LiDR: Living Deep Research

Anonymous ACL submission

Abstract

Many important questions do not have stable answers. Current deep-research and QA systems mostly produce one-shot snapshots, making them poorly suited for domains where knowledge evolves over time. We define **Living Deep Research** as the task of maintaining a cited, human-readable research document over a stream of incoming evidence. To evaluate this capability, we introduce **LiDR-Bench**, a benchmark that simulates real-time understanding of well-scoped research topics by backcasting against recent expert survey papers. Each survey supplies a future gold standard; references are revealed only when historically visible; and systems produce document trajectories rather than single reports. Evaluation measures both whether systems recover important citations and whether they capture claims about established practice, recent shifts, failure modes, and frontiers. Reference-system experiments show that LiDR-Bench exposes meaningful differences between memory-only, one-shot retrieval, and stateful update strategies, while leaving substantial headroom: current systems recover only a fraction of the expert citations and claims. LiDR-Bench makes living research measurable: systems must decide what evidence matters, preserve useful context, and update their synthesis as knowledge changes.

1 Introduction

The answers to many important questions are not stable over time. In science, technology, financial markets, and public policy, the correct answer to the same question can change as new evidence arrives. Yet current AI systems—deep research, RAG, retrieval-augmented QA—usually process every query de novo. They retrieve, synthesize, answer, and forget.

This is not how human experts work. A scientist tracking a fast-moving field does not re-read the entire literature every time a new paper is published. Instead, they continuously update an existing mental model: filing away new results, adjusting beliefs at the margins, and occasionally restructuring when a paradigm shifts. The goal is not to rebuild the answer from scratch every time, but to maintain a living synthesis through small, incremental updates.

Consider a concrete case. Between 2023 and 2025, the answer to "how should language models be post-trained?" moved from supervised instruction tuning and RLHF, to direct preference optimization and other reward-model-free methods, to synthetic data, RLAIF, process supervision, and reinforcement learning on verifiable tasks. Ex ante, it was hard to know which thread would matter: many approaches produced strong local results or compelling demonstrations, while later results clarified which directions scaled, which were subsumed by simpler methods, and which only worked under narrow conditions. Tracking this space required filtering signal from noise across hundreds of papers per month to maintain a cohesive situational awareness of the research landscape.

We call this missing capability **Living Deep Research**: the task of maintaining a cited, human-readable document that correctly covers a fast-moving domain over time, given a stream of incoming evidence. Concretely, a system must (i) retrieve and ingest a stream of potentially relevant sources, (ii) decide what matters by exercising taste about which developments are significant and which are noise, (iii) make parsimonious updates

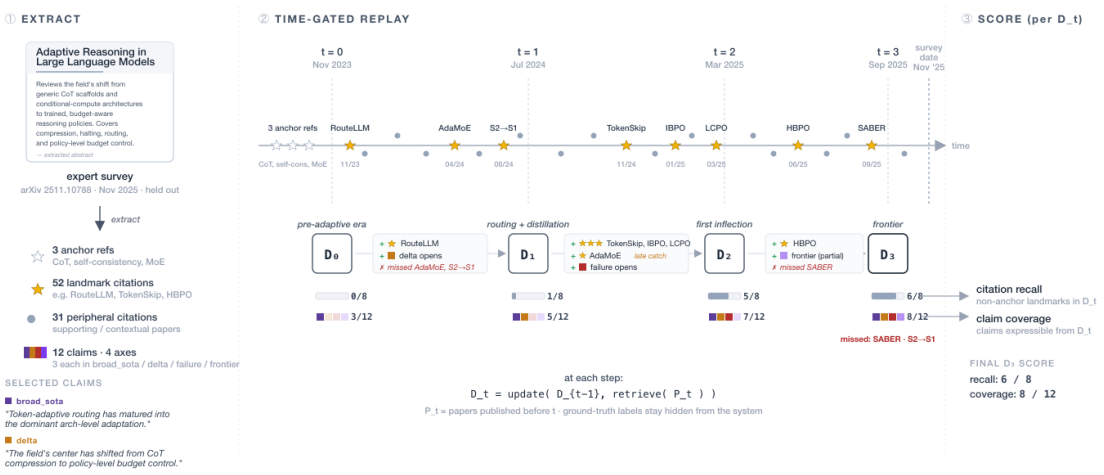

Figure 1: Worked LiDR-Bench instance. A held-out expert survey is converted into a time-gated living-research task: the survey supplies a future gold standard, including a research question, initialization anchors, dated gold citations, and claims about established practice, recent shifts, failures, and frontiers. During evaluation, the system never sees the survey. It receives the question and anchors, retrieves only from evidence that would have been publication-visible at each timestep, maintains a cited document trajectory, and is scored on whether its final document recovers the expert survey's citations and makes the survey's claims inferable.

to an existing synthesis rather than rewriting from scratch, and (iv) expose important changes so a human user can inspect, correct, and steer the document. The output is a trajectory of documents, rather than a single static report.

This makes living research a continual-learning-style evaluation problem: does updating an existing document produce a better synthesis than simply generating a fresh report from scratch every time new evidence arrives? Demand for living reports appears across domains: tracking a research field, monitoring competitive markets, synthesizing ongoing project updates across an organization, or curating a knowledge repository around a person or system. Living Deep Research sits at the intersection of amortized inference in agentic AI systems, knowledge management at organizational scale, and the broader challenge of maintaining coherent understanding in any domain where information moves fast across diverse corpora.

To measure these capabilities, we introduce **LiDR-Bench**, a benchmark for evaluating living research systems. The core difficulty is that living research is intrinsically temporal: to evaluate a system directly, one would have to run it forward for months or years and then wait to learn which papers, claims, failures, and open questions actually mattered. LiDR-Bench makes this problem testable by backcasting from expert survey papers in computer science—an appropriate domain because it is fast-moving, information-rich, and difficult to track. A survey published at time T supplies a retrospective gold standard including a research question, a bibliography, and claims about what changed, what works, what fails, and what remains open. LiDR-Bench replays the literature chronologically to test how well a system can match the expert survey's final citations and claims in "real time." In this way, the retrospective survey serves as an objective grading rubric for a continuous, forward-looking task.

We make three contributions:

1. **A formal task definition for living research.** We define living research as a time-indexed document-maintenance problem in which documents are explicit state, observations arrive over time, and systems produce document trajectories. In an evaluation setting, the trajectories can be scored against an objective gold standard.

2. **LiDR-Bench.** We construct a benchmark from expert computer science survey papers with time-gated chronologi-

cal evaluation, anchor papers, landmark and peripheral citation gold, frozen retrieval pools, and claim-coverage evaluation over four axes: established state-of-the-art practice, recent shifts, failure modes, and open frontiers.

3. **Empirical validation with reference systems.** On 100 surveys, LiDR-Bench discriminates among parametric, one-shot retrieval, gap-driven retrieval, and living-document update systems. The paired comparison between stateful document updates and stateless regeneration shows when accumulated document state helps, hurts, or merely preserves a stale scaffold.

We hope LiDR-Bench gives the community a concrete target for this problem. Our reference systems are validation probes, not proposed solutions: they show that the benchmark has signal, that stateful updating can help, and that the task remains far from saturated. Upon publication, we plan to release LiDR-Bench, the reference-system code, evaluation scripts, and a public leaderboard. We invite others to run stronger research agents, retrieval systems, memory mechanisms, update policies, and human-steerable workflows on it.

2 The Living Research Task

The four requirements of the Living Deep Research task compose into a simple loop. Given a query q, anchors A, and timesteps $T_0, \ldots, T_N$, a system produces a document trajectory $D_0, \ldots, D_N$:

Input	query q, anchors A, timesteps $T_0, \ldots, T_N$	
Output	document trajectory $D_0, \ldots, D_N$	
for $t = 0, \ldots, N$:		
P_t	$\leftarrow \{d : \mathrm{date}(d) \leq T_t\}$	*visible pool*
$\hat{O}_t$	$\leftarrow R(q, A, D_{t-1}, P_t)$	*retrieve/filter*
a_t	$\leftarrow \pi(D_{t-1}, \hat{O}_t)$	*update policy*
D_t	$\leftarrow U(D_{t-1}, a_t)$	*commit edit*
return	D_N	

The system's behavior is therefore defined by three primitives: R retrieves and filters a tractable observation set, π decides whether and how to update, and U applies the edit. Different choices of these primitives yield full rewrites, bounded patches, null updates, or other document-maintenance strategies. The output is a *trajectory* $D_0, \ldots, D_N$ rather than a single static report.

This loop enables inference amortized at the document level. The expensive work of organizing a field is done once and updated incrementally, serving as a materialized view over an evidence stream. Different choices of the three primitives yield different living-research systems; this is the design space the benchmark is built to expose.

Because the output of such a system is a trajectory of documents, evaluation must measure how well a document trajectory adapts over time as new information is observed. In LiDR-Bench, we score trajectories along two complementary axes: **citation recall** measures whether the document discovers the papers experts later judged important, and **claim coverage** measures whether it expresses the substantive state of the field.

3 LiDR-Bench

A peer-reviewed survey is an unusually clean ground truth for living research: experts have already done the hard work of deciding which papers were important, what has recently changed, what has publicly failed, and what remains open. LiDR-Bench turns that artifact into a time-gated evaluation task. We harvest recent surveys, extract their research questions, citations, and claim-bearing passages, label references as either landmark or peripheral, assign claims to four axes, and gate every reference by publication date. (See Figure 1) At timestep T_t, a system sees only what was visible before T_t and is asked to recover the citations and claims the survey author later landed on.

3.1 Task Instance and Time Gate

The current LiDR-Bench dataset consists of 100 recent ML/AI survey papers. Each instance contains: survey metadata and an extracted research question; arXiv-ID citation sets extracted from the survey bibliography, including 3–5 initialization anchors plus landmark and peripheral gold references; roughly 10K topical distractor references sourced from the arXiv API and selected to be semantically similar to the survey topic; and a claim bank organized around the four axes above. The gold and distractor references are frozen into the retrieval pool used at each timestep.

Candidate surveys are sourced by retrieving across Semantic Scholar, arXiv, and OpenAlex. We balance the cohort by topic clustering, extract references with Semantic Scholar / GROBID / arXiv fallbacks, freeze a per-survey retrieval pool containing the survey references plus roughly 10K topical distractors, and extract the claim bank from survey text. Run-specific configuration is reported with the experiments in §5.1, and pool coverage audits are reported in Appendix C. See Appendix A for a detailed description of the dataset construction.

The task is parameterized by a lookback window L, step size Δ, timesteps $T_0, \ldots, T_K$, a distractor cap N, and an evidence-selection budget k. For a survey published at T_{survey}, $T_0 = T_{\text{survey}} - L$ and T_K is the final pre-survey cutoff. At timestep T_t, the visible pool P_t is the subset of the frozen `pool` with publication date at or before T_t, plus the initialization `anchor_refs`. Retrieval is cumulative by default: a system may recover older visible evidence it missed earlier.

3.2 Two Scored Tasks

LiDR-Bench scores the maintained document along two complementary axes.

Citation recall (evidence discovery). A system gets credit when its document includes a citation marker matching a ground-truth arXiv ID that is date-visible at the current timestep. We split references into **landmark** and **peripheral** classes and report each separately. We also report `rec_new`, a citation-recall slice restricted to ground-truth references that were not available at initialization but became visible by the final timestep. This isolates whether a system acquires newly visible evidence rather than living off anchors, pretraining, or the initial document.

Claim coverage (synthesis quality). Claim coverage uses a fixed gold claim bank extracted from the held-out survey during benchmark construction. A claim is a short survey-derived statement about what the field has established, how it has changed, where methods fail, or what remains unresolved. Claims are organized along four axes: `broad_sota` for broad state-of-the-art methods and current best practices, `delta` for recent shifts or pivots, `failure` for known failures or broken assumptions, and `frontier` for open problems and future directions. At evaluation time, this claim bank is held fixed. Grading is an inferability test: for each candidate document and each gold claim, a stronger LLM judge sees only the claim and the living document and returns `present` (1.0), `partial` (0.5), or `absent` (0.0).

The two metrics are intentionally paired. Citation recall alone would reward bibliography stuffing; claim coverage alone would overestimate systems that paraphrase well from parametric memory. Together they test whether a system is finding the right evidence *and* integrating it into a useful synthesis.

3.3 Contamination Controls and Example

A frontier model may already know some of the cited papers or survey claims via parametric knowledge acquired during training. LiDR-Bench addresses this on two levels: (i) the cohort uses recent surveys, so much of the reference mass postdates frontier model training cutoffs; and (ii) we run a `parametric` baseline that generates reports solely from model parametric knowledge. In our primary run, this baseline has extremely poor performance. Appendix C.3 examines per-reference recall as a function of publication date: `parametric` recall on retrievable references is 0.1% before the LLM training cutoff and 0.0% after.

Figure 1 shows one benchmark instance built from a November 2025 survey on adaptive reasoning in large language models. The gold-standard survey is used to construct the benchmark artifacts: a research question, initialization anchors, gold citations, and a claim bank. It is never shown to the evaluated system. During evaluation, the system receives the question and anchors, retrieves from a date-gated pool containing the gold citations plus distractors, and produces a trajectory of cited research documents.

4 Reference Systems

We run a suite of reference systems to establish performance bounds and seed the design space. They are not claimed as optimized methods. Each system instantiates the loop from §2 differently: how it forms observations $\hat{O}_t$, whether the update policy π conditions on

the previous document D_{t-1}, and how `apply` commits the resulting action a_t.

`parametric` and `one-shot-rag` are controls: the first probes pretraining contamination by generating from memory alone, the second regenerates from time-gated retrieval at each step and approximates current deep-research and RAG usage. The remaining three systems isolate the role of document state. `lidr-rewrite` and `lidr-diff` use the same retrieval query as `one-shot-rag` but condition on the previous document, differing only in the update operator. `gap-driven` instead uses the document to steer retrieval, generating queries from gaps in the current document, then applies the same full-rewrite update operator as `lidr-rewrite`. For fairness, the three stateful systems share the same initial document D_0, built once per survey from the initial pool using the same retrieve-select-generate primitive used by `one-shot-rag`.

5 Experiments

5.1 Setup

We evaluate the 5 reference systems from §4 on the 100-survey 2026 LiDR-Bench cohort. The main run sets $L = 24$ months, $\Delta = 4$ months, $K = 6$ updates / 7 document timesteps per survey, runtime distractor cap $N = 1000$ using a stratified-time pool subset (`seed=0`), and cumulative retrieval at every timestep. All systems use the same `user_query`, `anchor_refs`, frozen `pool`, and timestep cutoffs for each survey. Retrieval uses top-50 search for `one-shot-rag`, `lidr-rewrite`, and `lidr-diff`; `gap-driven` issues 3 document-derived queries with top-30 search each, then deduplicates. The shared D_0 and all stateful update systems select $k = 24$ papers for synthesis/update after LLM reranking; `one-shot-rag` synthesizes directly from its top-50 retrieved papers.

The task instances are large enough that retrieval and synthesis both matter: each survey has 3–5 anchors, a claim bank over the four axes in §3.2, and 118.48 ± 7.77 scorable gold references on average at the final timestep. The frozen pools contain essentially all of those gold references, and 117.89 ± 7.70 are date-visible by the final pre-survey cutoff. Synthesis and update prompts use `gemini/gemini-2.5-flash`. Claim grading uses `gemini/gemini-3.1-pro-preview` with `max_tokens=65,536`. Reference-system prompts and guardrails are listed in Appendix B.

5.2 Main Result: Continuity Helps After Updates Begin

Table 2 gives the main result. Retrieval is necessary: `one-shot-rag` substantially improves over the memory-only `parametric` baseline. Document continuity adds another gain: `lidr-rewrite` and `lidr-diff` improve over `one-shot-rag` on final claim coverage, landmark recall, peripheral recall, and recall on newly available ground truth. We do not imply that these specific update methods are optimal. However, the results do confirm that LiDR-Bench successfully isolates the value of document continuity and demonstrates that accumulated state provides an advantage in these kinds of tasks.

The absolute scores also show that LiDR-Bench is far from saturated. Even the best reference system recovers only about half of the landmark/peripheral citation targets and covers well under half of the extracted survey claims. This is expected: survey bibliographies are expert selections rather than exhaustive truth sets, and the reference systems are simple probes rather than optimized agents. The headroom is the point of the benchmark.

Newly available evidence is the most direct citation-level test of living synthesis. The number of ground-truth references that become visible after T_0 varies widely across surveys: the average is 68.04, the median is 46, and the range is 9–281. We therefore report `rec_new` as a macro average: recall is computed separately within each survey, then averaged across surveys so that large-bibliography surveys do not dominate the metric. On this metric, `one-shot-rag` reaches 0.214, while `lidr-rewrite` and `lidr-diff` reach 0.307 and 0.313, a roughly 45% relative improvement. `gap-driven` is a useful counterexample: despite using document-derived queries, it recovers fewer newly available gold references and has lower final claim coverage, suggesting that gap queries can chase already-covered regions instead of finding the next important papers.

The reference systems are also diagnostic

Table 1: Reference systems.

System	Observation $\hat{O}_t$	State used by π	`apply` operator	What it tests
`parametric`	No retrieval; model memory only	None	Regenerate from memory	How much of the task comes from pretraining or temporal contamination?
`one-shot-rag`	Query the visible pool P_t at each timestep	None	Regenerate from retrieved evidence	How far does time-gated retrieval get without continuity?
`lidr-rewrite`	Query the visible pool P_t at each timestep	Previous document D_{t-1}	Replace with a full-document rewrite	Does preserving a prior synthesis help when new evidence arrives?
`lidr-diff`	Query the visible pool P_t at each timestep	Previous document D_{t-1}	Apply bounded markdown patches	Does a local edit operator maintain knowledge more stably than rewriting?
`gap-driven`	Query using gaps inferred from D_{t-1}	Previous document D_{t-1}	Replace with a full-document rewrite	Does using the document to steer retrieval find missing evidence, or chase redundant gaps?

Table 2: Main LiDR-Bench results on 100 surveys. Means across surveys with standard error. `cc_d0` is claim coverage at initialization, `cc_fin` is final claim coverage, Δ`cc` is final minus initial coverage, `L_rec` and `P_rec` are landmark and peripheral citation recall, `rec_new` is recall on ground-truth papers that became available after initialization, and `cited` is mean final citation count.

system	n	cc_d0	cc_fin	Δcc	L_rec	P_rec	rec_new	cited
`parametric`	100	0.222 ± 0.013	0.238 ± 0.013	0.016 ± 0.007	0.017 ± 0.003	0.010 ± 0.003	0.000 ± 0.000	6.22 ± 0.36
`one-shot-rag`	100	0.272 ± 0.013	0.344 ± 0.018	0.072 ± 0.015	0.351 ± 0.026	0.306 ± 0.020	0.214 ± 0.027	28.28 ± 1.04
`gap-driven`	100	0.260 ± 0.014	0.307 ± 0.016	0.048 ± 0.010	0.299 ± 0.018	0.369 ± 0.020	0.084 ± 0.016	29.69 ± 0.67
`lidr-rewrite`	100	0.260 ± 0.014	0.373 ± 0.017	0.114 ± 0.013	0.458 ± 0.028	0.472 ± 0.021	0.307 ± 0.034	41.49 ± 0.83
`lidr-diff`	100	0.260 ± 0.014	$\mathbf{0.381 \pm 0.018}$	$\mathbf{0.122 \pm 0.012}$	$\mathbf{0.475 \pm 0.030}$	$\mathbf{0.537 \pm 0.023}$	$\mathbf{0.313 \pm 0.036}$	46.19 ± 0.89

probes. `parametric` exposes parametric substitution: plausible claims from memory do not recover the time-gated evidence stream. `one-shot-rag` separates retrieval from continuity: it often finds useful papers, but regenerates a fresh synthesis each timestep. `gap-driven` exposes a failure mode specific to document-steered search: once the document appears to cover a region, gap queries can retrieve redundant evidence instead of the next frontier.

5.3 When the Advantage Appears

In our reference implementations, the impact of continuity is modest but consistent. Update systems have the largest mean claim-coverage gains from initialization to the final timestep, and the advantage over one-shot retrieval appears only after updates begin (Figure 3). Pairwise survey-level checks tell the same story: `lidr-diff` beats `one-shot-rag` on claim-coverage improvement in 62 of 98 comparable surveys, with a median advantage of 0.045; `lidr-rewrite` beats `one-shot-rag` in 58 of 97 comparable surveys, with a median advantage of 0.027.

A timestep-level paired analysis shows when the advantage appears. After the first evidence updates, `lidr-diff` and `lidr-rewrite` move ahead of `one-shot-rag`: by timestep 2 they lead by +0.031 and +0.035 claim-coverage points on average, and by the final timestep they remain ahead by +0.037 and +0.028. This pattern suggests that maintained document state is useful when evidence is revealed over time.

Appendix C.1 checks whether this gain is just a document-length effect. The update systems are somewhat longer than `one-shot-rag` on average (`lidr-diff`: 20.9K characters; `lidr-rewrite`: 18.3K; `one-shot-rag`: 15.5K), but the distributions overlap substantially and the gains in claim coverage are not simply explained by longer outputs.

5.4 How Continuity Helps: Documents Become Retrieval State

The most direct mechanism is that a synthesized research document serves as a form of "query enrichment". A static query is short and often underspecified; a document con-

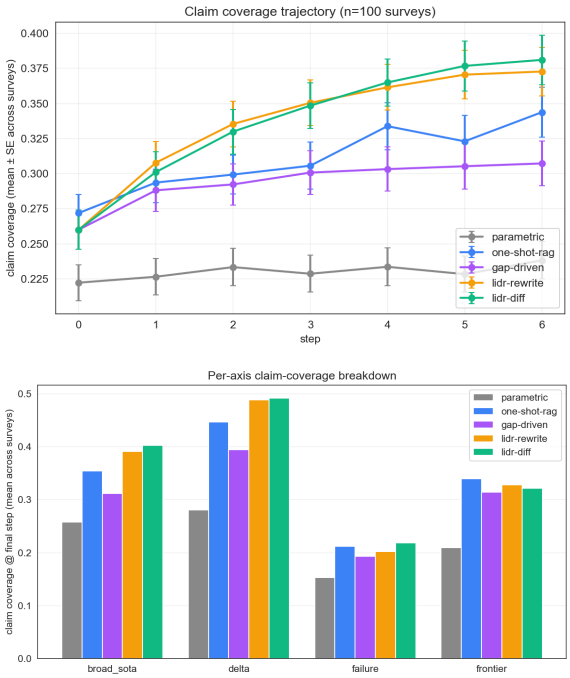

Figure 2: Claim coverage is easier to read visually than from the aggregate table alone. The time trajectory shows update systems steadily accumulating coverage while the parametric baseline remains nearly flat. The per-axis breakdown shows where the benchmark is hardest: failure-mode claims are substantially less covered than broad state-of-the-art or delta claims, even for the best update systems.

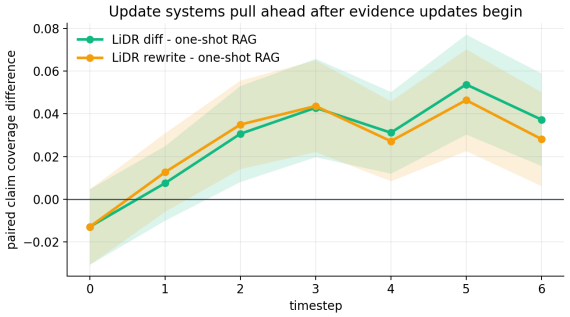

Figure 3: Paired claim-coverage difference over timesteps for the two update systems against `one-shot-rag`; the advantage emerges after evidence updates begin rather than at initialization.

tains the field's accumulated terminology, subproblems, method names. To test this mechanism without changing the generation system, we run a retrieval ablation over all non-initial timesteps: TF-IDF queries built from each system's prior document are compared against static queries built from the extracted `user_query` or survey title/abstract. Raw document queries improve recall@50 over the static user query: the prior `one-shot-rag` document reaches 0.731 recall@50 and the prior `lidr-diff` document reaches 0.725, versus 0.657 for `static_user_query`. In paired terms, the gains are +0.077 and +0.072 recall@50, with 88% and 86% win rates across survey-step pairs.

Two details matter for interpretation. First, this ablation supports document-grounded retrieval, not document continuity by itself: `one-shot-rag` and `lidr-diff` are essentially tied as retrieval text, while `parametric` still trails them. This suggests that synthesis itself is a strong retrieval aid, and that the gains in Table 2 likely combine better retrieval text, evidence retention, and update behavior. Second, asking an LLM to rewrite the `lidr-diff` document into three short search queries is worse than using the raw document text directly. The useful signal is already in the document; the extra query-rewriting layer adds noise.

5.5 Evidence Acquisition Under a Moving Denominator

A quirk of this benchmark is that the the denominator in the recall calculation changes over time: as more ground-truth papers become date-available, a raw recall scores may fall even when a document is citing more relevant papers in absolute terms. We therefore report both recall ratios and absolute found/available counts. The average survey has 68.04 ground-truth references that become available after initialization. `lidr-diff` cites 11.16 of these on average, compared with 7.85 for `one-shot-rag`, 3.20 for `gap-driven`, and effectively zero for `parametric`.

This is the most concrete account of how continuity helps in the current experiments. The stateful systems are exposed to more distinct relevant evidence over the trajectory, keep more of it in the final document, and use it to cover more survey claims. Appendix C makes the funnel explicit, and Appendix C.4 gives qualitative trajectory examples. The pool and time gate are almost complete; retrieval, evidence selection, and update retention are the bottlenecks. To keep those bottlenecks interpretable, the benchmark includes bounded document length,

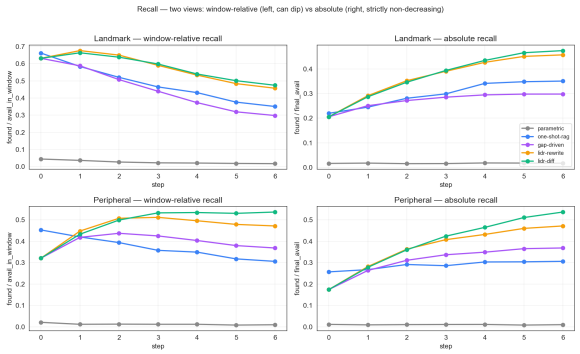

Figure 4: Temporal evidence acquisition under the time gate. The trajectory plot illustrates denominator drift: window-relative recall can fall as the available set grows, while absolute recall shows that update systems keep accumulating both landmark and peripheral evidence.

retry/fallback on collapsed rewrites, deterministic patch application with missing-anchor retries for `lidr-diff`, persistent retrieval audit rows, and explicit D0 build/grade metadata.

6 Related Work

Living systematic reviews. The closest intellectual ancestor is the living systematic review paradigm from evidence-based medicine (Elliott et al., 2017). Living reviews were developed because conventional systematic reviews go stale: new trials appear, conclusions shift, and full re-reviews are too slow. The difference is that living systematic reviews are updated manually by expert teams; LiDR-Bench asks whether AI systems can perform and evaluate this update process.

Deep research benchmarks. Recent deep-research benchmarks evaluate whether AI systems can search broadly, synthesize evidence, and produce long-form reports (Li et al., 2025; Java et al., 2025; Guo et al., 2026; Zhang et al., 2026). These benchmarks are valuable, but they mostly share a one-shot assumption: ask a question, produce a document, score the document. LiDR-Bench changes the evaluation unit from a report to a time-gated trajectory.

Automatic survey generation. Automated survey generators such as STORM, AutoSurvey, Agentic AutoSurvey, SurveyG, and related systems use multi-agent or retrieval-heavy pipelines to produce structured surveys from large paper collections (Shao et al., 2024; Wang et al., 2024; Liu et al., 2025; Nguyen et al., 2025). These systems are natural baselines or components for LiDR-Bench, but they do not by themselves address temporal evolution: a survey generated today may already be incomplete next week.

Scientific RAG and literature synthesis. Scientific RAG systems such as OpenScholar and PaperQA-style systems answer scientific questions with citations over large literature collections (Asai et al., 2024; Skarlinski et al., 2024). They address the static version of the problem: given a question and today's corpus, produce a grounded answer. LiDR-Bench asks whether that answer can be maintained as the literature moves.

Temporal QA, streaming benchmarks, and memory. Temporal QA benchmarks evaluate whether systems answer questions under time constraints or over changing corpora (Liška et al., 2022; Kasai et al., 2023; Jang et al., 2022). LiDR-Bench differs by evaluating continual synthesis rather than isolated answers. Knowledge editing and incremental summarization update model weights or summaries directly (Meng et al., 2022, 2023; Chang et al., 2024; D'Souza et al., 2025); RAG, memory, and continual-learning systems retrieve or store information across time (Sukhbaatar et al., 2015; Zhong et al., 2024; Lopez-Paz and Ranzato, 2017; Parisi et al., 2019; De Lange et al., 2022; van de Ven et al., 2022). LiDR-Bench asks the same style of question at the level of research synthesis: the carried state is an explicit document whose claims, structure, and citations must remain useful.

Conclusion. LiDR-Bench poses living research as a measurable challenge: systems should be judged not only by whether they can write an impressive report today, but by whether they can keep a synthesis current tomorrow.

7 Limitations

Survey papers are imperfect ground truth. A survey bibliography is an expert selection, not the unique correct set of papers. Strict citation metrics may penalize systems that cite valid alternatives outside the survey bibliography, so absolute recall should be interpreted as recovery of the expert survey's view rather than total coverage of the field. Claim coverage mitigates this by evaluating substantive content, but it is still derived from the survey text.

The current survey cohort is ML/AI focused. LiDR-Bench currently uses ML/AI survey papers from arXiv. It would be interesting and relevant to repeat the methodology in fields such as biomedicine, policy, patents, or global events. Expanding beyond ML/AI is future work.

Claim coverage is LLM-judged. Our claim bank and LLM-as-judge graders make large-scale evaluation possible, but LLM judgments can be noisy. The benchmark should report grader prompts, models, and validation checks, and future work should add human audits or multi-judge reliability analyses.

Reference systems are not optimized. The included systems demonstrate benchmark signal and establish starting points. Stronger agents, better retrievers, learned editors, or multi-document systems may change the ranking.

Mechanism ablations remain future work. A cleaner follow-up would separate document continuity from evidence retention by comparing normal continuity against variants that reset document text at each timestep or preserve only the previous bibliography. This would test whether gains come from preserving conceptual organization, retaining citations, or simply expanding retrieval exposure.

8 Ethical Considerations

Living research systems could help researchers keep up with fast-moving fields, but they also carry risks. A maintained synthesis may appear authoritative even when it contains stale claims, missing evidence, or judge-approved but incorrect summaries. Systems used in sensitive domains could amplify errors into downstream decisions. Versioning, citation provenance, human review, and clear uncertainty reporting are therefore important deployment requirements.

There are also labor and access considerations. Automating parts of expert synthesis may change who can produce literature reviews and who gets credited for synthesis work. Dataset release must respect copyright constraints around survey text and paper metadata; we plan to release code and derived benchmark artifacts under open licenses where permitted by upstream terms, without redistributing copyrighted full text. LiDR-Bench is built from public scholarly documents and metadata rather than private user data; released artifacts will be checked to avoid personal information beyond public bibliographic metadata and to exclude offensive or unsafe content not needed for the benchmark. Finally, continuous monitoring can be computationally expensive; benchmark reports should include token, latency, and cost accounting so researchers can reason about environmental and financial cost.

References

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, and 6 others. 2024. OpenScholar: Synthesizing scientific literature with retrieval-augmented LMs. *Preprint*, arXiv:2411.14199.

Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. BooookScore: A systematic exploration of book-length summarization in the era of LLMs. In *International Conference on Learning Representations*.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. 2022. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385.

Valentina D'Souza, Alaleh Azhir, Danielle F. Pace, Samuel Friedman, Arash Nargesi, Tristan Naumann, Christopher D. Anderson, Steven J. Atlas, Judy Hung, and Mahnaz Maddah. 2025. CLIN-SUMM: Temporal summarization of longitudinal clinical notes. medRxiv preprint.

Julian H. Elliott, Anneliese Synnot, Tari Turner, Mark Simmonds, Elie A. Akl, Steve McDonald, Georgia Salanti, Joerg Meerpohl, Harriet MacLehose, John Hilton, David Tovey, Ian Shemilt, James Thomas, and Living Systematic Review Network. 2017. Living systematic review: 1. introduction—the why, what, when, and how. *Journal of Clinical Epidemiology*, 91:23–30.

Beichen Guo, Zhiyuan Wen, Jia Gu, Senzhang Wang, Haochen Shi, Ruosong Yang, and Shuaiqi Liu. 2026. SurveyLens: A research discipline-aware benchmark for automatic survey generation. *Preprint*, arXiv:2602.11238.

Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. 2022. TemporalWiki: A lifelong benchmark for training and evaluating ever-evolving language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6237–6250, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Abhinav Java, Ashmit Khandelwal, Sukruta Midigeshi, Aaron Halfaker, Amit Deshpande, Navin Goyal, Ankur Gupta, Nagarajan Natarajan, and Amit Sharma. 2025. Characterizing deep research: A benchmark and formal definition. *Preprint*, arXiv:2508.04183.

Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Velocity Yu, Dragomir Radev, Noah A. Smith, Yejin Choi, and Kentaro Inui. 2023. RealTime QA: What's the answer right now? In *Advances in Neural Information Processing Systems*. Datasets and Benchmarks Track.

Minghao Li, Ying Zeng, Zhihao Cheng, Cong Ma, and Kai Jia. 2025. ReportBench: Evaluating deep research agents via academic survey tasks. *Preprint*, arXiv:2508.15804.

Adam Liška, Tomáš Kočiský, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, Cyprien de Masson d'Autume, Tim Scholtes, Manzil Zaheer, Susannah Young, Ellen Gilsenan-McMahon, Sophia Austin, Phil Blunsom, and Angeliki Lazaridou. 2022. StreamingQA: A benchmark for adaptation to new knowledge over time in question answering models. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 13402–13426. PMLR.

Yixin Liu, Yonghui Wu, Denghui Zhang, and Lichao Sun. 2025. Agentic AutoSurvey: Let LLMs survey LLMs. *Preprint*, arXiv:2509.18661.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *International Conference on Learning Representations*.

Minh-Anh Nguyen, Minh-Duc Nguyen, Ha Lan N. T., Kieu Hai Dang, Nguyen Tien Dong, and Dung D. Le. 2025. SurveyG: A multi-agent LLM framework with hierarchical citation graph for automated survey generation. *Preprint*, arXiv:2510.07733.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71.

Yijia Shao, Yucheng Jiang, Theodore A. Kanell, Peter Xu, Omar Khattab, and Monica S. Lam. 2024. Assisting in writing Wikipedia-like articles from scratch with large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 6252–6278, Mexico City, Mexico. Association for Computational Linguistics.

Michael D. Skarlinski, Sam Cox, Jon M. Laurent, James D. Braza, Michaela Hinks, Michael J. Hammerling, Manvitha Ponnapati, Samuel G. Rodriques, and Andrew D. White. 2024. Language agents achieve superhuman synthesis of scientific knowledge. *Preprint*, arXiv:2409.13740.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-To-End Memory Networks. In *Advances in Neural Information Processing Systems*, pages 2440–2448. ArXiv:1503.08895.

Gido M. van de Ven, Tinne Tuytelaars, and Andreas S. Tolias. 2022. Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197.

Yidong Wang, Qi Guo, Wenjin Yao, Hongbo Zhang, Xin Zhang, Zhen Wu, Meishan Zhang, Xinyu Dai, Min Zhang, Qingsong Wen, Wei Ye, Shikun Zhang, and Yue Zhang. 2024. AutoSurvey: Large language models can automatically write surveys. In *Advances in Neural Information Processing Systems*.

Hongzhi Zhang, Yuanze Hu, Tinghai Zhang, Jia Fu, Tao Wang, Junwei Jing, Zhaoxin Fan, Qi Wang, Ruiming Tang, Han Li, Guorui Zhou, and Kun

Gai. 2026. DeepSynth-Eval: Objectively evaluating information consolidation in deep survey writing. *Preprint*, arXiv:2601.03540.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. MemoryBank: Enhancing Large Language Models with Long-Term Memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19731.

A Dataset Construction Details

Dataset construction has five stages. First, we collect candidate 2026 computer-science survey papers from Semantic Scholar, arXiv, and OpenAlex. The selection pass uses `gemini/gemini-2.5-flash` for cheap labeling and cluster ranking, `gemini/gemini-embedding-001` for topic clustering, and `gemini/gemini-2.5-pro` for the final discriminating survey score. The goal is a topic-balanced cohort of recent surveys whose references are mostly arXiv-linked and date-resolved.

Second, each accepted survey is enriched into a task instance. We fetch the survey metadata and full text, extract references with Semantic Scholar / GROBID / arXiv fallbacks, backfill month-precision publication dates, and reject surveys that do not meet the fully-enriched contract: at least 10 landmark references, at least 2 anchor references, at least 5K characters of full text, arXiv-linked references, and month-precision reference dates. The enrichment LLM classifies references into landmark/peripheral sets, selects 3–5 pre-t_0 anchors when possible, and extracts the `user_query`.

Third, we build frozen retrieval pools. Pool construction uses the survey title, `user_query`, and anchor titles to retrieve topical distractors, with optional query expansion. The released 2026 pools target roughly 10K topical distractors per survey, require at least 1K visible distractors per timestep during construction, and are later subset at runtime to a cap of 1K distractors with the stratified-time policy used in §5.1.

Fourth, we extract the claim bank from each target survey. Claim extraction uses `gemini/gemini-3.1-pro-preview` in a three-pass procedure: identify recency-gated capability areas, extract gaps and failures per kept area, and extract cross-area connections. The extracted records are then deterministically projected into the four scored axes used in §3.2: `broad_sota`, `delta`, `failure`, and `frontier`. The release preserves the richer JSON provenance record so claims can be audited or reprojected without rerunning the LLM.

Fifth, runtime evaluation never gives the system the target survey. The target survey is excluded from its retrieval pool, all candidate evidence is filtered by timestep publication date, and the final timestep still ends before the survey date. Claim-coverage grading assigns `present`, `partial`, or `absent` verdicts, scored as 1.0, 0.5, and 0.0.

As a concrete example, one `failure` claim extracted for the Adaptive Reasoning survey (`2511.10788`) says: "During long-chain reasoning tasks, simple halting mechanisms fail because minor errors accumulate into hallucinations, requiring active backtracking." In the benchmark this lives inside a larger JSON claim record with fields such as axis, capability, and temporal scope; the sentence here is the human-readable claim text. A system only gets credit if the maintained document makes this claim inferable, not merely if it cites papers from the survey bibliography.

B Reference System Prompts

All reported reference systems use the same underlying synthesis primitives unless noted otherwise. The main run uses `gemini/gemini-2.5-flash` for synthesis and updates, provider-aware Gemini `max_tokens=65,536` for normal synthesis/update calls, a soft document target of 20K characters, and a hard cap of 30K characters. The `parametric` memory-only probe caps generation at 8K tokens. Rewrites that collapse below guardrail thresholds are retried or rolled back; `lidr-diff` patches are applied deterministically with at most 8 patches and 4K inserted characters per timestep.

The initial synthesis prompt asks the system to create a knowledge map that helps readers understand what capabilities and methods exist in the space, how different approaches relate to each other, what tradeoffs practitioners face, and where the field is heading. The rewrite prompt asks the system to add papers

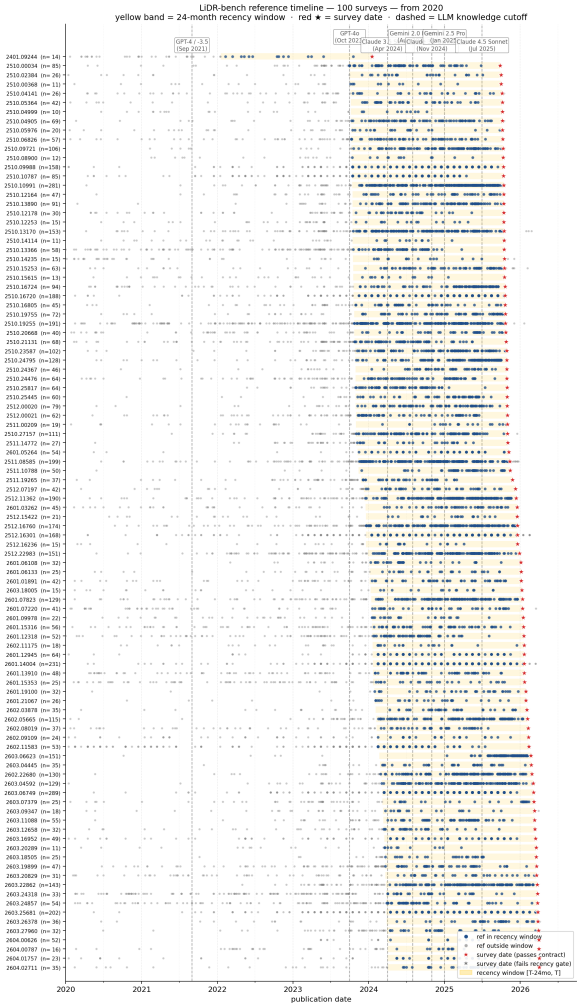

Figure 5: Reference timeline for the 100-survey 2026 cohort, shown from 2020 onward. Each row is a survey instance. Yellow bands mark the 24-month recency window before the survey date, blue points are references inside that window, gray points are older references, red stars are survey dates, and dashed vertical lines mark representative frontier-model knowledge cutoffs.

that meaningfully advance understanding, remove or replace citations that are superseded, tangential, or redundant, and keep the document tight. The markdown-patch prompt asks `lidr-diff` to issue ordered `replace`, `insert_after`, or `delete` patches under strict patch and inserted-character budgets.

C Additional Results

Table 4 shows the achievable-recall funnel for the main run. The pool and time-gate ceilings are effectively complete: the final timestep has 118.48 scorable gold references per survey on average, all of them are present in the frozen pools, and 117.89 are date-visible before the survey cutoff. Recall is capped downstream by retrieval and selection: top-50 retrieval returns 48.82 gold references on average, and top-24 evidence selection passes 24.09 gold references to synthesis at a single timestep. The trajectory-selection rows use saved `selected_ids` metadata from the stateful systems.

Table 3: Prompt families for reported reference systems.

System	Retrieval/query prompt	Synthesis/update prompt
`parametric`	none	memory-only survey prompt
`one-shot-rag`	static `user_query`	initial synthesis prompt at every timestep
`lidr-rewrite`	static `user_query`; selector sees prior document at update steps	initial synthesis at `t=0`, rewrite prompt for `t>0`
`lidr-diff`	static `user_query`; selector sees prior document and excludes already cited papers	initial synthesis at `t=0`, markdown-patch prompt for `t>0`
`gap-driven`	query-generation prompt at `t=0`; gap-query prompt over prior document for `t>0`	initial synthesis at `t=0`, rewrite prompt for `t>0`

Table 4: Achievable-recall funnel.

Quantity	Mean ± SE	Scope
Survey gold references	118.48 ± 7.77	target
Gold in frozen pool	118.48 ± 7.77	pool
Gold date-visible at final step	117.89 ± 7.70	time gate
Gold retrieved in top-50	48.82 ± 0.37	retrieval
Gold selected for synthesis	24.09 ± 0.13	selection
`gap-driven`: unique selected gold / final gold cited	35.19 ± 0.50 / 28.16 ± 0.61	trajectory
`lidr-rewrite`: unique selected gold / final gold cited	48.42 ± 0.37 / 39.76 ± 0.75	trajectory
`lidr-diff`: unique selected gold / final gold cited	48.40 ± 0.38 / 43.10 ± 0.73	trajectory

C.1 Document Length Diagnostic

C.2 Query Grounding Ablation

Table 5 reports the 100-survey query-grounding ablation summarized in §5.1. The view uses combined landmark+peripheral ground truth, cumulative scope, and non-initial timesteps, giving roughly 600

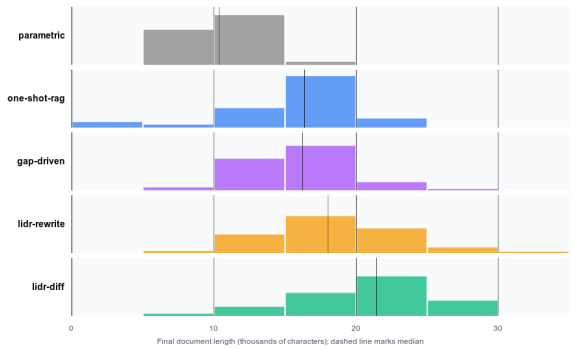

Figure 6: Final document length distributions by method. Each panel shows per-survey final document length in 5K-character bins; dashed lines mark medians.

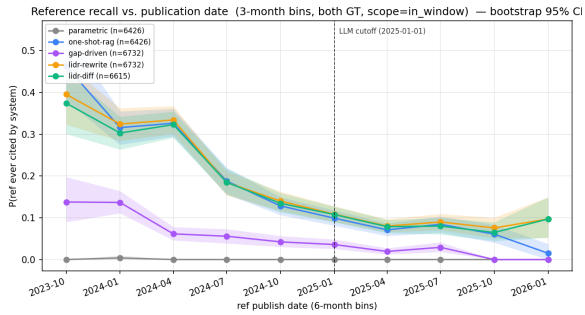

Figure 7: Per-reference recall as a function of publication date, restricted to references inside each survey's 24-month retrieval window. The `parametric` curve is flat at floor across the entire window; retrieval-grounded systems decline smoothly from the older edge of the window to the newer edge with no inflection at the cutoff date.

survey-step observations per strategy.

C.3 Contamination Probe

§3 establishes that the `parametric` baseline, which generates only from model memory, recovers essentially none of the newly-available ground-truth papers. This appendix extends that evidence by examining recall as a function of the cited reference's own publication date. The hypothesis is simple: if benchmark performance were inflated by training-data leakage, a memory-only system should recall references published before the synthesis model's training cutoff at a meaningfully higher rate than references published after.

Three observations follow. First, `parametric` recall on in-window references is at floor regardless of cutoff position: 0.1% before the LLM cutoff and 0.0% after. Second, the retrieval-grounded systems' pre/post differences reflect the fact that very recent references are sparser in the pool and harder to retrieve; the four retrieval systems share the same shape. Third, the smooth decline through the cutoff date with no inflection is what we would expect under no leakage and is not what we would expect under memorization. Taken together with `rec_new = 0.000` for `parametric`, the evidence supports the conclusion that the benchmark's discriminative signal is driven by retrieval and update behavior, not by training-data overlap.

C.4 Qualitative Trajectories

The adaptive-reasoning trajectory's field motion is a shift from generic efficiency controls to trained, budget-aware reasoning policies. At initialization, a good report should frame Chain-of-Thought, self-consistency, routing, MoE-style conditional compute, and older ideas about spending more inference on harder problems, but should not invent a 2025 budget-RL taxonomy. By the final checkpoint, the field's center has moved from "compress reasoning" to policy-level budget control: Self-Budgeter, BRPO/AnytimeReasoner, LASER, ThinkSwitcher, Plan-and-Budget, Route-to-Reason, Mixture of Reasonings, HBPO, SABER, BudgetThinker, and related entropy or convergence signals. A gold report should explain how these methods let the model internalize when to reason longer, when to compress, and how to expose controllable compute budgets to users.

The on-policy-distillation trajectory is more mixed: stateful updates preserve the useful taxonomy, but the hardest failure-mode claims remain underdeveloped. At initialization, a good report should explain OPD as a response to exposure bias: students should train on their own generated prefixes, not only teacher-forced data. By the final checkpoint, a gold report should not just list methods. It should separate token-adaptive and selective token-weighted KD; mixed-policy and Box OPD; stable OPD; self-distillation; privileged-information methods such as GATES; reasoning-transfer methods; and failure modes such as policy saturation, epistemic-verbalization loss, and RLVR cliff prompts.

Table 5: Query grounding ablation. Δ recall@50 and Δ ROC-AUC are paired against the static user query; win rate is the fraction of paired survey-step cases where recall@50 is at least as high as the static query.

Strategy	recall@10	recall@50	recall@100	ROC-AUC	Δ recall@50	Δ ROC-AUC	win%@50
Raw prior document (`one-shot-rag`)	0.422 ± 0.304	0.731 ± 0.258	0.867 ± 0.205	0.780 ± 0.196	$+0.077 \pm 0.220$	$+0.059 \pm 0.133$	88%
Raw prior document (`lidr-diff`)	0.421 ± 0.310	0.725 ± 0.257	0.868 ± 0.203	0.780 ± 0.196	$+0.072 \pm 0.224$	$+0.059 \pm 0.137$	86%
Three LLM-generated queries (`lidr-diff`)	0.358 ± 0.290	0.711 ± 0.252	0.859 ± 0.184	0.765 ± 0.172	$+0.058 \pm 0.252$	$+0.044 \pm 0.164$	77%
Raw prior document (`parametric`)	0.375 ± 0.302	0.698 ± 0.264	0.851 ± 0.208	0.754 ± 0.200	$+0.044 \pm 0.202$	$+0.034 \pm 0.119$	82%
Static title + abstract	0.361 ± 0.291	0.686 ± 0.270	0.844 ± 0.212	0.744 ± 0.199	$+0.029 \pm 0.221$	$+0.021 \pm 0.129$	78%
Static user query	0.337 ± 0.289	0.657 ± 0.274	0.825 ± 0.218	0.724 ± 0.202	–	–	–