can do with the benchmark. Low discovery scores direct inspection toward whether the request was resolved into suitable candidate places, whether preferences and location were applied, and whether the answer supplied enough evidence for a recommendation. Low wayfinding scores instead call for checking route availability, travel-mode assumptions, and whether the delivered directions satisfy the requested destination. A score alone cannot identify the faulty component; the task-linked criteria and recorded traces provide the next unit of investigation. The category means prioritize that investigation, while case-level review determines whether an apparent failure arises from the product, the evidence snapshot, or the judge.

The system-specific profiles and RayNeo routing analysis then narrow the diagnostic question: which requirement was missed, and which observation, recognition, retrieval, or synthesis step needs inspection? Reporting category, common sample, reference policy, and execution provenance makes that question interpretable. Controlled component tests can investigate mechanisms after trace review.

This separation also disciplines how the aggregate should be read. The 432-task common subset uses the same tasks across four systems and two reference conditions; it is not the full 500-task category distribution. It excludes 48 discovery tasks and eight wayfinding tasks, exactly the kinds of requests for which coverage can be difficult. Consequently, the common-subset profile is suited to paired comparison, while the full archive remains relevant for locating unassessable or poorly covered requests. A dashboard should expose both denominators. The observed image-routing association likewise identifies a trace slice to inspect, not a gain that would follow automatically from forcing image transmission.

Recurring diagnostic needs can also become reusable checks for location context, destination constraints, or travel information. The separate B–C development comparison shows how candidate atoms supply more rubric content. This links product diagnosis to evaluator development while versioned, frozen task rubrics preserve previous comparisons.

In practice, an evaluator can route a failure to one of three follow-ups. If the task rubric contains the right obligation but the product response misses it, the case enters product debugging. If the criterion asks for an obligation absent from the request, its applicability or narrowing rule needs review. If no criterion covers a recurring requirement, the gap becomes a candidate library change. The recorded requirement links make these routes visible to a reviewer. They also prevent a product weakness from being "fixed" by silently weakening the scoring standard after results are known.

7 BENCHMARK USE AND EXTENSION

For a new task, supply its observation, request, context, tool contract, and evidence; construct an answer-blind rubric from a declared library; freeze it before judging. Changed user constraints alter atom selection and instantiation, while repeated comparisons retain the chosen rubric. Repeated coverage gaps nominate reviewed library changes. Appendix E specifies unused-source/core validation, independent annotation, semantic-agreement and score-variability tests, and resource accounting. These experiments distinguish useful accumulation from merely growing the library; unused-source transfer remains a subsequent evaluation.

The workflow supports two different forms of adaptation. At evaluation time, a changed user request or tool contract can select and parameterize different atoms without changing the library. Over development time, repeated uncovered requirements can motivate an atom revision, but the proposed content must be evaluated before becoming active. To measure whether such revision helps, a future

study should compare old and new libraries on unused source groups and a frozen core, recording both new coverage and unintended obligations. To measure whether it stabilizes evaluation, repeated construction should compare semantic criterion agreement and downstream scores, while repeated judging under one frozen rubric estimates judge variation separately. These are prospective tests, not outcomes inferred from the present 18-task and 90-case studies.

A reusable benchmark record should therefore identify the source task, its input and evidence snapshot, the requirement plan, the library version, the selected atom versions and parameters, any narrowed or local criterion text, the frozen rubric, the answer and trace, and the judgement contract. This record lets another evaluator distinguish a product change from a criterion change or a changed evidence condition. The present archive and development studies preserve these distinctions for analysis, while a public, sanitized release of production-derived material still requires authorization and privacy review. For users who cannot access the original traces, aggregate tables support the reported descriptive comparisons; they do not substitute for an independent rerun of every judgement.

8 Scope and limitations

The findings describe four recorded products and selected development tasks, not a market-wide or isolated base-model comparison. The taxonomy is retrospective and retains 23 review flags. The merged snapshot includes seven historical executions with image/reference differences; its exclusion check quantifies their aggregate contribution. AI-reviewed references retain unresolved historical facts when collection-time evidence is unavailable. Default sampling permits judge variation, which task-bootstrap intervals do not estimate.

The 90-case replay measures reconstruction within the atomic pipeline, not an atomic-versus-freeform effect on semantic or score variability. Library revisions used development feedback, and the 15 controlled targets share five probes; sustained improvement across autonomous update cycles has not been evaluated. Independent held-out annotation remains necessary for transfer. Human review comprises development feedback and batch approval; the response-label mismatch remains under audit. Model names denote saved service profiles, not immutable provider checkpoints. Release requires source authorization and sanitization of images, locations, and traces.

9 Conclusion

GlassSearch-Bench matches criteria to tasks while preserving a consistent basis for repeated comparisons. Answer-blind planning, reusable atoms, explicit narrowing, and frozen rubrics separate adaptation from reconstruction; reviewed library revisions retain useful evaluation experience. Development studies document reuse and construction changes. The 4,000-record product analysis reveals distinct profiles and a shared gap between identifying objects and delivering location-based assistance, connecting benchmark use to concrete product and criterion-development priorities.

AI USE STATEMENT

Generative AI supported requirement planning, rubric construction, candidate-atom proposals, model judging, reference-answer review and source-assisted amendments, and the construction of controlled or perturbed responses. AI assistance also supported query-based taxonomy curation and review flags, implementation and analysis scripts, methodological discussion, interpretation of archived results, literature discovery and summarization, English and Chinese drafting and translation, reference formatting, and LaTeX preparation. Sections 3–5 document the generated material and review procedures. Numerical tables were recomputed from the recorded outputs, and evidence-file hashes were checked.

ETHICS STATEMENT

The development material includes production-derived requests, images, locations, and service traces. Redistribution is subject to source authorization, privacy review, and a defined access policy. The manuscript package excludes private trace contents, runtime credentials, and identifying local paths. Its system comparisons use the recorded interface and collection conditions.

REPRODUCIBILITY STATEMENT

Sections 3–5 specify the datasets, protocol versions, evidence inputs, and statistical units. Appendix A records provenance and protocol amendments; Appendix B documents the label audit; Appendix C details taxonomy review and task-conditioned trace analysis. The local development bundle contains extraction code, source hashes, and frozen outputs. A sanitized public bundle and its access conditions remain to be prepared.

REFERENCES

Kaustubh D. Dhole and Eugene Agichtein. RubricRAG: Towards Interpretable and Reliable LLM Evaluation via Domain Knowledge Retrieval for Rubric Generation, 2026. URL https://arxiv.org/abs/2603.20882.

Helia Hashemi, Jason Eisner, Corby Rosset, Benjamin Van Durme, and Chris Kedzie. LLM-Rubric: A Multidimensional, Calibrated Approach to Automated Evaluation of Natural Language Texts. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13806–13834. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.745. URL https://aclanthology.org/2024.acl-long.745/.

Zhuohang Jiang, Xu Yuan, Haohao Qu, Shanru Lin, Kanglong Liu, Wenqi Fan, and Qing Li. SUPERGLASSES: Benchmarking vision language models as intelligent agents for AI smart glasses, 2026. URL https://arxiv.org/abs/2602.22683.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Talat, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. Dynabench: Rethinking Benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4110–4124. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.324. URL https://aclanthology.org/2021.naacl-main.324/.

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*, 2024. URL https://arxiv.org/abs/2310.08491.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. Holistic Evaluation of Language Models, 2022. URL https://arxiv.org/abs/2211.09110.

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*, pp. 2511–2522. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.153. URL https://aclanthology.org/2023.emnlp-main.153/.

Yunqi Liu, Tong Niu, Zitong Wang, Zhenlong Dai, Yuqi Qing, Weiqiang Wang, and Jian Liu. EgoBench: An interactive egocentric multimodal benchmark for tool-using agents, 2026. URL https://arxiv.org/abs/2605.27820.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12076–12100. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.741. URL https://aclanthology.org/2023.emnlp-main.741/.

Xuying Ning, Dongqi Fu, Tianxin Wei, Mengting Ai, Jiaru Zou, Ting-Wei Li, Hanghang Tong, Yada Zhu, Hendrik Hamann, and Jingrui He. MC-Search: Evaluating and enhancing multimodal agentic search with structured long reasoning chains, 2026. URL https://arxiv.org/abs/2603.00873.

Delip Rao and Chris Callison-Burch. Autorubric: A Unifying Framework for Rubric-Based LLM Evaluation on Non-Verifiable Tasks, 2026. URL https://arxiv.org/abs/2603.00077.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4902–4912. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.442. URL https://aclanthology.org/2020.acl-main.442/.

Clemencia Siro, Pourya Aliannejadi, and Mohammad Aliannejadi. Learning to Judge: LLMs Designing and Applying Evaluation Rubrics. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 6371–6389. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-eacl.335. URL https://aclanthology.org/2026.findings-eacl.335/.

Mingyang Song, Mao Zheng, and Chenning Xu. Beyond the Illusion of Consensus: From Surface Heuristics to Knowledge-Grounded Evaluation in LLM-as-a-Judge, 2026. URL https://arxiv.org/abs/2603.11027.

Mengyuan Sun, Yu Li, Zhuohao Yu, Shikun Zhang, and Wei Ye. Support vector rubrics: Closing the gap between self-generated and human rubrics, 2026. URL https://arxiv.org/abs/2606.08077.

Siyuan Wang, Zhuohan Long, Zhihao Fan, Xuanjing Huang, and Zhongyu Wei. Benchmark self-evolving: A multi-agent framework for dynamic LLM evaluation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 3310–3328. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.coling-main.223/.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Sandha, Siddartha Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. LiveBench: A Challenging, Contamination-Limited LLM Benchmark. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/e4a46394ba5378b3f9a186a5b4c650d1-Abstract-Conference.html.

Haocheng Yang, Licheng Pan, Xiaoxi Li, Zhichao Chen, Zhiheng Zhang, Yuan Lu, Haoxuan Li, and Hao Wang. Rubrics on Trial: Evolving Rubrics from a Single Query via Synthetic Pairwise Evidence, 2026. URL https://arxiv.org/abs/2607.15092.

Ke Yang, Kevin Ros, Shankar Kumar Senthil Kumar, and ChengXiang Zhai. JIR-Arena: The first benchmark dataset for just-in-time information recommendation, 2025. URL https://arxiv.org/abs/2505.13550.

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*, 2024. URL https://arxiv.org/abs/2307.10928.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. Evaluating Large Language Models at Evaluating Instruction Following. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.07641.

Qiyuan Zhang, Junyi Zhou, Yufei Wang, Fuyuan Lyu, Yidong Ming, Can Xu, Qingfeng Sun, Kai Zheng, Peng Kang, Xue Liu, and Chen Ma. RubricBench: Aligning model-generated rubrics with human standards, 2026. URL https://arxiv.org/abs/2603.01562.

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, 2023. URL https://arxiv.org/abs/2306.05685.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-Following Evaluation for Large Language Models, 2023. URL https://arxiv.org/abs/2311.07911.

A PROVENANCE AND PROTOCOL AMENDMENTS

Evidence identifiers in the text refer to the following development artifacts, not to independent scholarly publications:

E1. The 500-task historical evaluation archive.

E2. The 500-task reference-answer sensitivity experiment and frozen protocol.

E3. The 18-task development runs, criterion contracts, and v2.1 aggregation rules.

E4. The 90-case development manifest, frozen selection outputs, and coverage audit.

E5. Full-pipeline and requirements-frozen selector replay comparisons.

E6. The 15-question evidence manifest, 15 snapshots, and five image assets.

E7. Recorded labels, perturbation manifest, and six three-condition judge-output files.

E8. Archived qualitative inspection of development failures.

E9. The frozen 500-task taxonomy, category-score analysis, source-hash manifest, and classification, analysis, and verification scripts.

E10. The refreshed reference set, current-first merged 4,000-record result set, historical-reuse provenance, recomputed category and routing analyses, bootstrap results, and independent arithmetic checks dated 2026-09-22.

These identifiers connect reported results to their archived development artifacts.

The reference refresh retains the saved v13 rubric and scoring policy. All 500 references were reviewed using original images and questions; source-assisted review added recoverable external facts while preserving missing historical evidence. The final reference statuses comprise 59 identified, 251 qualified, and 190 insufficient-evidence cases. The current run varies four reference fields: answer, note, source, and status. The client omits temperature and uses server defaults. The earlier historical evaluator uses a separate evidence and scoring policy and produces numeric aggregates on 59 of the 500 tasks for each system. The later response-level pilot also has its own aggregation contract. [E1, E2, E7, E10]

The current run yields 3,993 usable execution records, including three recovered by a one-to-one correction of rubric-ID hexadecimal suffixes after their semantic prefixes and complete item coverage were checked. Verdicts, scores, explanations, and evidence are unchanged. Seven remaining slots for task 391 use the specified historical successful executions. Livis with references retains its current score; the other seven system–condition slots retain their historical scores, each with its execution identifier. Historical execution files mention a transformed image, while an archived user correction states that the original was used. This image lineage remains unresolved, and the three reused with-reference records also use earlier reference fields. The main tables include these seven records under the declared merged-snapshot policy. Excluding task 391 leaves 431 common tasks, with with-reference means of 0.3087, 0.2551, 0.3049, and 0.2342 for Livis, Quark, Rokid, and RayNeo. [E10]

The v1 library experiment reduces local criteria from 19 in A to seven in C. It also changes controlled-target matches from 15/15 to 14/15: additional disclosure and repetition requirements reject a task-correct coordinate-based answer. The v2 protocol introduces a shared requirement plan, recorded template narrowing, separate judgement axes, and revised completion aggregation, and reuses the development answers. Table 3 reports both versions.

In the revised run, 157 of 162 logical jobs are directly valid, and five require citation repair: four mechanical normalizations and one citation-only model request. Resource accounting totals 175 API attempts and 671,742 tokens, including failed attempts. [E3]

The 90-case selection archive contains 78 proposal instances, of which 70 lack a nonempty `requirement_ids` field, affecting 66 cases. These records remain candidate proposals. The response-level pilot skips criteria without valid links before applying local fallback. [E4, E7]

Qualitative inspection identifies a verdict accepting a right-side answer while its justification describes the left side, and a completed route-guidance verdict with an unresolved walking-reachability

claim. These development cases motivate inspection of verdict–justification consistency and dependencies between goals and supporting claims. [E8]

The original evidence extraction for E1–E8 records hashes for 19 source files, validates 20 evidence assets, and recomputes the original experimental tables. E9 records the frozen taxonomy and its original analysis. E10 verifies 914 frozen files, the 3,990 successful outputs preserved before ID recovery, and all 4,000 unique merged task–system–condition keys. Independent Python and JavaScript calculations reconcile coverage, paired changes, common-task means, and category summaries. Routing summaries join the same merged scores to the original trace-status metadata.

Reference and selection diagnostics. Of the 17 previously unassessable with-reference pairs identified in the earlier study, 15 now receive numeric scores.

We also retain two sensitivity checks. Removing task 391 changes each with-reference common-subset mean by less than 0.0006 and preserves their observed ordering. Independently, the current no-reference arm differs from the archived no-reference arm by -0.0284, -0.0417, -0.0314, and -0.0153 on their respective paired subsets. This run-to-run variation accompanies the reference comparison. The reported task-bootstrap intervals use 10,000 paired resamples with seed 20260922 and quantify task-level variation conditional on the selected scores; they do not estimate repeated-judge variability. [E10]

The 90-case pipeline uses `qwen3.5-omni-plus` for selection, with historical answers and Gold excluded from its frozen inputs. It produces 70 MIXED, 15 SELECT, and five NO_MATCH decisions, comprising 212 selected atom instances and 78 proposal instances. This exposes both reuse opportunities and candidate gaps across the development pool; proposals retain a separate status from active library atoms. A single user-level batch review approves the displayed requirement decompositions and selections. An automated historical-answer coverage audit assigns 56 good, 25 partial, eight poor, and one not-judgeable labels. These records describe development review and coverage, respectively. [E4]

B RECORDED LABELS FOR THE 25-RESPONSE PILOT

The response-level pilot judges the same 15 historical answers with a common GPT-5.6 Luna profile and evidence snapshots. Its three conditions are requirement-only criteria, requirement criteria with explicit evidence restrictions, and selected/proposed atom criteria with local fallback. The archived names are Legacy, Repaired, and Evolved. These conditions evaluate response-level construction and aggregation, separately from the A–D library study. [E7]

All three conditions complete the 15-answer run and its 10 perturbed-answer extension. The stored user labels mark all 25 responses pass, including variants constructed to alter central facts. This unresolved difference between construction intent and recorded labels places the pilot in the label-audit track; it is excluded from judge-accuracy comparisons. Appendix B reports the recorded agreement and outcome counts.

Table 5: Agreement with the recorded all-pass labels in the response-level pilot. The table describes the stored label audit.

Archived condition	Original 15 responses	Perturbed 10 responses	Combined agreement
Legacy: requirements only	5/15	2/10	7/25 (28%)
Repaired: evidence restrictions	4/15	1/10	5/25 (20%)
Evolved: atom criteria and fallback	6/15	2/10	8/25 (32%)

The 25 responses come from 15 source questions, and all stored labels are pass. Perturbation-construction hints specify five fail, four partial, and one evidence-limited response. Resolving the difference requires clarification of the recorded review and case-level evidence assessment; the current figures are excluded from judge-accuracy comparisons.

With no non-pass human labels, the false-acceptance denominator is zero and the rate is stored as null. Judge outcomes are 7 pass/17 fail/1 not-judgeable for Legacy, 5/20/0 for Repaired, and 8/15/2 for Evolved. Agreement with this all-pass file records pass frequency, while discrimination requires labelled positive and negative responses.

C TAXONOMY REVIEW AND TASK-CONDITIONED DIAGNOSTICS

Classification procedure and review flags. The query-based taxonomy was curated with AI assistance by reading all 500 query texts and their archived category and scenario metadata. The classification did not re-inspect every image. Each record retains the original labels, the new primary category, overlapping demand tags, and a review flag. The archive contains 500 distinct task identifiers and 500 distinct image hashes, with 488 unique query strings; repeated query wording is retained with its task context.

The 23 review flags cover 11 object-identity, two place-identification, two knowledge, four discovery, and four wayfinding tasks. Flags identify uncertainty about the request or category, not incorrect product answers. Human and image-level adjudication of these flagged classifications remains pending. The mapping was frozen before the category-score computation, with previously known overall results treated as part of its retrospective development context.

Classification sensitivity. Excluding flagged tasks changes the common-subset counts to 57, 181, 36, 119, 16, and two in taxonomy order. In the with-reference condition, the highest-mean system remains unchanged for object identity, knowledge, and discovery. For place identification, it changes from Livis to Quark; for wayfinding, it changes from Rokid to Livis as the sample drops from 20 to 16. These changes support reporting task profiles and classification sensitivity together, especially when leading means are close or the category is small. [E9, E10]

Image transmission within place identification. Across RayNeo's 190 place-identification tasks, 171 record image transmission and 19 record no triggered send. All 190 have scores in both reference conditions in the merged result set used by Table 6. This is a RayNeo-specific paired sample, distinct from the 432-task cross-system subset. The ASR-matched rows further restrict these records to the archived recognition-match status, which ignores punctuation and spaces.

Table 6: RayNeo place-identification mean rubric scores by recorded image-send status, with an additional ASR-match restriction. Both score columns use the same tasks within each row in the merged result set. The groups contain different tasks and were not randomized. [E9, E10]

Image / recognition status	n	Without reference	With reference
Image sent	171	0.310	0.270
Send not triggered	19	0.146	0.068
Image sent (ASR matched)	164	0.313	0.269
Send not triggered (ASR matched)	18	0.115	0.048

The lower observed means for untriggered image transmission occur in both reference conditions and persist within ASR-matched records. Category conditioning and the recognition-status restriction make the comparison more specific, while task difficulty, image content, and routing decisions can still differ between groups. The result supports targeted trace review and a subsequent controlled routing test, not a causal estimate from this archive.

D COLLECTION INVENTORY AND CATEGORY DEFINITIONS

Table 7: Collections and their evaluation roles. The 15-question pilot is selected from the 90-case pool, and its 10 perturbations remain attached to existing questions. Collection sizes are reported separately because the assets have different sampling and development roles. [E1–E7]

Collection	Evaluation unit	Size	Demonstrated use
Archived system outputs	Task with four response sets	500 tasks; 2,000 responses	Task-level system comparison and reference sensitivity
Targeted development study	Task with generated and controlled answers	18 tasks; 33 responses	Library and protocol comparisons
Production-derived development pool	Request and available context	90 cases	Requirement selection, coverage audit, repeated construction
Evidence-backed response pilot	Selected source question	15 questions	Three-condition judge execution and label review
Answer perturbations	Variant attached to a pilot question	10 responses	Contrast review and label audit

The 500-task archive contains four recorded response sets: Li Auto Livis (hereafter Livis), Quark, Rokid, and RayNeo. They were collected through system-specific interfaces and traces. The 90-case development pool contains 30 image requests and 60 text requests from two internal application sources, contributing 60 and 30 cases respectively. It supplies additional task contexts for requirement construction and replay. [E1, E4]

The 15-question pilot has 15 recovered trace snapshots: tool-return material for 10 text questions and original-image records for five image questions. Integrity checks match the manifest hashes for all 15 snapshots and five images. These records supply the evidence inputs for the response-level experiment. [E6]

D.1 CATEGORY DEFINITIONS AND COVERAGE

CheckList organizes behavioural tests by capabilities and test types, and HELM organizes evaluation across scenarios and metrics (Ribeiro et al., 2020; Liang et al., 2022). Following this diagnostic perspective, we classify tasks by the user's primary requested outcome, using five main categories and a long-tail decision category (Table 8). Each task receives one primary category; overlapping demand tags record properties such as spatial reference, location context, and explicit constraints. The categories separate identifying a place from retrieving suitable destinations and from answering distance or route requests.

AI-assisted curation used query texts and the archived category and scenario metadata, excluding candidate answers, reference answers, scores, judge verdicts, and image-transmission status from the classification inputs. Labels were frozen before computing the new category summaries. This is a retrospective analysis of an archive whose overall results were already known. Ambiguous wording, transcription, or missing context prompted review flags for 23 tasks; Appendix C records the classification audit and sensitivity analysis. [E9]

The demand tags identify 164 tasks with spatial-reference wording, 204 with location context, and 81 with specific preferences. These are overlapping request properties: a spatial reference locates the queried object but does not by itself establish a spatial-reasoning test. Object identification can also require external knowledge.

E EXTENSION AND RELEASE EVALUATION PROTOCOL

A benchmark run supports two connected development activities. Product teams can use category scores, requirement-level failures, and traces to decide which agent behaviour to investigate.

Table 8: Task taxonomy and comparison coverage. All counts tasks in the complete archive. Common counts tasks assessable for all four systems under both reference-answer conditions in the merged result set; category scores use this common subset. The two game-decision tasks are retained as long-tail coverage. [E9, E10]

Primary category	Requested outcome	All	Common
Object identity and attributes	Identify an object or its attributes	68	67
Place and landmark identification	Identify a building, place, or geographic feature	190	183
Situated knowledge and explanation	Explain an observed entity or retrieve facts about it	41	37
Place discovery and recommendation	Find destinations meeting a location or preference need	171	123
Distance estimation and wayfinding	Provide distance, travel time, direction, or route guidance	28	20
Situated decision-making	Choose an action in a game situation	2	2
Total		500	432

Evaluator developers can use repeated coverage and applicability gaps to propose reusable criteria. The same task record therefore supports both product diagnosis and the accumulation of evaluation knowledge, with changes routed to the component responsible for the gap.

For a new task, the intended workflow supplies the observation, request, context, tool contract, and evidence; constructs an answer-blind rubric from a declared library version; freezes that rubric; and then judges candidate responses. Changes in user constraints or evidence needs can change which atoms apply and how they are instantiated. Comparisons within a version reuse the frozen task requirements and rubrics. Evaluation studies can separately freeze requirements to examine selection, freeze the rubric to examine judging, or vary reference answers to examine evidence sensitivity.

The taxonomy also guides new task collection. Only six archived queries contain explicit quantitative constraints, and none explicitly asks for a time-sensitive state such as current opening status or stock availability under the defined tag policy. Future collections can target these requests and expand wayfinding coverage. These counts describe explicit query demands; they do not imply that the archive contains no information that changes over time. [E9]

For subsequent library releases, repeated development gaps nominate a new atom or an applicability revision. The protocol freezes the candidate before evaluation on unused source groups and the frozen core. Independent requirement and answer annotation, with adjudication of disagreements, supplies the validation targets. The comparison measures requirement coverage, unintended requirements, false acceptance, false rejection, and abstention. Useful accumulation means that an update covers recurring needs while preserving judgement quality; library size alone is insufficient. This extension tests transfer while retaining the development records used to select the revision.

Version comparisons use paired source-task analysis, with higher-level grouping where tasks share source material. Resource accounting includes input and output tokens, retries, repair calls, and human review effort. To evaluate reconstruction control, repeated freeform and atomic construction should use the same tasks, requirements, evidence, model, and criterion budget, measuring semantic criterion agreement and downstream score variability separately. Freezing the rubric while repeating judgement isolates judge variation from construction variation.

The release decision combines reusable coverage on new tasks with performance on the frozen core. Human review authorizes the version change, and prior results retain their original contracts. The present development studies establish the comparison artifacts and workflow; unused-source transfer remains a subsequent evaluation.