From Intent to Evidence: A Categorical Approach for Structural Evaluation of Deep Research Agents

Shuoling Liu[1,2], Zhiquan Tan[2,*], Kun Yi[2], Hui Wu[2], Yihan Li[2], Jiangpeng Yan[2], Liyuan Chen[2], Kai Chen[1], Qiang Yang[3]

[1]The Hong Kong University of Science and Technology
[2]E Fund Management Co., Ltd.
[3]The Hong Kong Polytechnic University

Abstract

Deep Research Agents (DRAs) aim to answer complex questions by searching the web, checking evidence, and synthesizing conclusions across heterogeneous sources. We introduce a category-theoretic framework for evaluating and improving such agents, treating deep research as a structured mapping from user intent to evidence-grounded conclusions, making retrieval traces, cross-source alignment, and final synthesis explicit. Guided by this view, we derive a mechanism-aware benchmark of 296 bilingual questions targeting four structural skills: following multi-hop evidence chains, verifying claims across sources, re-ordering fragmented information, and rejecting unsupported assumptions. We evaluate 16 frontier systems with human verification and find that these structural tasks remain highly challenging: the best system reaches only 19.9% average accuracy. Strong agents can sometimes reorganize evidence and detect false premises, but still struggle with long-horizon synthesis and intersection-heavy verification. The same theory also leads to practical improvements: tracked search (which preserves retrieval traces) and category tools (which add explicit verification and synthesis steps) measurably improve API-based deep research systems. Our work provides both a challenging benchmark and concrete design guidance for more reliable research agents.[2]

1 Introduction

The recent leap in large language models (LLMs) has sparked the rapid development of autonomous agents, with Deep Research Agents (DRAs) emerging as a practical solution for complex tasks (Wu et al., 2025). Unlike early systems confined to static knowledge recall (Wei et al., 2025; Zhou et al., 2025; Chen et al., 2025), true deep research is inherently dynamic (Nie et al., 2024): an agent must actively seek out, cross-validate, and stitch together information from diverse and frequently conflicting sources (Hu et al., 2025), filtering noise, resolving contradictions, and synthesizing fragmented evidence into well-grounded judgments (Du et al., 2025).

However, a rigorous theoretical modeling of deep research agent behavior remains conspicuously absent. Without a formal mathematical foundation, current evaluation paradigms rely on purely empirical and ad-hoc designs (Xu & Peng, 2025) that blur genuine reasoning, basic retrieval, and simple memorization (Shi et al., 2025); moreover, their queries rarely pose the difficulty needed to genuinely stress-test long-horizon synthesis and ambiguity resolution. The community therefore lacks a theoretically grounded benchmark combining systematic capability verification with appropriately

*Corresponding author.
[2]Our implementation will be available at https://github.com/tzq1999/CDR.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

challenging tasks—a gap that calls for shifting from fragmented empirical testing toward a formal framework that first models, and then evaluates, the agent's entire research ability.

We formalize the behavior of the Deep Research Agent through the lens of category theory (Mac Lane, 2013), since the deep-research workflow is fundamentally a structural mapping: an abstract intent is mediated through a noisy external web and transformed into a logically constrained conclusion space. This perspective models the workflow as a composition of structure-preserving maps (functors), enabling precise, testable statements about agent behaviours. The framework is *normative* rather than descriptive: the propositions below verify the internal consistency of an ideal reference model—real agents need not be functors; their failure to preserve the ideal structure is exactly what we measure. Grounded in this framework, we construct a challenging benchmark of 296 bilingual questions that adversarially stress-tests agents along four interpretable axes: traversing sequential connectivity chains, verifying intersections within V-structure pullbacks, imposing topological ordering on retrieved substructures, and performing ontological falsification via the Yoneda Probe (illustrated in Figure 6, Appendix F).

We systematically benchmark 16 frontier systems—reasoning models, search-augmented systems, closed-source deep research products, and our own theory-guided API-based variants—using a strict, human-verified evaluation pipeline. The best system reaches only 19.9% average performance, and the results reveal a sharp capability split: advanced pipelines increasingly handle dynamic topological re-ordering and exhibit robust ontological verification, yet almost universally collapse on multi-hop structural synthesis and exhibit blind spots under specific mathematical constraints. The large variance across systems and task families indicates brittle heuristics rather than unified structural understanding.

Importantly, these findings are not only diagnostic. Because our framework factorizes deep research into retrieval structure and evidence-grounded synthesis, it also suggests concrete system interventions, which we instantiate as two practical modules. First, *tracked search* turns latent browsing jumps into explicit, auditable retrieval-graph edges, preserving the provenance needed by the Search Functor. Second, *category tools* implement pullback-style verification and colimit-style synthesis, so that cross-source claims are merged only when they admit evidentiary pre-images and provenance-preserving aggregation, directly targeting failures of the Reasoning Functor. In controlled API-based systems, adding category tools on top of tracked search improves GPT-5 from 45.68% to 47.53% on DeepResearchBench and from 10.48% to 13.29% on our benchmark, with similar gains for Grok-4.1-fast. The theory thus functions as both an evaluation language and a design blueprint for stronger deep research agents.

This paper makes three contributions. First, we provide a category-theoretic formulation of deep research in terms of the Search Functor, the Reasoning Functor, pullbacks, colimits, and Yoneda-style falsification, turning informal workflow intuitions into precise structural claims. Second, we instantiate this theory as a challenging benchmark and a human-verified evaluation protocol that expose concrete capability bottlenecks beyond broad report quality; the benchmark is a complementary, theory-guided stress test whose strict binary criterion requires every specified entity, date, subset, comparison, or null constraint to be satisfied, so that fluent or lengthy presentation cannot compensate for a broken structural dependency. Third, we translate the theory into concrete design modules—tracked search preserves retrieval traces, while categorical verification and synthesis operators improve evidence alignment and aggregation—yielding measurable gains in API-based implementations. Top-tier agents increasingly unify search and reasoning, yet generalized mastery over complex structural topologies remains a formidable open challenge.

2 Preliminary

We briefly recap the categorical notions used below; the concepts are standard (Mac Lane, 2013), and the formal definitions are reproduced in Appendix A for completeness. A *category* $\mathcal{C}$ consists of objects and morphisms equipped with an associative composition law and identity morphisms; in our context, objects correspond to state spaces (queries, web entities, propositions) and morphisms to structure-preserving transitions between them. A *functor* $F : \mathcal{C} \to \mathcal{D}$ maps objects and morphisms between categories while preserving composition and identities. Given morphisms $f : X \to Z$ and $g : Y \to Z$, their *pullback* is the most general object P projecting to X and Y such that the two composites agree—in the category of sets, $P = \{(x, y) \in X \times Y \mid f(x) = g(y)\}$—formalizing the

alignment of two information sources under a shared constraint. *Limits* and *colimits* generalize this pattern: limits aggregate a whole diagram into the maximal jointly consistent object, while colimits "glue" fragments along shared structure. These four notions suffice for our formulation: categories model the agent's state spaces, functors model structure-preserving search and reasoning, pullbacks model multi-source verification, and colimits model report synthesis.

3 Categorical Formulation of Deep Research Agents

We formalize the end-to-end workflow of a Deep Research Agent (DRA) as a structural mapping from an abstract problem space to a structured conclusion space, mediated by the Web. Category theory strictly defines the requisite capabilities of an ideal DRA through standard constructs, and provides a rigorous vocabulary to diagnose common failure modes. Detailed proofs are deferred to Appendix B; see Figure 1.

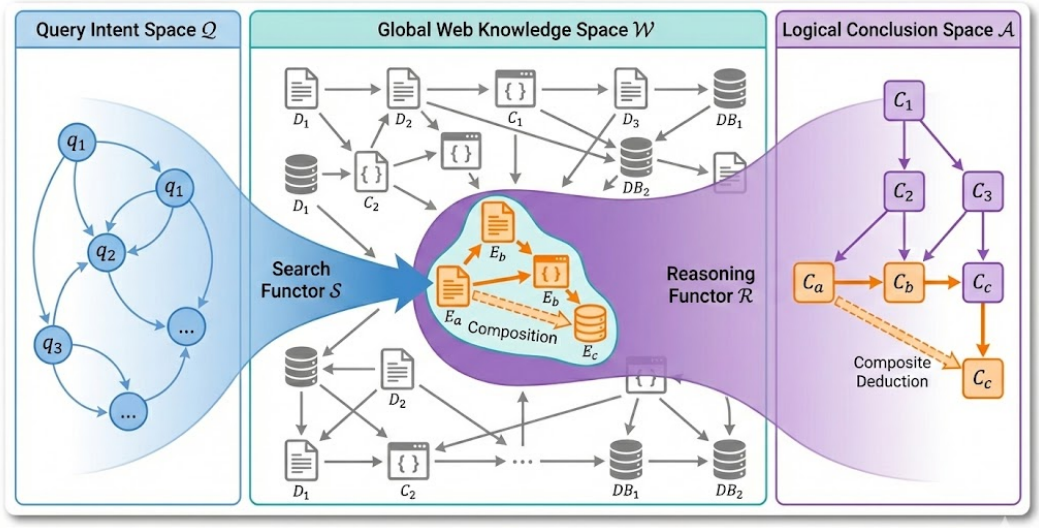


Figure 1: Categorical view of the deep research agent workflow.

3.1 Agent State Spaces as Categories

We first establish the state-space categories, specifying objects (Ob) and morphisms (Hom) for each conceptual phase.

- **The Intent Category $\mathcal{Q}$ (Query Space):** objects are user queries, sub-problems, or atomic intent units; a morphism $q_1 \xrightarrow{\text{dep}} q_2$ denotes *task dependency*—resolving q_2 strictly depends on resolving q_1—formalizing the agent's task decomposition and planning graph.
- **The Knowledge Category $\mathcal{W}$ (Web Space):** objects are informational entities on the Web (web pages, documents, code snippets, database records); a morphism $w_1 \xrightarrow{\text{ref}} w_2$ denotes a *structural or evidentiary linkage*: a hyperlink, or content of w_1 objectively acting as a supporting premise for w_2.
- **The Retrieval Subcategory $\mathbf{Sub}(\mathcal{W})$ (Retrieved Context):** since LLMs have finite context windows, the agent interacts with a localized retrieval subspace whose objects are retrieved knowledge subgraphs $G \subseteq \mathcal{W}$ and whose morphisms are *inclusion mappings* ($G_1 \hookrightarrow G_2$), formalizing the monotonic accumulation of evidence.
- **The Reasoning Category $\mathcal{A}$ (Logic/Answer Space):** objects are propositions, intermediate arguments, or final statements; a morphism $a_1 \xrightarrow{\text{implies}} a_2$ denotes strict *logical entailment* ($a_1 \implies a_2$).

Theorem 3.1 (Categorical Validity of Agentic State Spaces)**.** *Let $\mathcal{Q}$, $\mathcal{W}$, $\mathbf{Sub}(\mathcal{W})$, and $\mathcal{A}$ be the agentic state spaces equipped with their respective objects and relations. If we define the morphisms*

in $\mathcal{Q}$, $\mathcal{W}$, and $\mathcal{A}$ as above, then all four spaces strictly satisfy the Eilenberg-Mac Lane axioms and thus constitute mathematically valid categories.

These state spaces let us model deep research as a sequence of structure-preserving functors bridging $\mathcal{Q}$, $\mathbf{Sub}(\mathcal{W})$, and $\mathcal{A}$, concretely defining the agent's core capabilities of searching and reasoning.

3.2 Agent Workflows as Functors

Search Functor We formalize the (ideal) search operation as a functor $S : \mathcal{Q} \to \mathbf{Sub}(\mathcal{W})$ that maps an abstract "problem flow" to a concrete "evidence flow"; an ideal search functor must preserve morphisms and composition.

Definition 3.2 (Morphism Mapping and Dependency Preservation)**.** For any query dependency $f : q_1 \to q_2$ in $\mathcal{Q}$, the Search Functor must map it to a valid morphism $S(f) : S(q_1) \to S(q_2)$ within the retrieval subspace. Concretely, if q_2 logically depends on q_1, the retrieved material subobject $S(q_2)$ should extend $S(q_1)$ via a monomorphism $\iota : S(q_1) \hookrightarrow S(q_2)$ (Remark B.1 in Appendix B): the agent should preserve and make available its entire access history (provenance) rather than restarting from a disconnected lookup.

Proposition 3.3 (Chain-of-Search Consistency)**.** *An ideal deep research agent preserves compositionality during multi-hop retrieval (deep search). For a sequence of sub-task dependencies $q_1 \xrightarrow{f} q_2 \xrightarrow{g} q_3$, it strictly holds that $S(g \circ f) = S(g) \circ S(f)$.*

The Reasoning Functor The logical synthesis process is formalized as an ideal functor $R : \mathcal{W} \to \mathcal{A}$, mapping unstructured entities from the web knowledge space $\mathcal{W}$ into structured propositions in the abstract logic space $\mathcal{A}$ (in practice, acting upon the retrieved evidence subgraph $\mathrm{Im}(S)$).

Definition 3.4 (Morphism Mapping and Causality Preservation)**.** For any evidentiary morphism $f \in \mathrm{Hom}_{\mathcal{W}}(w_1, w_2)$—such as a hyperlink or an objective causal premise—the reasoning functor must produce a valid logical implication $R(f) \in \mathrm{Hom}_{\mathcal{A}}(R(w_1), R(w_2))$. Consequently, for any such pair (w_1, w_2), the functor induces a mapping between hom-sets: $R_{w_1,w_2} : \mathrm{Hom}_{\mathcal{W}}(w_1, w_2) \to \mathrm{Hom}_{\mathcal{A}}(R(w_1), R(w_2))$. This ensures that the generated logic is strictly anchored to objective external evidence.

Remark 3.5 (Formal Characterization of **Hallucination**). In categorical terms, an agent hallucinates precisely when the induced mapping R_{w_1,w_2} is not surjective (not full functor): asserting a morphism $h \in \mathrm{Hom}_{\mathcal{A}}(R(w_1), R(w_2))$ with $h \notin \mathrm{Im}(R_{w_1,w_2})$ fabricates a logical implication without any valid evidentiary pre-image. The extreme case is $\mathrm{Hom}_{\mathcal{W}}(w_1, w_2) = \varnothing$—no evidentiary link whatsoever—yet the agent still asserts h, generating deductive claims out of thin air.

Proposition 3.6 (Functoriality and Transitivity of Deduction)**.** *By the composition axiom of functors, an ideal reasoning agent inherently preserves deductive transitivity. For any composable evidentiary morphisms $w_1 \xrightarrow{f} w_2 \xrightarrow{g} w_3$ in $\mathcal{W}$, it strictly holds that $R(g \circ f) = R(g) \circ R(f)$.*

Report Generation The final report for a query q should be the *colimit* of the graph of partial arguments produced by the Reasoning Functor R over the retrieved evidence $S(q)$: $Rep(q) \cong \mathrm{colim} R(S(q))$ (Mac Lane, 2013). If a generated report fails to become a colimit—for instance, if conflicting conclusions remain unresolved, or if fabricated summaries are added—the deep research process is non-natural during the aggregation stage.

4 Design of Evaluation

The inherent scale and topological complexity of the real-world knowledge and logic categories challenge the strict functoriality of an agent's Search and Reasoning operations. To systematically quantify an agent's capacity for structural preservation against context truncation and hallucination, we propose a four-dimensional categorical evaluation framework: (i) *Connectivity via Simple Chains* evaluates linear, transitive reasoning across sequential information nodes; (ii) *Consistency via V-Structures (Pullbacks)* assesses multi-source verification, intersection finding, and conflict resolution; (iii) *Organization via Substructure Re-ordering* tests the disentangling of complex dependencies into ordered flows; and (iv) *Structure Validity via the Yoneda Probe* verifies the ontological reality of entities by probing their external relationships rather than internal attributes. Table 4 and full descriptions of the fifteen task paradigms below appear in Appendix F.

Labeling protocol and boundary cases. The four type labels identify the *primary structural bottleneck* of a query rather than asserting that a task contains only one operation. Each query was assigned to exactly one type by two annotators following explicit inclusion/exclusion rules, with disagreements resolved by discussion: Type II requires aligning two or more disjoint evidence arms under a shared entity, date, unit, or constraint (a pullback-style intersection), whereas Type III requires first constructing and preserving a filtered or strictly ordered substructure before any downstream computation. Boundary cases follow the locus of the decisive operation: Task 11 does involve aligning NAV values across sources, but its decisive operation is preserving the ranked Top-20 set before aggregation, so it is Type III; conversely, Task 4 aligns disjoint clues without constructing any globally ordered subset, so it is Type II. Type IV is reserved for validity probing bounded to claims attributed to named, verifiable authorities, so that falsification targets a checkable ground truth.

4.1 Type I: Connectivity via Simple Chains

Type I tasks follow the principle of *Retrieval-Verification Asymmetry*: the forward search path (Problem $\rightarrow$ Answer) traverses an obscure, step-by-step trajectory in which each node entity is found on a single webpage, while backward verification is straightforward.

Identity Resolution via Textual Fingerprints (Task 1). Resolve the author of a semantically generic yet syntactically unique text snippet and cross-reference their platform activity (e.g., GitHub contributions on a specific UTC date); *adversarial lexical selection* and *cross-domain hopping* (academic prose to structured commit logs) defeat keyword matching and search-engine shortcuts.

Hierarchical Filtering in Vertical Knowledge Bases (Task 2). Identify an entity (e.g., a fund company) satisfying a conjunction of quantitative constraints across disjoint temporal windows (e.g., Net Asset Value at T_1, Total Shares at T_2), then answer a secondary query on a dynamic attribute; this *vertical search* demands *schema awareness* (knowing *where* to look) and *temporal precision* that defeats memorized knowledge.

Inverse Entity Resolution via Numerical Fingerprinting (Task 3). Recover a fund manager's identity and distant career history (e.g., employer ten years prior) from obscure numerical rankings serving as a "numerical fingerprint"; *anti-shortcut engineering* samples long-tail ranks (e.g., 130th or 746th) to force raw database queries, and the *temporal multi-hop* link penalizes hallucination from current affiliations.

4.2 Type II: Consistency via V-Structures (Pullbacks)

Type II evaluates the agent's capacity to model V-structures, formally known as *pullbacks*: information from disjoint sources $(S_1, S_2, \ldots, S_n)$ must be synthesized to discover latent constraints, verify intersections, or detect conflicts. The core challenge is the *asymmetry of synthesis*: individual data points are trivial to retrieve, but recognizing their necessary intersection requires structural awareness.

Distributed Quantitative Synthesis (Task 4). Deduce a metric by synthesizing sparse, interdependent quantitative constraints scattered across non-adjacent nodes (e.g., disparate temporal reports); since the answer exists nowhere verbatim, this forces *implicit equation solving* over aligned dynamic schemas, and *anti-memorization via long-tail sampling* (e.g., the 7th largest holding of a fund) necessitates raw multi-document extraction.

Entity Resolution under Hard-Negative Interference (Task 5). Profile an entity defined strictly by a unique ID while a high-profile individual sharing the exact same name acts as a hard negative—stress-testing *parametric prior versus contextual fact*: anchoring on the deterministic pullback (matching the ID) versus hallucinating from the pre-training distribution (matching the famous name).

Implicit Boundary Condition Detection (Task 6). Discover and apply an unmentioned, domain-specific boundary condition that mathematically invalidates a standard instruction (e.g., a hypothetical trade exceeding a statutory limit)—*meta-cognitive rule triggering*—then retrieve real-time baseline data (e.g., net asset value) and flag the latent violation.

Cross-Domain Anomaly Intersection (Task 7). Detect that a benign-looking request covertly embeds partial, fragmented identifiers matching an entity in an external risk database (e.g., a fugitive's features in a routine customer-service scenario); although not explicitly instructed, compliance implicitly dictates a background check—a *fuzzy database join* testing *safety alignment override*.

Latent Dependency Satisfiability (Task 8). Recover a dependency conflict disguised in a dense narrative (e.g., a multi-year software setup timeline): *unstructured constraint extraction* plus historical compatibility matrices reveal latent mutual exclusivity, such as version conflicts rendering the described outcome physically impossible.

4.3 Type III: Organization via Substructure Re-ordering

Type III tests the agent's capacity to act as a structure processor—performing *substructure reordering*: constructing a strictly ordered subcategory from chaotic data and evaluating limits or tracing morphisms only within it. Since search engines return flat, relevance-based results, the agent must impose its own mathematical ordering on retrieved raw data and use it as an anchor for downstream reasoning.

Rank-Conditioned Temporal Multi-Hop (Task 9). Identify an entity by a dynamic ordinal rank (e.g., the 5th highest-performing asset) at time T_1, then jump to a volatile, state-dependent attribute at a distant T_2; engines index static names rather than daily dynamic ranks, so the Top-N substructure must be self-constructed, and choosing T_2 right after a state transition (e.g., right after a manager resigns) enforces *temporal state precision*.

Range-Bounded Topological Sorting (Task 10). Filter a massive cross-sectional dataset by a continuous floating-point interval $[a, b]$ and dynamically generate a strict ordinal ranking—a query engines cannot execute natively—demanding *algorithmic exhaustiveness* and *granular float-point comparison* without omitting entities near the boundary conditions.

Top-K Substructure Aggregation and Synthesis (Task 11). Over a self-constructed, strictly ordered Top-K substructure (by a primary metric X, e.g., daily growth rate), compute macroscopic statistical invariants—mean, variance, or range—over a parallel metric Y (e.g., net asset value)—a *Map-Reduce cognitive emulation* that heavily probes *context window integrity* across many entities in working memory.

Rank-Anchored Time-Series Computation (Task 12). After identifying a target entity via its rank at a single timestamp T_{rank}, pivot to a continuous window $[T_{\text{start}}, T_{\text{end}}]$ for a longitudinal statistic (e.g., volatility of cumulative values); this *orthogonal dimension shifting* between cross-sectional and longitudinal retrieval demands strict schema isolation between the ranking metric and the time-series metric.

4.4 Type IV: Structure Validity via the Yoneda Probe

Type IV, rooted in the philosophy of falsification and the Yoneda Lemma (Mac Lane, 2013), evaluates the agent's capacity to confidently prove non-existence: if an entity yields null retrieval morphisms within the knowledge space $\mathcal{W}$, the agent must halt execution and refute the premise, rather than fabricating a hallucinated structure. Because LLMs inherently suffer from *sycophancy* and *premise acceptance*, we design three adversarial paradigms that test zero-shot resistance to such presuppositional traps.

Schema-Level Asymmetry and Null-Field Refutation (Task 13). Extract a highly specific quantitative metric (e.g., the asset scale of the largest underlying holding) across a batch of peer entities, such as the Q1 investment reports of five fund companies—where the reports exist for all entities, but the target field is structurally omitted by a subset; this induces *peer-pressure hallucination* toward symmetric output, and success requires *exhaustive null-verification* and explicitly reporting the $\emptyset$ state.

Macro-Ontological Invalidity via the Phantom Entity Probe (Task 14). Interpret the "core arguments" of a completely fabricated report attributed to a high-authority source (e.g., a Goldman Sachs report on "Quantum Encryption in Modern Agricultural Irrigation"), pitting *authority bias against ground truth*; we deliberately bound these probes to claims attributed to named, verifiable authorities so that non-existence is checkable, and upon finding zero legitimate references ($\text{Hom}(X, Y) = \emptyset$) to the exact report, the agent must execute a *Yoneda falsification*, deducing non-existence from this absolute absence of connectivity.

Asymmetric Comparative Hallucination (Task 15). Compare two highly correlated entities (e.g., the 2024 annual reports of CATL and EVE Energy) regarding a plausible latent feature that exists in

at most one of the reports; comparative instruction-following creates a strong gradient to hallucinate the missing feature—*instruction-forced feature alignment*—and to pass, the agent must break the relational presupposition and declare an asymmetric falsification, privileging factual topology over the requested format.

5 Experiments

We evaluate 16 leading mainstream models and AI products spanning three core competencies: (i) **Reasoning Models (5 products):** Gemini-3-pro-Reasoning (API) (Google, 2025), Claude-4.5-Sonnet-Reasoning (Anthropic, 2025), Doubao-Reasoning (API) (ByteDance, 2025), GPT-5-Reasoning (API) (OpenAI, 2025b), Qwen3-Reasoning (API) (Qwen Team, 2025); (ii) **Search-augmented Models (5 products):** Qwen3-Search (web) (Qwen Team, 2026), Doubao-Search (web) (ByteDance, 2025), Perplexity-Sonar-Pro-Search (Perplexity, 2025b), GPT-5-Search (web) (OpenAI, 2025b), Claude-4.6-Sonnet-Search (Anthropic, 2026); and (iii) **Deep Research Agents (6 products):** Doubao Deep Research (web) (ByteDance, 2025), Qwen Deep Research (web) (Qwen Team, 2026), Grok Deep Research (web) (xAI, 2025), OpenAI Deep Research (OpenAI, 2025a), Perplexity Deep Research (Perplexity, 2025a), Gemini-3-Pro deep Research (Google, 2025).

For task evaluation, we adopt a strict binary scoring metric (0 for failure, 1 for success). To mitigate the biases of automated evaluators while maintaining scalability, we implement a hybrid pipeline integrating "LLM-as-a-Judge" with human verification: GPT-5 first generates initial assessments from predefined rubrics ($\approx$95% agreement with experts on a random-sample audit); subsequently, a panel of 10 domain experts with finance and AI backgrounds manually reviews all preliminary scores, with at least two independent, model-blind reviews per item and disagreements resolved by discussion. All reported scores and rankings are based on the final expert-verified labels.

The 296 queries are distributed as 65 (Type I), 117 (Type II), 80 (Type III), and 34 (Type IV), yielding 4,736 system–query evaluations (16 systems $\times$ 296 queries). Because official deep-research interfaces are slow and quota-limited, we prioritized full-dataset, cross-model coverage over repeated runs, reporting a single complete run per system and interpreting all comparisons descriptively. Every potentially dynamic question specifies an explicit date, cutoff, or reporting window, so evaluation targets a fixed historical state rather than a live page. The full query set, reference answers, task IDs, judge prompts, and executable evaluation code are released in the supplementary material (further protocol details in Appendix D).

5.1 Evaluation Results

Figure 2 summarizes the results across the four task paradigms (type averages computed over all samples). While overall performance remains modest in absolute terms, the frontier is no longer concentrated in a single model family: Perplexity Deep Research attains the best Type I (10.8%) and Type II (32.7%) averages, Grok Deep Research leads Type III (26.3%) and ties Qwen3 w/Reasoning for the best Type IV score (46.9%), and Grok achieves the strongest Total Average (19.9%). The broader pattern is one of distributed specialization rather than universal dominance. Detailed per-task scores appear in Appendix E.

A note on statistical interpretation. We interpret the leaderboard descriptively, without claiming statistically significant fine-grained superiority: each system is evaluated once, so run-to-run variability is not captured. Under item resampling, a paired-item bootstrap over queries yields pointwise 95% CIs that exclude zero for the intervention differences (e.g., adding category tools to tracked search) across all four task families; such intervals capture item-resampling uncertainty only.

Structural synthesis remains difficult, but progress is no longer uniformly sparse. Type I and Type II tasks continue to expose the cost of retrieval-verification asymmetry: Type I remains especially challenging (best model 10.8%), while Type II shows clearer separation among search-heavy systems—Perplexity Deep Research (32.7%), Claude-4.6-Sonnet w/Search (28.6%), OpenAI Deep Research (20.4%)—suggesting that stronger retrieval orchestration yields tangible gains in structural verification.

Topological re-ordering and premise falsification favor different design choices. Type III continues to punish pure parametric reasoning: most reasoning-only models remain near zero, while

search-augmented and deep research systems perform markedly better (Doubao w/Search 16.3%, Grok Deep Research 26.3%), showing that explicit retrieval loops provide a more reliable substrate for re-ranking and aggregation. Type IV reveals even sharper variance: Grok Deep Research matches Qwen3 w/Reasoning at a benchmark-leading 46.9%, while other deep research systems remain far lower (Qwen Deep Research 4.1%, Gemini-3-Pro deep Research 5.0%)—rejecting false premises depends on careful system design rather than the mere presence of an agentic loop.

Frontier Deep Research systems are strong, but their strengths are highly specialized and task-local. The strongest systems divide the benchmark by capability: Perplexity Deep Research leads Type I and Type II, Grok Deep Research dominates Type III and shares the best Type IV score while retaining the highest total average, and OpenAI Deep Research occupies a strong middle ground (11.41 total average). These gains are sharply task-specific: in Type II, OpenAI Deep Research succeeds on Task 4 (11.1%) and Task 6 (10.0%) and Gemini-3-Pro deep Research breaks through on Task 8 (16.0%), yet no system is consistently strong across all five tasks; similarly, each of Tasks 9–12 is led by a different system. Current agents thus rely on narrow, task-local competencies rather than unified structural mastery. **Some hard configurations are beginning to crack, but true null-verification remains unresolved.** Task 13 (Type IV) still yields a universal 0.0% success rate—the clearest surviving blind spot—while Task 10 is no longer a universal collapse (Gemini-3-Pro deep Research 10.0%) and Task 12 sees breakthroughs (OpenAI and Perplexity Deep Research both 20.0%). The persistence of Task 13 suggests schema-level null verification remains a deeper challenge than ranking or temporal aggregation.

Disentangling Search and Reasoning. Beyond binary success, we provide a dual Search (S) / Reasoning (R) diagnostic allocating each task's primary bottleneck to retrieval versus reasoning difficulty (rubrics in Table 1): the S-score quantifies the adversarial complexity of the retrieval path, and the R-score reflects the cognitive load of integrating and reasoning over the retrieved evidence. The resulting profiles (Appendix C) reveal complementary frontiers: Perplexity Deep Research achieves the highest Search Score (12.9%), while Grok Deep Research attains the strongest Reasoning Score (21.6%). We emphasize that S and R are *indirect task-level diagnostics* derived from task design, not direct certifications of an agent's internal trajectory.

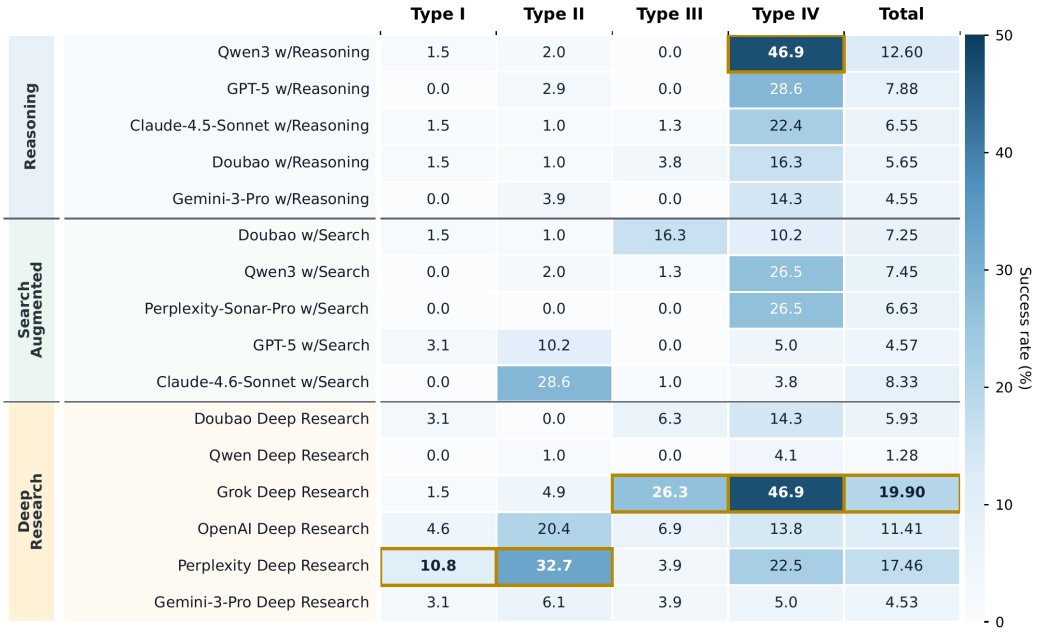


Figure 2: Mechanism-grouped performance heatmap across four structural task types. Columns report average success rates (%) for each task type and the total average; gold outlines mark the best score within each column.

6 Related Work

Category theory in AI. Category theory has provided mathematical foundations for various branches of AI (Shiebler et al., 2021), spanning backpropagation (Fong et al., 2019), probability (Fritz, 2020), conditional independence (Mahadevan, 2022a), learning paradigms (Harris, 2019; Mahadevan, 2022b), natural language (Bradley et al., 2022), foundation models (Yuan, 2023), and neural representations (Chen et al., 2023). While prior works build categories specific to those areas, we instead model the ideal behavior of deep research agents.

Deep research benchmarks. GAIA (Mialon et al., 2023), WebWalker (Wu et al., 2025), and WebShaper (Tao et al., 2025) measure analytical capabilities but largely bypass the research phase, while open-domain variants relying on static archives Bigeard et al., 2025 test internal memory rather than active search (Wei et al., 2025). Moving toward end-to-end autonomy, DeepResearchBench (Du et al., 2025) evaluates open-ended report generation, the BrowseComp series (Wei et al., 2025; Zhou et al., 2025; Chen et al., 2025) tests persistent web navigation, and FinSearchComp (Hu et al., 2025) designs time-sensitive financial tasks; none isolates structural failures through a theory-guided lens. Closest in spirit, LiveDRBench (Sharma et al., 2025) formalizes *which tasks constitute deep research* and evaluates claim completeness via precision/recall/F1, whereas we formalize *which structures an ideal research process should preserve* and derive controlled violations as benchmark families with scoring obligations and theory-guided interventions—complementary claim-coverage and structural-preservation axes.

7 From Theory to Practice: Design Implications for Deep Research Agents

Our framework serves not only as an evaluation lens but as a source of concrete design guidance: system quality should improve along the two structural axes of Section 2—the Search Functor S and the Reasoning Functor R. We implement a theory-informed pipeline that externalizes the workflow into inspectable artifacts: intent is decomposed into bounded research units, retrieval is materialized into persistent evidence objects, cross-source agreement is resolved through explicit categorical operators, and the report is produced only after provenance-preserving synthesis.

Search Functor in practice: why `tracked search` matters. Our implementation adopts a two-level architecture: a coordinator manages task decomposition and final synthesis, while researcher sub-agents execute web search, webpage reading, and PDF parsing. The first intervention is `tracked search`: every query is immediately written into a per-task evidence file (query string, timestamps, returned snippets, extracted URLs), turning search from a hidden jump into an observable edge in the retrieval graph. Per Definition 3.2 and Proposition 3.3, if q_2 depends on q_1, the system extends and preserves the existing retrieval state rather than overwriting it with a disconnected lookup—crucial for Type I and Type III tasks, where failures often stem from broken retrieval-chain composition.

Reasoning Functor in practice: why `category tools` matter. The second intervention concerns the Reasoning Functor R. Since Remark 3.5 characterizes hallucination as a logical morphism in $\mathcal{A}$ lacking a valid evidentiary pre-image in $\mathcal{W}$, we insert explicit `category tools` between evidence collection and final synthesis: a pullback-style verifier that asks which facts from two noisy, partially overlapping sources correspond to the same underlying object under a shared reference frame, and a colimit-style synthesizer that glues verified fragments into a report while preserving provenance and downgrading ungrounded claims. These tools operationalize pullbacks for multi-source consistency and colimits for report generation, targeting Type II and Type IV tasks.

Bounded autonomy and reflective control. We intentionally keep the control policy conservative: the coordinator defaults to a single researcher, increases parallelism only for clearly decomposable questions, and enforces hard caps on sub-agents and research rounds; each action is followed by an explicit reflection step assessing evidence sufficiency, unresolved contradictions, and remaining gaps. This is also theory-motivated: uncontrolled branching fragments the retrieval state and encourages premature aggregation.

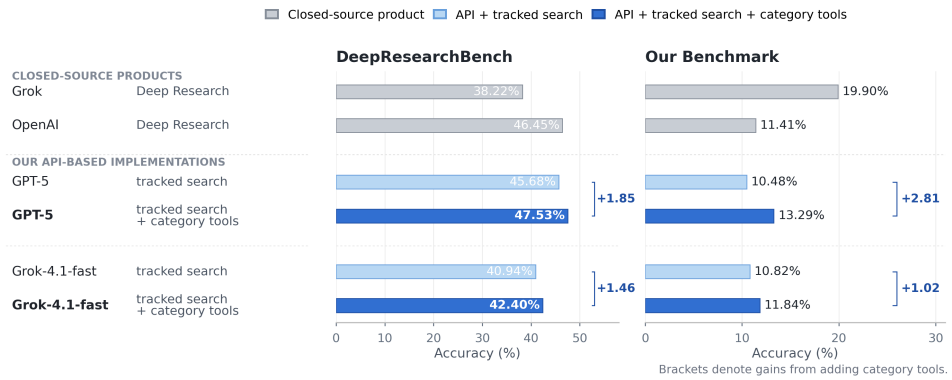


Figure 3: Performance comparison between DeepResearchBench and our benchmark. Grok Deep Research and OpenAI Deep Research are closed-source products; the remaining systems are our API-based implementations. Brackets indicate gains from adding category tools on top of tracked search.

Figure 3 admits two readings. First, the last four rows form a controlled ablation over our own API-based systems: `tracked search` intervenes on the Search Functor, while adding `category tools` inserts explicit pullback and colimit steps before answer generation. The gains are consistent across both model families: for GPT-5, category tools improve DeepResearchBench from 45.68% to 47.53% and our benchmark from 10.48% to 13.29%; for Grok-4.1-fast, from 40.94% to 42.40% and from 10.82% to 11.84%—as the theory predicts, preserving retrieval structure strengthens S, while explicit verification and gluing regularize R.

Second, the first two rows compare against closed-source frontier products. On the OpenAI side, our GPT-5-based variant with `tracked search & category tools` slightly surpasses OpenAI Deep Research on both DeepResearchBench (47.53% vs. 46.45%) and our benchmark (13.29% vs. 11.41%). On the Grok side, our Grok-4.1-fast variant exceeds Grok Deep Research on DeepResearchBench (42.40% vs. 38.22%), yet remains far below it on our benchmark (11.84% vs. 19.90%)—broad report quality and structural stress-test performance are not equivalent; the closed-source Grok product likely contains additional mechanisms for compositional and falsification-sensitive tasks.

Taken together, the large gap between DeepResearchBench and our benchmark shows that structurally grounded evaluation surfaces bottlenecks hidden by broad answer quality alone, while the controlled gains show that the theory identifies concrete design levers—`tracked search` for stabilizing S and `category tools` for regularizing R—with substantial headroom remaining on the hardest tasks.

8 Conclusion

We introduce a category-theoretic benchmark and structural account of deep research, replacing fragmented heuristics with systematic stress-testing along four axes: simple chains, V-structures, substructure re-ordering, and the Yoneda Probe.

Our human-verified evaluation of 16 frontier systems exposes the difficulty of formal structural stress-testing: the best system reaches only 19.9% average performance. While systems still struggle on sequential traversal and multi-source intersection, the strongest agents increasingly handle re-ordering and ontological verification. The message is not purely diagnostic: preserving retrieval traces and inserting pullback- and colimit-inspired operators yields measurable gains in controlled API variants.

Limitations. The formulation is normative: propositions verify the consistency of an ideal reference model, not behavioral properties of real agents. Each system was evaluated once, so run-to-run variability is not measured and differences should be read descriptively. The strict binary criterion penalizes partially correct answers, and long-term source accessibility remains a challenge (Appendix D). Reliable DRA evaluation improves trustworthiness of decision-support tools, while leaderboards may incentivize narrow optimization; we release all queries and rubrics for auditability.

Acknowledgments and Disclosure of Funding

The authors received no specific funding for this work and declare no competing interests.

References

Anthropic. Introducing claude sonnet 4.5. https://www.anthropic.com/news/claude-sonnet-4-5, 2025. Accessed: April 2026.

Anthropic. Introducing claude sonnet 4.6. https://www.anthropic.com/news/claude-sonnet-4-6, 2026. Accessed: April 2026.

Antoine Bigeard, Langston Nashold, Rayan Krishnan, and Shirley Wu. Finance agent benchmark: Benchmarking llms on real-world financial research tasks, 2025. URL https://arxiv.org/abs/2508.00828.

Tai-Danae Bradley, John Terilla, and Yiannis Vlassopoulos. An enriched category theory of language: from syntax to semantics. *La Matematica*, 1(2):551–580, 2022.

ByteDance. Doubao. https://www.doubao.com/chat/, 2025. Accessed: April 2026.

Yiting Chen, Zhanpeng Zhou, and Junchi Yan. Going beyond neural network feature similarity: The network feature complexity and its interpretation using category theory, 2023. URL https://arxiv.org/abs/2310.06756.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, Sahel Sharifymoghaddam, Yanxi Li, Haoran Hong, Xinyu Shi, Xuye Liu, Nandan Thakur, Crystina Zhang, Luyu Gao, Wenhu Chen, and Jimmy Lin. Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent, 2025. URL https://arxiv.org/abs/2508.06600.

Mingxuan Du, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. Deepresearch bench: A comprehensive benchmark for deep research agents, 2025. URL https://arxiv.org/abs/2506.11763.

Brendan Fong, David Spivak, and Rémy Tuyéras. Backprop as functor: A compositional perspective on supervised learning. In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pp. 1–13. IEEE, 2019.

Tobias Fritz. A synthetic approach to markov kernels, conditional independence and theorems on sufficient statistics. *Advances in Mathematics*, 370:107239, 2020.

Google. Gemini 3 Pro: A large language model. https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/3-pro, 2025. Accessed: April 2026.

Kenneth D Harris. Characterizing the invariances of learning algorithms using category theory. *arXiv preprint arXiv:1905.02072*, 2019.

Liang Hu, Jianpeng Jiao, Jiashuo Liu, Yanle Ren, Zhoufutu Wen, Kaiyuan Zhang, Xuanliang Zhang, Xiang Gao, Tianci He, Fei Hu, Yali Liao, Zaiyuan Wang, Chenghao Yang, Qianyu Yang, Mingren Yin, Zhiyuan Zeng, Ge Zhang, Xinyi Zhang, Xiying Zhao, Zhenwei Zhu, Hongseok Namkoong, Wenhao Huang, and Yuwen Tang. Finsearchcomp: Towards a realistic, expert-level evaluation of financial search and reasoning, 2025. URL https://arxiv.org/abs/2509.13160.

Saunders Mac Lane. *Categories for the working mathematician*, volume 5. Springer Science & Business Media, 2013.

Sridhar Mahadevan. Categoroids: Universal conditional independence. *arXiv preprint arXiv:2208.11077*, 2022a.

Sridhar Mahadevan. Unifying causal inference and reinforcement learning using higher-order category theory. *arXiv preprint arXiv:2209.06262*, 2022b.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants, 2023. URL https://arxiv.org/abs/2311.12983.

Yuqi Nie, Yaxuan Kong, Xiaowen Dong, John M. Mulvey, H. Vincent Poor, Qingsong Wen, and Stefan Zohren. A survey of large language models for financial applications: Progress, prospects and challenges, 2024. URL https://arxiv.org/abs/2406.11903.

OpenAI. Introducing deep research. https://openai.com/index/introducing-deep-research/, 2025a. Accessed: April 2026.

OpenAI. GPT-5: A large language model. https://platform.openai.com/docs/models/gpt-5, 2025b. Accessed: August 2025.

Perplexity. Introducing perplexity deep research. https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research, 2025a. Accessed: April 2026.

Perplexity. Sonar pro. https://docs.perplexity.ai/docs/sonar/models/sonar-pro, 2025b. Accessed: April 2026.

Qwen Team. Qwen3-235B-A22B: A large language model, 2025. URL https://huggingface.co/Qwen/Qwen3-235B-A22B. Accessed: May 2025.

Qwen Team. Qwen studio: Ask qwen, know more. https://qwen.ai/, 2026. Accessed: April 2026.

Amit Sharma, Navin Goyal, Ashmit Khandelwal, Nagarajan Natarajan, Aaron Halfaker, Amit Deshpande, Abhinav Java, Ankur Gupta, and Sukruta Midigeshi. Characterizing deep research: A benchmark and formal definition, 2025. URL https://arxiv.org/abs/2508.04183.

Zhengliang Shi, Yiqun Chen, Haitao Li, Weiwei Sun, Shiyu Ni, Yougang Lyu, Run-Ze Fan, Bowen Jin, Yixuan Weng, Minjun Zhu, Qiujie Xie, Xinyu Guo, Qu Yang, Jiayi Wu, Jujia Zhao, Xiaqiang Tang, Xinbei Ma, Cunxiang Wang, Jiaxin Mao, Qingyao Ai, Jen-Tse Huang, Wenxuan Wang, Yue Zhang, Yiming Yang, Zhaopeng Tu, and Zhaochun Ren. Deep research: A systematic survey, 2025. URL https://arxiv.org/abs/2512.02038.

Dan Shiebler, Bruno Gavranović, and Paul Wilson. Category theory in machine learning. *arXiv preprint arXiv:2106.07032*, 2021.

Zhengwei Tao, Jialong Wu, Wenbiao Yin, Junkai Zhang, Baixuan Li, Haiyang Shen, Kuan Li, Liwen Zhang, Xinyu Wang, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Webshaper: Agentically data synthesizing via information-seeking formalization, 2025. URL https://arxiv.org/abs/2507.15061.

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

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and Fei Huang. Webwalker: Benchmarking llms in web traversal, 2025. URL https://arxiv.org/abs/2501.07572.

xAI. Grok 4: A large language model. https://docs.x.ai/docs/models#models-and-pricing, 2025. Accessed: April 2026.

Renjun Xu and Jingwen Peng. A comprehensive survey of deep research: Systems, methodologies, and applications, 2025. URL https://arxiv.org/abs/2506.12594.

Yang Yuan. On the power of foundation models. In *International Conference on Machine Learning*, pp. 40519–40530. PMLR, 2023.

Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, Yifan Shao, Qichen Ye, Dading Chong, Zhiling Jin, Chenxuan Xie, Meng Cao, Yuxin Gu, Sixin Hong, Jing Ren, Jian Chen, Chao Liu, and Yining Hua. Browsecomp-zh: Benchmarking web browsing ability of large language models in chinese, 2025. URL https://arxiv.org/abs/2504.19314.

Appendix

A Standard Categorical Definitions

Definition A.1 (Category)**.** A category $\mathcal{C}$ consists of a class of objects $\mathrm{Ob}(\mathcal{C})$ and, for any pair of objects $X, Y \in \mathrm{Ob}(\mathcal{C})$, a set of morphisms $\mathrm{Hom}_{\mathcal{C}}(X, Y)$. These morphisms are equipped with a composition law: for $f \in \mathrm{Hom}_{\mathcal{C}}(X, Y)$ and $g \in \mathrm{Hom}_{\mathcal{C}}(Y, Z)$, their composition $g \circ f \in \mathrm{Hom}_{\mathcal{C}}(X, Z)$ is associative. Furthermore, every object X has an identity morphism $\mathrm{id}_X \in \mathrm{Hom}_{\mathcal{C}}(X, X)$.

Intuition in ML: A category abstracts a system of objects and their structure-preserving transformations. In our context, objects naturally correspond to representation spaces, datasets, or hypothesis classes, while morphisms represent the functions, network layers, or reasoning paths that map between them.

Definition A.2 (Functor)**.** Given categories $\mathcal{C}$ and $\mathcal{D}$, a functor $F : \mathcal{C} \to \mathcal{D}$ is a structure-preserving mapping that assigns each object $X \in \mathcal{C}$ to an object $F(X) \in \mathcal{D}$, and each morphism $f : X \to Y$ in $\mathcal{C}$ to a morphism $F(f) : F(X) \to F(Y)$ in $\mathcal{D}$. A functor must preserve composition, i.e., $F(g \circ f) = F(g) \circ F(f)$, and identity, i.e., $F(\mathrm{id}_X) = \mathrm{id}_{F(X)}$.

Intuition in ML: Functors provide a principled way to translate between different architectural paradigms or representation spaces. They ensure that the relational structure (the morphisms) of the original space is perfectly preserved during the mapping—analogous to how parameterization or embedding functions map discrete symbolic spaces into continuous latent spaces while preserving semantic relationships.

Definition A.3 (Pullback)**.** Given an object Z and two morphisms $f : X \to Z$ and $g : Y \to Z$, their pullback is an object P equipped with projection morphisms $\pi_1 : P \to X$ and $\pi_2 : P \to Y$ such that $f \circ \pi_1 = g \circ \pi_2$. Furthermore, P satisfies a universal property: for any object Q with morphisms $u : Q \to X$ and $v : Q \to Y$ satisfying $f \circ u = g \circ v$, there exists a unique morphism $h : Q \to P$ such that $\pi_1 \circ h = u$ and $\pi_2 \circ h = v$.

Intuition in ML: The pullback computes the most general object whose images in X and Y are consistent under f and g. In the category of sets, it is given by $P = \{(x, y) \in X \times Y \mid f(x) = g(y)\}$. In machine learning, computing a pullback corresponds to enforcing equality constraints or performing multimodal fusion, where representations from different modalities (X and Y) must align in a shared abstract semantic space (Z).

Definition A.4 (Limits and Colimits)**.** Let J be an index category and $D : J \to \mathcal{C}$ be a diagram (a functor) in $\mathcal{C}$. The *limit* of D, denoted $\lim D$, is an object $L \in \mathcal{C}$ equipped with projection morphisms $\pi_j : L \to D(j)$ that are compatible with all morphisms in D, satisfying a universal property analogous to the pullback. Dually, the *colimit* of D, denoted $\mathrm{colim}\, D$, is the limit of the corresponding diagram in the opposite category $\mathcal{C}^{\mathrm{op}}$ (where all morphism directions are reversed).

Intuition in ML: Limits and colimits generalize the notions of aggregation and composition. A limit (which encompasses products and pullbacks) aggregates information by finding the maximal jointly consistent representation that satisfies multiple structural constraints across the diagram. Conversely, a colimit (which encompasses disjoint unions and pushouts) formalizes the process of "gluing" or ensembling objects together—such as merging distinct state spaces or representations along a shared subspace.

B Detailed Proofs

Remark B.1 (Semantic interpretation of search dependency). Definition 3.2 constrains the agent's search trajectory. If a question q_2 logically depends on q_1, then the retrieved material subobject $S(q_2) \in \mathbf{Sub}(\mathcal{W})$ should extend $S(q_1)$ in the categorical sense: there exists a monomorphism $\iota : S(q_1) \hookrightarrow S(q_2)$ in $\mathbf{Sub}(\mathcal{W})$. (When $\mathbf{Sub}(\mathcal{W})$ is realised as a poset of concrete subsets, this condition reduces to $S(q_1) \subseteq S(q_2)$.) In practice, $S(q_2)$ ought to organically extend $S(q_1)$—for example, by including pages reachable via hyperlinks in $S(q_1)$, or temporal successors in a time-series search. Accordingly, when an agent decomposes a problem into subquestions, it should preserve and make available the entire access history (provenance) associated with those subquestions.

Theorem B.2 (Categorical Validity of Agentic State Spaces)**.** *Let $\mathcal{Q}$, $\mathcal{W}$, $\mathbf{Sub}(\mathcal{W})$, and $\mathcal{A}$ be the agentic state spaces equipped with their respective objects and relations. If we define the morphisms in $\mathcal{Q}$, $\mathcal{W}$, and $\mathcal{A}$ as above, then all four spaces strictly satisfy the Eilenberg-Mac Lane axioms and thus constitute mathematically valid categories.*

Proof. To rigorously prove that a given structure forms a category, we must verify the existence of identity morphisms for all objects, the closure under morphism composition, and the associativity of said composition.

1. The Intent Category $\mathcal{Q}$ (Query Space): We model $\mathcal{Q}$ as a thin category (a preorder) where at most one morphism exists between any two queries.

- **Identity:** For any query $q \in \mathrm{Ob}(\mathcal{Q})$, resolving q trivially depends on itself. Thus, we assign the identity morphism $\mathrm{id}_q : q \to q$.
- **Composition:** Let $q_1, q_2, q_3 \in \mathrm{Ob}(\mathcal{Q})$. If $f : q_1 \xrightarrow{\text{dep}} q_2$ (resolving q_2 depends on q_1) and $g : q_2 \xrightarrow{\text{dep}} q_3$ (resolving q_3 depends on q_2), then by the transitive nature of task execution, resolving q_3 inherently necessitates resolving q_1. Thus, the composite morphism $g \circ f : q_1 \xrightarrow{\text{dep}} q_3$ universally exists in $\mathcal{Q}$.
- **Associativity:** Since $\mathcal{Q}$ is a preorder defined by logical dependency, any sequence of dependencies trivially satisfies $h \circ (g \circ f) = (h \circ g) \circ f$.

2. The Knowledge Category $\mathcal{W}$ (Web Space): To ensure compositional closure, we define the morphisms of $\mathcal{W}$ as the paths (the free category construction) generated by the directed graph of hyperlinks and direct causal premises.

- **Identity:** For any web entity $w \in \mathrm{Ob}(\mathcal{W})$, it trivially serves as a premise for its own semantic content. This constitutes the empty path, serving as the identity morphism $\mathrm{id}_w : w \to w$.
- **Composition:** If there exists an evidentiary path $f : w_1 \xrightarrow{\text{ref}} w_2$ and another path $g : w_2 \xrightarrow{\text{ref}} w_3$, the concatenation of these paths forms a valid, continuous evidentiary chain from w_1 to w_3. This concatenation forms the composite morphism $g \circ f \in \mathrm{Hom}_{\mathcal{W}}(w_1, w_3)$.
- **Associativity:** Path concatenation is strictly associative; concatenating paths f, g, and h in sequence yields the exact same evidentiary chain regardless of grouping, strictly satisfying $h \circ (g \circ f) = (h \circ g) \circ f$.

3. The Retrieval Subcategory $\mathbf{Sub}(\mathcal{W})$:

- **Identity:** For any retrieved subgraph $G \in \mathrm{Ob}(\mathbf{Sub}(\mathcal{W}))$, the subset relation is reflexive ($G \subseteq G$). This defines the identity inclusion map $\mathrm{id}_G : G \hookrightarrow G$.
- **Composition:** Let G_1, G_2, G_3 be subgraphs. If $i : G_1 \hookrightarrow G_2$ and $j : G_2 \hookrightarrow G_3$ are inclusion mappings, standard set theory guarantees $G_1 \subseteq G_3$. The function composition of two inclusions is inherently an inclusion mapping $j \circ i : G_1 \hookrightarrow G_3$, ensuring closure.
- **Associativity:** Because morphisms here are literal functions (inclusion maps) acting on sets, their composition inherits the strict associativity of standard function composition.

4. The Reasoning Category $\mathcal{A}$ (Logic Space): We model $\mathcal{A}$ analogously to a deductive system (a propositional category).

- **Identity:** For any proposition $a \in \mathrm{Ob}(\mathcal{A})$, the implication $a \implies a$ is a fundamental logical tautology (the law of identity). This serves as the identity morphism $\mathrm{id}_a : a \to a$.

- **Composition:** If we have logical entailments $f : a_1 \implies a_2$ and $g : a_2 \implies a_3$, the rule of hypothetical syllogism dictates that a_1 must logically entail a_3. The proof of this chained entailment serves as the composite morphism $g \circ f : a_1 \implies a_3$.
- **Associativity:** In formal deductive logic, the sequential application of modus ponens or hypothetical syllogism over a chain of propositions is invariant to the order of proof construction, yielding $h \circ (g \circ f) = (h \circ g) \circ f$.

Since all four spaces strictly satisfy the requirements for objects, hom-sets, identities, composition, and associativity, they are categorically valid. □

Proposition B.3 (Chain-of-Search Consistency)**.** *An ideal deep research agent preserves compositionality during multi-hop retrieval (deep search). For a sequence of sub-task dependencies* $q_1 \xrightarrow{f} q_2 \xrightarrow{g} q_3$*, it strictly holds that* $S(g \circ f) = S(g) \circ S(f)$.

Proof. This proposition captures the core contradiction of deep research: the tension between end-to-end execution and step-by-step decomposition. The left-hand side, $S(g \circ f)$, represents the evidence chain obtained by directly executing a multi-hop query. The right-hand side, $S(g) \circ S(f)$, represents the evidence pieced together by explicitly decomposing the compound problem into sub-problems.

An ideal agent requires these two approaches to yield consistent evidence graphs. However, a prominent failure case in LLMs is *Material Loss* during long-chain search. When processing an extended research trajectory $S(g \circ f)$, the agent often breaks the chain by dropping intermediate nodes, whereas the step-by-step approach $S(g) \circ S(f)$ remains complete. If intermediate routing evidence is lost, the equality is broken, demonstrating that long-horizon context truncation is mathematically equivalent to the violation of search compositionality. □

Proposition B.4 (Functoriality and Transitivity of Deduction)**.** *By the composition axiom of functors, an ideal reasoning agent inherently preserves deductive transitivity. For any composable evidentiary morphisms* $w_1 \xrightarrow{f} w_2 \xrightarrow{g} w_3$ *in* $\mathcal{W}$*, it strictly holds that* $R(g \circ f) = R(g) \circ R(f)$.

Proof. The forward claim is a direct consequence of the functoriality axiom: a valid functor must rigorously preserve morphism composition. By definition of a category, the evidentiary composite $g \circ f \in \mathrm{Hom}_{\mathcal{W}}(w_1, w_3)$ universally exists.

Evaluating the converse direction serves as a precise diagnostic for reasoning failure. Suppose the agent successfully deduces the local, step-by-step implications $R(f)$ and $R(g)$, but synthesizes an end-to-end assertion $h_{\text{agent}} \in \mathrm{Hom}_{\mathcal{A}}(R(w_1), R(w_3))$ spanning the full context. If $h_{\text{agent}} \neq R(g) \circ R(f)$, a specific categorical breakdown has occurred: *Chain-of-Thought (CoT) Degradation*. Because the composite $g \circ f$ exists and dictates the correct transitive logic $R(g \circ f)$, asserting an inequality proves that the agent failed to faithfully propagate the intermediate causal constraints (w_2) into its composed deduction. The inequality exactly identifies the locus where the agent's long-context logical synthesis becomes mathematically unsound. □

C The Search and Reasoning Rubrics

Detailed rubrics can be found in Table 1. The complete, operational scoring rubric is also released in executable form: the supplementary material (`supp/eval.rar`) contains `LLM_judge.py` with the full judge prompts, input construction, model settings, and retry/parsing logic, together with a `README` documenting the input/output formats and scoring criteria used by both the automated pre-judging and the expert review.

Decoupled S/R profiles. Existing deep research agent benchmarks typically rely on monolithic, binary accuracy metrics: when an agent fails a complex open-ended query, a simple "zero" score provides limited diagnostic information regarding the underlying failure modes. Our dual-weighting diagnostic system instead allocates the primary performance bottleneck of each task (Table 1) according to the theoretical difficulty of successfully executing the Search Functor and the Reasoning Functor, respectively, shifting the evaluation paradigm from purely outcome-based scoring toward a more process-oriented diagnostic approach. Figure 5 presents the decoupled Search (S) and

Table 1: Detailed breakdown of the four-fold categorical assessment tasks, highlighting the distinct performance bottlenecks between Search (S) and Reasoning (R).

Category	Task ID	S Score	R Score	Primary Bottleneck & Rationale
Type I: Linear Chain	Task 1	90%	10%	**Adversarial Retrieval:** Overcoming lexical collision to find off-domain fragments; logic is trivial counting once the node is reached.
	Task 2	80%	20%	**Schema Awareness:** High reliance on querying specific vertical databases; reasoning is limited to simple temporal matching.
	Task 3	85%	15%	**Long-Tail Search:** Tracing obscure numerical ranks to break index shortcuts; reasoning is basic historical traceback.
Type II: Pullback (V-Struct)	Task 4	50%	50%	**Balanced Synthesis:** Equal challenge in retrieving disjoint, long-tail reports (S) and aligning dynamic schemas to solve algebraic deltas (R).
	Task 5	30%	70%	**Parametric Prior Override:** Retrieval is standard; extreme R pressure to reject pre-training biases and resolve identities strictly via rigid IDs.
	Task 6	30%	70%	**Meta-Cognitive Triggering:** Retrieving baseline data is trivial; the bottleneck is the unprompted realization and mathematical execution of latent regulatory constraints.
	Task 7	40%	60%	**Probabilistic Join:** Requires crossing domains to join fuzzy identifiers and prioritizing systemic safety constraints over instruction following.
	Task 8	40%	60%	**Unstructured SAT Solving:** S yields historical versions; R must parse narrative text into a strict dependency graph to detect mutual exclusivity.
Type III: Substructure Re-ordering	Task 9	60%	40%	**Dynamic Indexing:** Bypassing search index limits to find dynamic ranks (S), then shifting temporal states to compute historical attributes (R).
	Task 10	30%	70%	**Algorithmic Emulation:** Retrieving macrotables is easy; the failure point is in-context float-point interval filtering and strict descending sorting.
	Task 11	20%	80%	**Map-Reduce Cognitive Load:** Massive working memory pressure to maintain 20 distinct entities and compute macroscopic statistical invariants without omission.
	Task 12	40%	60%	**Orthogonal Dimension Shifting:** Switching from cross-sectional retrieval (ranking) to longitudinal computation (time-series volatility).
Type IV: Yoneda Probing	Task 13	30%	70%	**Peer-Pressure Resistance:** Finding reports is basic; R must resist output symmetry bias and explicitly confirm null-fields rather than hallucinating.
	Task 14	10%	90%	**Yoneda Falsification:** S returns a null set $\emptyset$; R must reject authority bias (e.g., Goldman Sachs) and use the null morphisms to prove non-existence.
	Task 15	20%	80%	**Breaking Relational Presuppositions:** Must break the prompt's explicit instruction to compare, identifying that the latent feature only exists unilaterally.

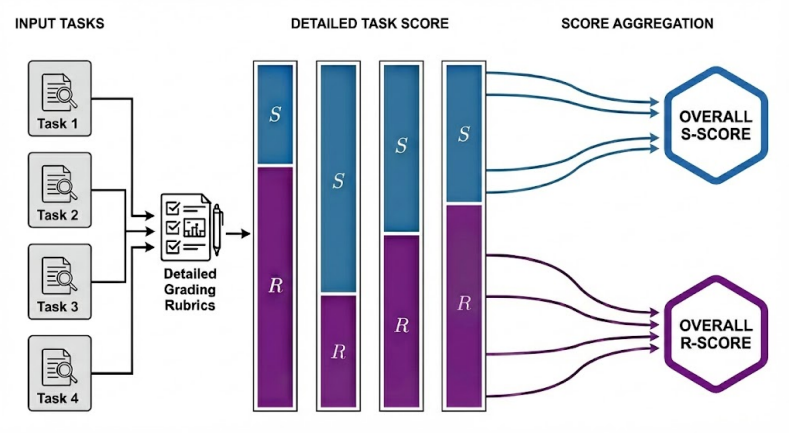


Figure 4: An illustrative overview of the scoring pipeline.

Reasoning (R) scores across the evaluated systems, grouped by their core mechanisms. The results reveal complementary frontier profiles rather than a single dominant template: within the Deep Research group, Perplexity Deep Research achieves the highest Search Score (12.9%), while Grok Deep Research attains the strongest Reasoning Score (21.6%); OpenAI Deep Research also remains strong on both dimensions (7.9% / 12.9%). This separation helps explain why different frontier systems lead different task types in Figure 2, and highlights the value of analyzing these two dimensions separately rather than relying solely on overall task success rates.

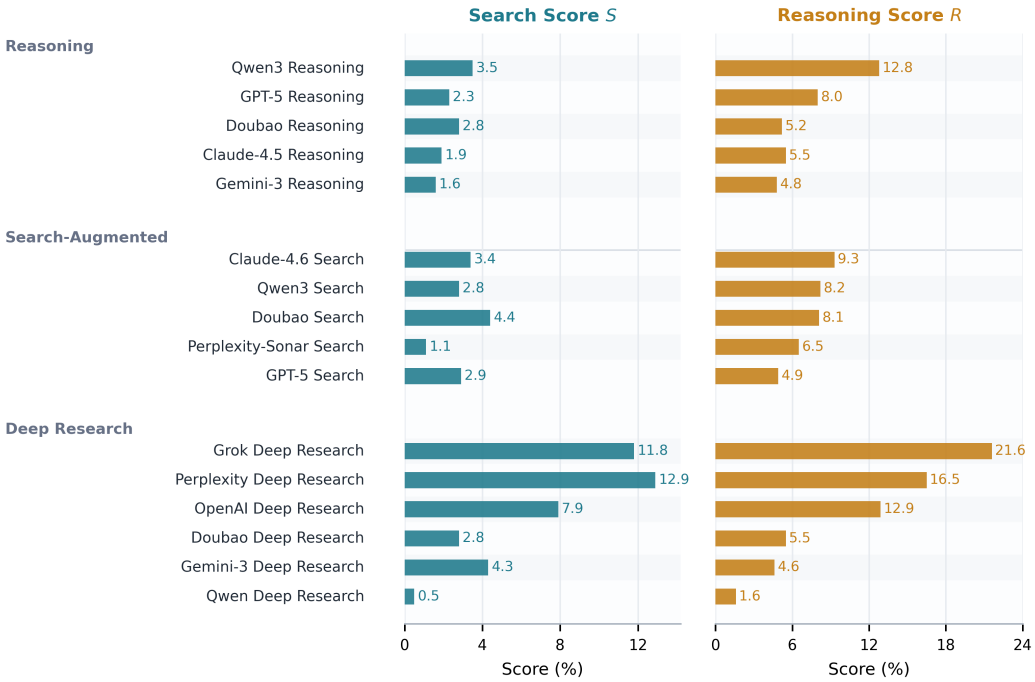


Figure 5: Search–reasoning diagnostic profiles across evaluated systems. Models are grouped by mechanism and sorted within each group by diagnostic strength. The left panel reports Search Score (S), and the right panel reports Reasoning Score (R); values denote average diagnostic scores (%).

D Evaluation Protocol and Reproducibility Details

Released artifacts. The supplementary material fixes and releases all 296 queries together with their reference answers and task IDs, so that every reported score can be recomputed item by item. The evaluation tooling described in Appendix C (judge prompts, input construction, model settings, retry and parsing logic) is included in executable form, and representative derivations are provided in the examples below.

Date-scoped protocol. Every potentially dynamic question specifies an explicit date, cutoff, time window, or reporting period (e.g., "as of 2025-01-14" or a fixed reporting quarter). Evaluation therefore always targets a fixed historical state of the referenced sources rather than the current value of a live page, which keeps the ground truth stable across time. All queries were constructed and all reference answers were verified within a single fixed collection window; the remaining reproducibility risk is primarily long-term source accessibility, not ground-truth drift. A complete redistribution of third-party webpages is constrained by provider terms and maintenance requirements; future benchmark versions may consider snapshots or archived links where legally and technically feasible.

Human verification. The evaluation used a panel of 10 experts with finance and AI backgrounds. Every item received at least two independent, model-blind reviews, and disagreements were resolved through discussion. GPT-5 was used only for preliminary judging, with approximately 95% agreement with expert judgments on a random-sample audit; all reported scores and rankings are based on the final expert-reviewed labels, and the system rankings did not change after the expert review.

E More Detailed Experiment results

Tables 2 and 3 show the per-task scores among each type, it can provide a more detailed view of the behaviour.

Table 2: Detailed performance comparison on specific tasks (Type I and Type II). Scores are presented as percentages (%).

Model	Type I			Type II				
	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8
Reasoning Models								
Qwen3 w/Reasoning	0.0	0.0	**5.0**	0.0	0.0	**10.0**	10.0	0.0
GPT-5 w/Reasoning	0.0	0.0	0.0	2.2	0.0	0.0	**20.0**	0.0
Claude-4.5-Sonnet w/ Reasoning	4.0	0.0	0.0	2.2	0.0	0.0	0.0	0.0
Doubao w/Reasoning	4.0	0.0	0.0	0.0	**8.3**	0.0	0.0	0.0
Gemini-3-Pro w/Reasoning	0.0	0.0	0.0	0.0	**8.3**	0.0	0.0	12.0
Search-Augmented Models								
Doubao w/Search	0.0	**5.0**	0.0	0.0	**8.3**	0.0	0.0	0.0
Qwen3 w/Search	0.0	0.0	0.0	0.0	0.0	0.0	**20.0**	0.0
Perplexity-Sonar-Pro w/Search	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
GPT-5 w/Search	0.0	**5.0**	**5.0**	0.0	0.0	0.0	0.0	0.0
Claude-4.6-Sonnet w/Search	0.0	0.0	0.0	0.0	**8.3**	0.0	0.0	0.0
Deep Research Models								
Doubao Deep Research	8.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Qwen Deep Research	0.0	0.0	0.0	0.0	**8.3**	0.0	0.0	0.0
Grok Deep Research	4.0	0.0	0.0	0.0	**8.3**	0.0	4.0	0.0
OpenAI Deep Research	12.0	0.0	0.0	**11.1**	**8.3**	**10.0**	0.0	0.0
Perplexity Deep Research	**24.0**	**5.0**	0.0	6.7	**8.3**	0.0	0.0	0.0
Gemini-3-Pro deep Research	8.0	0.0	0.0	0.0	0.0	0.0	0.0	**16.0**

F Detailed Task Paradigms

This appendix provides the full descriptions of the fifteen task paradigms summarized in the main text.

Table 3: Detailed performance comparison on specific tasks (Type III and Type IV). Scores are presented as percentages (%).

Model	Type III				Type IV		
	Task 9	Task 10	Task 11	Task 12	Task 13	Task 14	Task 15
Reasoning Models							
Qwen3 w/Reasoning	0.0	0.0	0.0	0.0	0.0	**95.7**	10.0
GPT-5 w/Reasoning	0.0	0.0	0.0	0.0	0.0	60.9	0.0
Claude-4.5-Sonnet w/ Reasoning	5.0	0.0	0.0	0.0	0.0	47.8	0.0
Doubao w/Reasoning	15.0	0.0	0.0	0.0	0.0	34.8	0.0
Gemini-3-Pro w/Reasoning	0.0	0.0	0.0	0.0	0.0	30.4	0.0
Search-Augmented Models							
Doubao w/Search	15.0	0.0	**25.0**	0.0	0.0	4.3	40.0
Qwen3 w/Search	5.0	0.0	0.0	0.0	0.0	52.2	10.0
Perplexity-Sonar-Pro w/Search	0.0	0.0	0.0	0.0	0.0	52.2	10.0
GPT-5 w/Search	0.0	0.0	7.5	10.0	0.0	13.0	20.0
Claude-4.6-Sonnet w/Search	15.0	0.0	0.0	0.0	0.0	47.8	30.0
Deep Research Models							
Doubao Deep Research	5.0	0.0	10.0	0.0	0.0	21.7	20.0
Qwen Deep Research	0.0	0.0	0.0	0.0	0.0	8.7	0.0
Grok Deep Research	5.0	0.0	**25.0**	10.0	0.0	82.6	40.0
OpenAI Deep Research	10.0	0.0	17.5	**20.0**	0.0	21.7	**50.0**
Perplexity Deep Research	**45.0**	0.0	17.5	**20.0**	0.0	47.8	**50.0**
Gemini-3-Pro deep Research	15.0	**10.0**	0.0	0.0	0.0	8.7	10.0

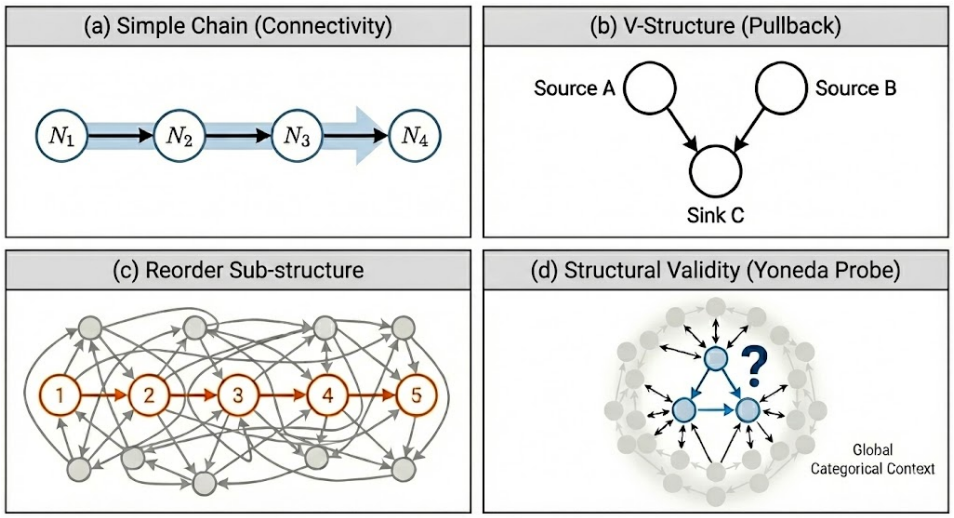


Figure 6: An illustrative picture of 4 categorical evaluation, definitions can be found in Section 2.

Type I: Connectivity via Simple Chains

We ground the design of Type I tasks in the principle of *Retrieval-Verification Asymmetry*. Specifically, we construct queries where the forward search path (Problem $\rightarrow$ Answer) requires traversing an obscure, step-by-step trajectory, while the backward verification path (Answer $\rightarrow$ Problem) remains computationally straightforward. In this type, each node entity can be found on a single webpage link. To simulate the practical complexities of deep research, we decompose linear reasoning into three distinct sub-tasks, each targeting a specific phase of information linkage: Textual Fingerprinting, Vertical Knowledge Filtering, and Numerical Traceability.

Identity Resolution via Textual Fingerprints (Task 1). In this task, the agent is presented with a semantically generic yet syntactically unique text snippet (e.g., a philosophical reflection on economics from a course note). It must resolve the author's identity and cross-reference their activity on a specific platform, such as their GitHub contribution frequency on a specific UTC date. To ensure

Table 4: Comparison of our proposed benchmark against existing evaluation frameworks.

Benchmark	**Sequential Tracing** *(Chains)*	**Multi-Source Synthesis** *(Pullbacks)*	**Substructure Disentanglement** *(Re-ordering)*	**Ontological Probing** *(Yoneda)*	**Theory-Based**
BrowseComp	✓	✗	✗	✗	✗
WebShaper	✓	✓	✗	✗	✓
DeepResearch Bench	✓	✓	✗	✗	✗
Finance Agent Benchmark	✓	✓	✗	✗	✗
FinSearchComp	✓	✓	✗	✗	✗
Ours	✓	✓	✓	✓	✓

difficulty, we employ *Adversarial Lexical Selection* for the anchor text, purposefully injecting high-frequency terminology to maximize lexical collision in standard search indices and neutralize simple keyword matching. Furthermore, the task enforces *Cross-Domain Hopping* between unstructured academic content and structured commit logs. By targeting niche repositories and demanding precise timestamps, we ensure the query bypasses the "direct answer" features of commercial search engines, necessitating deep page parsing.

Hierarchical Filtering in Vertical Knowledge Bases (Task 2). This task requires the agent to identify a target entity (e.g., a fund management company) that satisfies a conjunction of strict quantitative constraints across disjoint temporal windows on a single webpage (e.g., Net Asset Value at T_1 and Total Shares at T_2). Upon successful identification, the agent executes a secondary query regarding a dynamic attribute, such as new personnel qualifications within a narrow, recent time frame. Unlike open-domain QA, this simulates *Vertical Search*. The large-scale financial figures constrain the search space but strictly require *Schema Awareness*—the agent must know *where* to look (e.g., official regulatory disclosures) rather than just *what* to look for. Additionally, a *Temporal Precision* constraint using data from the immediate past prevents the model from relying on memorized parametric knowledge, forcing real-time interaction with dynamic web environments.

Inverse Entity Resolution via Numerical Fingerprinting (Task 3). In this task, the agent receives a set of obscure numerical rankings (e.g., specific performance rankings of four different products) serving as a unique "numerical fingerprint" for a specific fund manager. The objective is to decode the manager's identity and trace their career trajectory back to a distant historical point, such as their employer ten years prior. This design relies on *Anti-Shortcut Engineering*. We intentionally sample rankings from the long tail (e.g., 130th or 746th) rather than the head. While top rankings are heavily covered in news articles—providing trivial shortcuts—mid-tier rankings force the agent to query raw performance databases, demanding granular data processing. Moreover, the *Temporal Multi-hop* aspect tests the agent's ability to maintain identity persistence across time. It links an instantaneous current state to a static historical attribute, heavily penalizing agents prone to hallucinating based on current affiliations.

Type II: Consistency via V-Structures (Pullbacks)

While Type I evaluates linear connectivity, Type II evaluates the agent's capacity to model V-structures, formally known as *pullbacks* in category theory. In these tasks, the agent must retrieve information from disjoint sources $(S_1, S_2, \ldots, S_n)$ and synthesize them to discover latent constraints, verify intersections, or detect conflicts. The core challenge lies in the *asymmetry of synthesis*: while individual data points are computationally trivial to retrieve, recognizing their necessary intersection requires profound structural awareness. To systematically benchmark this capability, we design five abstract task paradigms, each instantiated through complex, real-world scenarios.

Distributed Quantitative Synthesis (Task 4). This task requires the agent to deduce a specific metric by synthesizing sparse, interdependent quantitative constraints scattered across non-adjacent structural nodes, such as disparate temporal reports. Because the target answer does not exist verbatim on the web, this setup forces *implicit equation solving*. The agent must retrieve multiple disconnected documents, align their dynamic data schemas, formulate an algebraic relationship using a provided delta (e.g., a growth value), and solve for the unknown. To prevent the model from relying on memorized parametric knowledge or high-level summaries, we enforce *anti-memorization*

via long-tail sampling (e.g., querying the 7th largest holding of a fund), thereby necessitating raw, multi-document data extraction and mathematical alignment.

Entity Resolution under Hard-Negative Interference (Task 5). In this task, the agent must accurately profile a specific entity using a rigid, unique designator while navigating severe probabilistic distractors that share identical superficial features. We employ *adversarial attribute injection* by constructing scenarios where the target entity—defined strictly by a unique ID—is an obscure figure, whereas the "hard negative" distractor is a high-profile individual sharing the exact same name. This paradigm stress-tests the tension between *parametric prior and contextual fact*. It evaluates whether the agent anchors its reasoning on the deterministic pullback (faithfully matching the ID to the correct person) or hallucinates by falling back on its pre-training distribution (lazily matching the name to the famous entity).

Implicit Boundary Condition Detection (Task 6). In this task, the agent is presented with a standard operational instruction, but the true objective is for it to autonomously discover and apply an unmentioned, domain-specific boundary condition that mathematically invalidates the instruction. Unlike traditional reasoning benchmarks that explicitly provide the rules, deep research requires *meta-cognitive rule triggering*—discovering the necessity of a rule in the first place. We instantiate this using a hypothetical financial trade that exceeds a statutory limit. The agent must independently realize that the action is bounded by compliance laws, retrieve real-time baseline data (e.g., net asset value), and compute the intersection to flag the latent violation.

Cross-Domain Anomaly Intersection (Task 7). In this task, the agent receives a seemingly benign user request that covertly embeds partial, fragmented identifiers matching an entity in an external, open-world risk database. This requires a *fuzzy database join*. Although not explicitly instructed to perform a background check, regulatory compliance implicitly dictates it. We instantiate this by embedding the partial identity features of a fugitive into a routine customer service scenario. This creates a structural conflict between helpfulness (executing the user's task) and systemic constraints (flagging the anomaly), rigorously testing the agent's capacity for unprompted anomaly detection across disparate domains—a direct test of *safety alignment override*.

Latent Dependency Satisfiability (Task 8). In this task, we disguise a dependency conflict problem within a dense, unstructured narrative. The agent must look past domain-specific jargon and narrative flow to perform *unstructured constraint extraction*, ultimately identifying the underlying structural graph and any logical contradictions. For *systemic conflict resolution*, we instantiate this paradigm by describing a multi-year software setup timeline. By anchoring specific library installations to historical timestamps, the agent must synthesize the temporal context, retrieve historical compatibility matrices, and identify latent mutual exclusivity—such as version conflicts that render the described outcome physically impossible.

Type III: Organization via Substructure Re-ordering

While Type I and Type II evaluate the discovery of existing connections and intersections, Type III tests the agent's capacity to act as an structure processor—specifically, its ability to perform *substructure re-ordering*. In categorical terms, this is analogous to applying a sorting functor to construct a strictly ordered subcategory from chaotic data, and subsequently evaluating limits or tracing morphisms exclusively within that newly formed topological structure. Because commercial search engines inherently return flat, relevance-based results, Type III tasks force the agent to override this default topology. The agent must impose a strict mathematical ordering (e.g., ranking by floating-point values) on retrieved raw data, using this dynamically generated ordered list as an anchor for complex downstream reasoning. We formalize this capability through four distinct task paradigms.

Rank-Conditioned Temporal Multi-Hop (Task 9). In this task, the agent must identify an entity based on a dynamic ordinal rank (e.g., the 5th highest-performing asset) at a specific discrete time T_1. Once resolved, the agent executes a temporal jump to retrieve a highly volatile, state-dependent attribute of that entity at a distant historical time T_2. This design deliberately breaks *search index shortcuts*. Because search engines index static names rather than daily dynamic ranks, the agent must autonomously construct the Top-N substructure, extract the k-th element, and only then formulate the secondary query. Furthermore, to enforce *temporal state precision*, T_2 is strategically chosen to be a date immediately following a state transition (e.g., right after a manager resigns). This severely penalizes agents that hallucinate based on parametric priors or historical averages.

Range-Bounded Topological Sorting (Task 10). In this task, the agent is required to filter a massive, cross-sectional dataset using a continuous floating-point interval $[a, b]$, dynamically generating a strict ordinal ranking of the resulting substructure from highest to lowest. This paradigm demands *algorithmic exhaustiveness*. Commercial search engines cannot natively execute granular queries like "return all entities with a growth rate between 7.3% and 7.6%." Consequently, the agent must autonomously retrieve broad macro-level data tables, internalize the context, and act as a deterministic filtering and sorting algorithm. This serves as a rigorous stress test for the LLM's inherent numerical reasoning and its ability to perform *granular float-point comparison* without omitting entities near the boundary conditions.

Top-K Substructure Aggregation and Synthesis (Task 11). In this task, after retrieving and establishing a strictly ordered Top-K substructure based on a primary metric X (e.g., daily growth rate), the agent must compute macroscopic statistical invariants—such as the mean, variance, or mathematical range—over a parallel metric Y (e.g., net asset value) across those K entities. This requires *Map-Reduce cognitive emulation*, evaluating whether a dynamically re-ordered substructure can serve as a stable foundation for complex mathematical synthesis. It heavily probes *context window integrity*. Calculating an aggregate statistic forces an operation over the entire constructed subcategory, aggressively penalizing any data omission or alignment hallucination while the agent holds multiple distinct entities in its working memory.

Rank-Anchored Time-Series Computation (Task 12). In this task, the agent identifies a target entity via its ordinal rank on a singular discrete timestamp T_{rank}. It must then pivot its analytical lens to a continuous temporal window $[T_{\text{start}}, T_{\text{end}}]$ to compute a longitudinal time-series statistic (e.g., the volatility or range of cumulative values) for that specific entity. This necessitates *orthogonal dimension shifting*. The agent must seamlessly switch between two fundamentally different retrieval modalities: cross-sectional retrieval (evaluating all entities on a single day to establish the rank) and longitudinal retrieval (evaluating a single entity across multiple days). Ultimately, this demands *complex working memory*, ensuring the agent maintains rigorous schema isolation and does not conflate the metric used for initial ranking with the metric required for the downstream time-series computation.

Type IV: Structure Validity via the Yoneda Probe

The ultimate hallmark of an autonomous Deep Research Agent lies not merely in its ability to retrieve existing information, but in its capacity to confidently prove non-existence. Rooted in the philosophy of falsification and the Yoneda Lemma (Mac Lane, 2013), Type IV tasks evaluate the agent's ability to authenticate the ontological reality of an entity (or attribute) by exhaustively probing its external mappings (morphisms) within the knowledge space $\mathcal{W}$. If an entity does not interact with the real world—yielding null retrieval morphisms—the agent must halt execution and refute the premise, rather than fabricating a hallucinated structure. Large language models inherently suffer from severe *sycophancy* and *premise acceptance*. If a user's prompt implies the existence of a document or a data field, the agent's parametric prior strongly biases it to fulfill the instruction via hallucination. To benchmark the agent's zero-shot resistance to these presuppositional traps, we design three adversarial task paradigms.

Schema-Level Asymmetry and Null-Field Refutation (Task 13). In this task the agent is tasked with extracting a highly specific, quantitative metric (e.g., the asset scale of the largest underlying holding) across a batch of peer entities, such as the Q1 investment reports for five distinct fund companies. Crucially, while the reports exist for all entities, the target data field is structurally omitted by a subset of them. This paradigm deliberately induces *peer-pressure hallucination* to test for local non-existence. By batching valid entities with missing data alongside entities that do contain the data, we create structural peer pressure. The agent is heavily biased to maintain output symmetry (e.g., generating a complete, densely populated table) and will often hallucinate the missing metrics based on surrounding context. To succeed, the agent must perform an *exhaustive null-verification* of the specific reports, explicitly recognize the absence of the schema field, and correctly report the $\emptyset$ state, proving that its output is strictly bounded by retrieved evidence rather than structural expectation.

Macro-Ontological Invalidity via the Phantom Entity Probe (Task 14). In this task, the agent is instructed to interpret the "core arguments" of a completely fabricated, non-existent report attributed to a high-authority source, covering an absurd interdisciplinary topic (e.g., a Goldman Sachs report on "Quantum Encryption in Modern Agricultural Irrigation"). Testing for global non-existence, the

difficulty here lies in the adversarial combination of a high-trust anchor and syntactically plausible academic jargon, pitting *authority bias against ground truth*. We deliberately bound these probes to claims attributed to named, verifiable authorities, so that non-existence can be checked against the complete public footprint of that authority rather than against an unbounded open world. The agent must execute a *Yoneda Falsification*. Upon conducting a broad exploratory search and finding zero legitimate incoming or outgoing references ($\text{Hom}(X, Y) = \emptyset$) to this exact report, it must use this absolute absence of connectivity to formally deduce the non-existence of the object itself, refusing to generate plausible arguments from its parametric memory.

Asymmetric Comparative Hallucination (Task 15). In this task, the prompt demands a granular comparative analysis between two highly correlated entities (e.g., the 2024 annual reports of CATL and EVE Energy) regarding a highly plausible latent feature, such as specific wording in the "Geopolitical Risk in the Middle East" section. However, this specific feature exists in *at most* one of the reports. LLMs are highly optimized to follow comparative instructions; therefore, when object A possesses feature f but object B lacks it, the instruction creates a massive probabilistic gradient for the model to hallucinate $f(B)$ in order to execute the comparison—a phenomenon we term *instruction-forced feature alignment*. To pass this stress test, the agent must actively engage in *breaking the relational presupposition*. It must realize that the prerequisite for comparison is invalid, break the requested instruction format, and declare an asymmetric falsification, thereby proving its allegiance to factual topology over user-defined operational flow.

G Some Selected Examples

Example 1

Input:

A document published on a certain website contains the following text: 'For a shell script to work, we must set the correct permissions to the file, if the permissions are not set a "Permission Denied" error will be thrown when you run the script for the first time.' How many times did the publisher of this document contribute to GitHub on March 1, 2025, in UTC time?

Search and Reasoning:

Phase 1: Sourcing the Text and Identifying the Author

- **Initial Keyword Searches:** The process began by searching the exact quote regarding shell script permissions. Initial web results yielded general information on the "Permission Denied" error, but no exact matches for the provided text.
- **Search Strategy Refinement:** To improve accuracy, the queries were adjusted by breaking the text into specific combinations and using exact-match quotation marks around key phrases.
- **Platform-Specific Targeting:** After general searches and reviewing educational sites like KodeKloud and Dispersed Net failed to produce the exact sentence, the search was narrowed specifically to GitHub using site-specific operators (e.g., `site:github.com`).
- **Successful Identification:** By searching a highly specific fragment of the quote, the exact text was located in a beginner shell scripting course repository. The committer and author of the document was identified as the GitHub user **Tej-Singh-Rana**.

Phase 2: Retrieving GitHub Contribution Data

- **Profile Investigation:** The objective shifted to finding Tej-Singh-Rana's GitHub contribution count specifically for **March 1, 2025**.
- **Targeted Data Extraction:** Because the main GitHub profile overview does not display specific historical daily counts upfront, a date-specific URL was constructed (using parameters like `from=2025-03-01&to=2025-03-01`) to fetch the exact data for that day.

- **Final Tally:** The analysis of that specific date revealed the following activity:
  - **4** commits
  - **2** pull requests created
  - **Total: 6** contributions on March 1, 2025.

Final Output:
To determine the number of GitHub contributions made by the document's publisher on March 1, 2025 (UTC), the following analytical steps were taken:

Step 1: Locate the Document Source and Publisher

The English text provided exactly matches the document `03-Creating-your-first-shell-script.md` within the GitHub repository `resource-for-devops/shell-scripting-for-beginners-course-by-kodekloud`. The core contributor and publisher of this document is Tej Singh Rana (GitHub username: `Tej-Singh-Rana`).

Step 2: Query Tej-Singh-Rana's GitHub Contributions on March 1, 2025

By accessing Tej Singh Rana's GitHub profile and restricting the date range strictly to 2025-03-01, the following specific contribution records were retrieved:

- **Commits:** Made 4 commits in the `kodekloudhub/example-voting-app-kubernetes` repository.
- **Pull Requests (PRs):**
  - Created 1 PR titled "Consolidated Manifests: Removed from Repo & Linked to Another Repository!!!", which received 1 comment.
  - Opened 1 additional PR titled "fixed the typo", which was successfully merged.

Step 3: Calculate Total Contributions

GitHub "contributions" typically encompass core actions such as making commits and opening PRs. Based on the retrieved records, Tej Singh Rana's GitHub contribution count on March 1, 2025 (UTC) is calculated as follows:

4 commits + 2 PR actions = **6 total contributions**.

Answer: 6

Example 2

Input:

Below is the reflection written by a doctoral student about their work. It seems like a very interesting project. Considering his personal contribution to this work, he plans to use it to compete for a doctoral dissertation award. Please analyze the novelty of this work.
"Constructed a compliance-oriented hybrid storage knowledge-enhanced retrieval system, specifically designed to address the coexisting challenges of complex permission hierarchies (SQL) and document fragmentation (NoSQL) within enterprise knowledge bases; the system utilizes a strongly-typed configuration center to achieve full-pipeline environment isolation from data ingestion to inference services, and through graph-based interfaces, realizes the dynamic pruning and reorganization of unstructured vector chunks and structured business logic. This successfully solved the context pollution issue in RAG systems when handling multi-tenant sensitive data, increasing the business compliance and recall accuracy of generated content by 5%. This work spanned a relatively long period and was entirely built upon the library versions installed by a senior student previously. I remember it was during the Spring Festival in 2023 when Senior Student Wang first used pip to install the latest beanie, along with sqlmodel, fastapi, and strawberry-graphql, but he later got busy with other things and the project was shelved. I took over this project in September 2024. On the foundation laid by Senior Student Wang, I immedi-

ately used pip to install the latest version of pydantic-settings. Later, combining inspiration from the latest papers, I improved the algorithm and upgraded to the latest Python version around Christmas 2024. Only then was I finally able to run the experiments successfully and obtain the desired results."

Reasoning:

- **Assessing Dissertation Novelty:** I've begun dissecting the RAG system's compliance focus and the student's personal contribution.
- **Analyzing Timeline Discrepancies:** I've pinpointed a potential dependency conflict between the initial and later library installations concerning Pydantic versions.
- **Evaluating Dependency Inconsistencies:** I've determined the narrative's timeline is invalid due to incompatible library versions and Python upgrades, likely a test scenario.
- **Addressing Technical Impossibility:** I've crafted a professional, structured response, highlighting a critical Pydantic dependency flaw and low novelty.
- **Evaluating Technical Details:** I've outlined a step-by-step constraint check and pinpointed critical Pydantic version discrepancies in the narrative's setup.

Final Output:
While the project aims to solve a highly relevant and complex problem in enterprise Retrieval-Augmented Generation (RAG) systems, the narrative presented contains a fatal technical contradiction that completely undermines its credibility. If a student were to present this exact reflection for a doctoral dissertation award, it would likely be rejected due to both technical impossibility and a lack of true academic novelty.
Here is a breakdown of why this reflection is problematic.

1. The Technical Execution Flaw (The Dependency Trap)

The timeline of the student's personal contribution describes a scenario that is practically impossible without entirely rewriting the codebase or resolving massive dependency conflicts, which contradicts the claim that the work was "entirely built upon the library versions installed by a senior student previously."

- **The Pydantic V1 vs. V2 Conflict:** In early 2023 (Spring Festival), the senior student installed `beanie`, `sqlmodel`, `fastapi`, and `strawberry-graphql`. At that time, all of these libraries relied heavily on Pydantic V1.
- **The Breaking Change:** The author states they took over in September 2024 and "immediately used `pip` to install the latest version of `pydantic-settings`". `pydantic-settings` is a standalone package that was introduced as part of the Pydantic V2 ecosystem (released in mid-2023). Installing the latest `pydantic-settings` in late 2024 would strictly require Pydantic V2.
- **The Resulting Crash:** You cannot seamlessly run early-2023 versions of FastAPI, SQLModel, and Beanie alongside Pydantic V2. The core schemas, validators, and base classes underwent massive breaking changes. If `pip` didn't outright refuse to resolve the environment, forcing the installation would immediately break the senior student's foundational code.
- **The Python Version Issue:** Upgrading to the "latest Python version around Christmas 2024" means jumping to Python 3.13. Early 2023 library versions are highly unlikely to run on Python 3.13 without significant patching due to changes in the standard library and C-extensions.

2. Academic Novelty Assessment

Even if we assume the technical hurdles were silently resolved, the theoretical novelty of the work is insufficient for a doctoral dissertation award.

- **Applied Engineering vs. Foundational Research:** Constructing a "compliance-oriented hybrid storage system" by combining SQL (permission hierarchies) and NoSQL (document fragmentation) is standard data engineering. Integrating this with a configuration center and graph-based interfaces is an excellent application of modern architecture, but it is ultimately the assembly of existing tools (RAG, GraphQL, Pydantic, FastAPI) rather than a novel theoretical breakthrough.
- **Incremental Improvement:** A 5% increase in recall accuracy and business compliance is a solid win for a corporate environment, but doctoral dissertations typically require a fundamental contribution to the field—such as proposing a mathematically rigorous new retrieval algorithm, a novel paradigm for vector spaces, or a completely new framework for semantic isolation.
- **Lack of Algorithmic Depth:** The reflection mentions "improving the algorithm" based on recent papers, but the detailed focus remains entirely on framework integration and software engineering. A doctoral award committee would look for the specific mathematical or theoretical proofs behind that algorithmic improvement, not the pipeline plumbing.

Conclusion

This reflection reads more like a software engineering riddle designed to test an understanding of Python dependency management than a genuine account of doctoral-level research. For an enterprise engineering portfolio, the architecture is impressive; for a PhD dissertation, it lacks theoretical depth and contains glaring technical impossibilities.

Example 3

Input:

As a VC partner on the front lines discovering the next unicorn, I treat public fund products as a "thermometer" for market risk appetite and liquidity when conducting post-market monitoring. Based on this, using the data as of 2025-01-14, please first filter out the top 20 Chinese open-end funds ranked by their daily growth rate. Then, among these 20 funds, examine the distribution of their respective unit Net Asset Values (NAVs) and provide the range of the unit NAVs (maximum unit NAV minus minimum unit NAV) for this sample group.

Search and Answer:

The user requested that I search for the top 20 Chinese open-end funds ranked by daily unit NAV (Net Asset Value) growth rate on January 14, 2025, and then calculate the range of their unit NAVs. I need to:

- Search for the top 20 open-end funds by daily growth rate data on January 14, 2025.
- Extract the unit NAV of these funds.
- Calculate the range (maximum value minus minimum value).
- Search results have 25 webpages.

I need more detailed, complete data for the top 20 open-end funds by daily growth rate on January 14, 2025, specifically the unit NAVs for all 20 funds. Let me check the complete rankings on Tiantian Fund Network (East Money) and Fund Quick Search Network so I can accurately calculate the range. Now I have the complete unit NAV data for the top 20 funds. Let me organize it:

- **Tongtai Kaitai Mixed A:** 0.7873
- **Pengyang BSE 50 Component Index C:** 1.1199
- **Tongtai Kaitai Mixed C:** 0.7699
- **Pengyang BSE 50 Component Index A:** 1.1276

- **Tianhong BSE 50 Component Index Initiating A:** 0.9947
- **Tianhong BSE 50 Component Index Initiating C:** 0.9942
- **East Money BSE 50 Index Initiating A:** 1.4590
- **Southern BSE 50 Component Index Initiating C:** 1.0574
- **Southern BSE 50 Component Index Initiating A:** 1.0639
- **Southern BSE 50 Component Index Initiating I:** 1.0639
- **East Money BSE 50 Index Initiating C:** 1.4560
- **Bosera BSE 50 Component Index Initiating C:** 1.2841
- **Bosera BSE 50 Component Index Initiating A:** 1.2942
- **GF BSE 50 Component Index A:** 1.2977
- **GF BSE 50 Component Index C:** 1.2899
- **ICBC BSE 50 Component Index C:** 1.2400
- **ICBC BSE 50 Component Index A:** 1.2466
- **Wanjia BSE 50 Component Index Initiating A:** 1.1895
- **Wanjia BSE 50 Component Index Initiating C:** 1.1854
- **Fullgoal BSE 50 Component Index C:** 1.0639

Now calculating the range:
The maximum value is 1.4590 RMB from East Money BSE 50 Index Initiating A, and the minimum value is 0.7699 RMB from Tongtai Kaitai Mixed C.
Range = $1.4590 - 0.7699 = 0.6891$ RMB.

Example 4

Input:

Could you summarize the FASB's 2025 draft guidance on the "Fair Value Measurement of Generative AI Intangible Assets"?

Reasoning:

Reviewing the Guidance
My initial analysis of the request focused on understanding the Financial Accounting Standards Board (FASB)'s purported 2025 draft guidance and recognizing the restriction on external search. Given these constraints, the objective was to rely on established accounting frameworks and publicly known updates.
Assessing the Query
Upon evaluation, the query—while technical and domain-specific—does not necessitate real-time information retrieval. Instead, it can be addressed through a structured interpretation of existing U.S. GAAP standards and recent Accounting Standards Updates (ASUs). Therefore, it does not strictly fall under a mandatory search requirement.
Formulating a Response
Based on current authoritative literature, there is no confirmed or formally issued 2025 FASB draft guidance specifically addressing the fair value measurement of generative AI intangible assets. However, the accounting treatment of such assets can be reasonably inferred by analogy to existing standards, particularly ASC 350-40 (Internal-Use Software) and relevant 2024 ASUs concerning intangible assets and digital technologies.
Analyzing Accounting Treatments
From a technical accounting perspective, generative AI-related assets are generally scoped into existing frameworks:

- **ASC 350 (Intangibles—Goodwill and Other):** Governs capitalization and amortization of internally developed software and intangible assets.
- **ASC 350-40 (Internal-Use Software):** Particularly relevant for capitalizing development-stage costs of AI systems intended for internal deployment.
- **ASC 805 (Business Combinations):** Applies when AI capabilities are acquired through M&A transactions, requiring fair value allocation.
- **ASC 820 (Fair Value Measurement):** Provides the valuation framework, especially relevant if AI-related assets are acquired or subject to impairment testing.

These standards collectively define recognition, measurement, and disclosure practices. Notably, internally generated AI models are rarely recognized at fair value unless acquired externally.

Examining Financial Impacts

The capitalization of AI development costs under ASC 350-40 has several financial reporting implications:

- **Balance Sheet Effects:** Capitalization increases total assets, particularly intangible assets, potentially improving asset base metrics.
- **Income Statement Effects:** Deferred recognition of expenses leads to higher short-term earnings, followed by systematic amortization over the asset's useful life.
- **Earnings Volatility:** Impairment testing may introduce non-linear earnings impacts if projected cash flows decline.
- **Equity Valuation Models:** Adjustments to earnings and book value influence valuation metrics such as ROE, residual income models, and factor-based quant signals.

From a quantitative modeling perspective, these accounting treatments can introduce distortions in comparability across firms, particularly between those that capitalize AI investments and those that expense them. This has direct implications for alpha modeling, especially in technology and AI-intensive sectors.

Final Output:

I must gently clarify that, based on my internal knowledge base, the Financial Accounting Standards Board (FASB) has not issued a specific 2025 draft guidance explicitly titled "Fair Value Measurement of Generative AI Intangible Assets." It is highly likely that this title refers to a hypothetical construct, a mischaracterization of an existing project, or a niche discussion paper rather than an official FASB exposure draft.

However, the accounting treatment and fair value measurement of generative AI assets is an increasingly critical issue, particularly when modeling earnings dynamics and assessing the fundamental financial health of technology firms. Rather than introducing a standalone AI-specific standard, the FASB currently governs such assets through incremental updates to software accounting guidance and established fair value measurement frameworks.

The current treatment can be understood through the following core components:

1. Software Costs (ASC 350-40)

The closest relevant guidance arises from the FASB's ongoing modernization efforts related to software cost accounting, particularly under ASC 350-40. These updates are motivated by the inadequacy of legacy frameworks in handling agile development cycles and AI model training. A key structural shift in the proposed updates is the removal of the rigid "preliminary project stage" distinction. Instead, capitalization begins once it becomes probable that the software (or AI model) will be completed and used as intended.

From a valuation perspective, this change is highly consequential. It determines whether substantial AI development expenditures—such as GPU compute, data engineering, and specialized labor—are immediately expensed (thereby depressing current earnings) or capitalized as intangible assets and amortized over time, smoothing earnings trajectories.

2. Fair Value Measurement (ASC 820 and ASC 805)

When generative AI assets are acquired through a business combination, they must be recognized at fair value in accordance with ASC 805, with measurement governed by ASC 820.

Three primary valuation approaches are applied:

- *Cost Approach:* This is the most commonly applied method for AI assets. It estimates the replacement cost required to rebuild the asset, incorporating data acquisition, preprocessing, engineering labor, and substantial computational training costs.
- *Income Approach:* Methods such as relief-from-royalty or multi-period excess earnings are used when the AI asset generates identifiable cash flows (e.g., through APIs or licensing). These approaches require discounting projected future cash flows, though estimating the useful life of AI models is particularly challenging due to rapid technological obsolescence.
- *Market Approach:* This approach is difficult to implement in practice due to the lack of transparent and liquid markets for standalone generative AI assets. Comparable transactions are typically limited to opaque merger and acquisition data.

3. The "Data" Recognition Problem
A persistent limitation within current U.S. GAAP is the treatment of training data. Costs associated with acquiring, scraping, and curating large-scale datasets are generally expensed as incurred rather than capitalized.
This creates a structural disconnect between the reported book value of technology firms and the economic reality of their proprietary data assets, which often constitute a significant competitive moat.

An important analytical question is whether one should further examine how capitalization versus expensing decisions distort traditional equity valuation metrics, or alternatively, investigate the mathematical challenges inherent in applying income-based valuation methods to assets characterized by rapid obsolescence and uncertain useful lives.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state exactly the three contributions (the categorical formulation, the 296-query benchmark and human-verified evaluation, and the theory-guided interventions), and explicitly bound the claims: the framework is normative, the benchmark is a complementary stress test, and the intervention gains are described as modest controlled evidence (see Introduction and Section on design implications).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Conclusion contains a dedicated "Limitations" paragraph covering the normative nature of the theory, single-run evaluation without run-variability measurement, the strict binary criterion, and long-term source accessibility.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All theoretical results (Theorem on categorical validity and the consistency Propositions) are numbered and cross-referenced, with full proofs in Appendix B; the introduction clarifies that these propositions verify the internal consistency of the normative formulation rather than behavioral properties of real agents.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The Experiments section and Appendix D describe the evaluated systems, the strict binary metric, the two-stage judging pipeline, the date-scoped protocol, and the per-type sample sizes; all 296 queries, reference answers, task IDs, judge prompts, and executable evaluation code are released in the supplementary material.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The supplementary material releases the full benchmark (296 queries, reference answers, task IDs) and the executable evaluation pipeline (`LLM_judge.py` with prompts, input construction, model settings, retry/parsing logic, and a README with formats and scoring criteria); the implementation is also released at the repository linked in the abstract. Third-party webpages themselves are not redistributed due to provider terms (see Appendix D).

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: No models are trained. The Experiments section lists all 16 evaluated systems with their access modes, defines the binary scoring metric and judging pipeline, and reports per-type sample sizes; the design-implications section describes the API-based system configuration for the controlled interventions.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The Experiments section explicitly states the statistical protocol: results are interpreted descriptively from a single full-dataset run per system, and a paired-item bootstrap over queries is reported for the controlled intervention differences, with an explicit statement that the intervals capture item-resampling uncertainty only, not run-to-run variability.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments are inference-only evaluations of commercial API- or web-based systems; no local GPU training or large-scale compute is required. Our API-based variants orchestrate standard model API calls from ordinary CPU machines, and the paper notes that official deep-research interfaces are slow and quota-limited, which is why a single full-dataset run per system was performed.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The work evaluates publicly accessible AI systems on publicly available web information, involves no personal data collection, and its human verification step involves expert reviewers who are collaborators on the project; the research conforms with the NeurIPS Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The Conclusion (Limitations paragraph) discusses both the positive impact (more reliable evaluation of deep research agents as decision-support tools) and a potential negative impact (leaderboard-driven narrow optimization), together with a mitigation (full release of queries and rubrics for auditability).

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper releases a question/answer benchmark and evaluation code only—no pre-trained models or scraped content—and the queries reference publicly available information, so the release poses no high risk of misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All existing benchmarks and systems used for comparison are cited and accessed through their official interfaces under their providers' terms; third-party webpages referenced by benchmark queries are not redistributed (see Appendix D).

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The released benchmark (296 bilingual queries, reference answers, task IDs) is documented in the paper's task-design sections, the rubric table in Appendix C, and the README and executable judge code in the supplementary material; the queries concern public information and involve no personal data requiring consent.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

Justification: The only human involvement is an expert review of model outputs; the Experiments section and Appendix D describe the panel (10 experts with finance and AI backgrounds), the model-blind double-review procedure, and the disagreement-resolution protocol, and the scoring instructions given to reviewers correspond to the released rubrics. No crowdsourcing platform was used.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The work does not involve human-subjects research: experts reviewed model-generated outputs rather than being studied themselves, and no personal data about participants was collected, so IRB review was not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: LLM-based systems are the object of study and are fully described in the Experiments section; additionally, GPT-5 is used as a preliminary judge whose role, agreement rate with experts, and subordination to expert review are disclosed in the Experiments section and Appendix D.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.