CONTRASTIVE Discovery: Open-Ended Scientific Discovery over Competing Explanations

Ziang Liu [1 *] **James J. Kim** [1 *] **Yijia Dai** [1] **Jennifer J. Sun** [1]

Abstract

LLM agents are increasingly used for open-ended, data-driven scientific discovery, yet most systems remain *pointwise*: they evaluate one candidate hypothesis at a time for surprise, novelty, or evidential support. This is information-inefficient: rejecting a hypothesis reveals that one explanation is unlikely, but not which alternatives are better supported. We argue that the proper epistemic unit of discovery is not a single hypothesis but a scientific question together with the mutually exclusive and collectively exhaustive (MECE) explanations that could answer it. We introduce **CONTRASTIVE Discovery**, which scores candidate discoveries by how evidence redistributes belief across competing explanations. This allows a single experiment to both rule out hypotheses and identify the most plausible alternative. Across six scientific domains, CONTRASTIVE Discovery yields substantially more resolved scientific questions than *pointwise* baselines under the same evidence-generation budget. Among top *pointwise*-surprisal candidates, 89% are rejected; on the identical evidence, representing explicit alternatives instead resolves 73% to a specific competing explanation. These gains come almost entirely from corrective updates, showing that an explicit representation of alternatives turns the negative evidence that *pointwise* discovery discards into resolved scientific questions.

1. Introduction

The shift toward autonomous scientific discovery has been accelerated by agents that can generate hypotheses, design experiments, collect evidence, and iterate on candidate findings. Recent systems have made substantial progress in scaling this loop, allowing agents to evaluate large numbers of possible discoveries across scientific domains (Huang et al., 2025; Yamada et al., 2025; Jansen et al., 2025; Schmidgall et al., 2025; Liu et al., 2026; Baek et al., 2025; Majumder et al., 2024; Lu et al., 2024; Skarlinski et al., 2024; Gottweis et al., 2025; Spangler et al., 2014; Wang et al., 2024; Yang et al., 2024). This creates a new measurement problem. In closed tasks, an agent's progress can be measured against a fixed objective, answer set, benchmark, or reward (Novikov et al., 2025; Yuksekgonul et al., 2026). However, open-ended scientific discovery offers no oracle specifying which discovery should come next: only a combinatorial space to explore. As agents scale to hundreds or thousands of evaluated hypotheses, progress depends not only on how each candidate is scored, but on what *unit* of scientific knowledge the agent treats as learned.

Many current discovery systems use individual hypotheses as the unit of progress, where a candidate hypothesis is proposed, evaluated against evidence, scored, then retained, revised, or discarded (Gupta et al., 2025; Garikaparthi et al., 2025; Gottweis et al., 2025; Hu et al., 2025; Yang et al., 2025). The score may come from an LLM judge, novelty or feasibility estimate, evidential-support assessment, or belief-shift objective (Agarwal et al., 2026; Baek et al., 2025; Li et al., 2024; Qi et al., 2024; Yang et al., 2025). Bayesian surprisal sharpens this *pointwise* scheme by treating discovery as evidence-induced belief change, offering a formal alternative to subjective interestingness judgments (Agarwal et al., 2026; Shi & Evans, 2023; Itti & Baldi, 2005). Yet the representational unit remains unchanged: the update is attached to a single focal hypothesis and its implicit complement.

This *pointwise* representation is epistemically limiting. Scientific evidence rarely bears on a hypothesis in isolation: it matters because it discriminates among plausible alternatives. This limitation is clearest when evidence rejects a hypothesis. Rejection is scientifically valuable, but *pointwise* rejection does not say where belief should move. If evidence rules out one explanation, it may simultaneously support another; yet *pointwise* discovery collapses all alternatives into an unstructured complement. The agent records that the focal hypothesis failed, while the better-supported explanation remains unrepresented. Under a fixed evidence-generation budget, this makes discovery information-inefficient: **an ex-**

[*]Equal contribution [1] Department of Computer Science, Cornell University, Ithaca, NY, USA. Correspondence to: Ziang Liu <zl873@cornell.edu>.

periment can produce useful evidence without resolving the question it implicitly asks.

We argue that open-ended discovery requires a larger unit of progress: **a scientific question together with the plausible explanations that could answer it**. This is the core insight of contrastive accounts of scientific explanation: evidence is informative not merely because it changes belief in one claim, but because it distinguishes among relevant alternatives (van Fraassen, 1980; Lipton, 2004). We introduce **CONTRASTIVE Discovery**, a framework that operationalizes this principle for LLM-driven open-ended scientific discovery. Each discovery step is represented as a contrast set: a local partition of mutually exclusive and collectively exhaustive (MECE) hypotheses answering the same scientific question. The agent elicits prior and posterior beliefs over the set and scores how evidence redistributes support across competing explanations. A question is *resolved* when this update concentrates support on one explanation; otherwise, the evidence remains informative but the question is *unresolved*. This preserves the belief-shift principle of surprisal-based discovery while upgrading the unit of progress from an isolated claim to a structured question, allowing the agent to identify both the best-supported explanation and the alternatives ruled out by the same evidence.

We evaluate CONTRASTIVE Discovery across six domains spanning materials science, astronomy, mortgage discrimination, online discourse, software engineering practices, and longitudinal social behavior. Under a shared scaffold and evidence-generation budget, *pointwise* baselines are guided by surprisal, novelty, or feasibility, while CONTRASTIVE Discovery is guided by belief shift over contrast sets. Our results show that contrastive representation recovers information that *pointwise* discovery leaves unresolved. Among top-ranked *pointwise* surprisal hypotheses, 89% are rejected; when those same hypotheses are paired with explicit alternatives and evaluated using the same evidence, 73% resolve to an alternative explanation. Running CONTRASTIVE Discovery directly yields more resolved scientific questions across evidence-generation budgets, especially for corrective updates where evidence changes which explanation is best supported. CONTRASTIVE Discovery also matches prior *pointwise* surprisal baselines when evaluated on cumulative surprisal, preserving the domain-expert-aligned belief-shift signal (Agarwal et al., 2026) while resolving alternatives that *pointwise* discovery leaves implicit.

2. Open-Ended, Data-Driven Discovery

Discovery loop. We study open-ended, data-driven scientific discovery, where an agent uses a dataset D to propose hypotheses $h \in \mathcal{H}$ and evaluates them using evidence computed from the data. Unlike supervised prediction or fixed-objective optimization, the target discoveries are not specified in advance: there is no pre-defined reward function that determines the value of a candidate discovery. This setting is open-ended because what constitutes a meaningful discovery is context-dependent.

We model a discovery agent in terms of three components: a proposer π, an evaluator Eval that computes evidence from D, and a scoring rule S that assigns value to a hypothesis given its evidence. At each discovery step t, the agent conditions on the history $\mathcal{A}_{t-1}$ and produces $h_t \sim \pi(\cdot \mid \mathcal{A}_{t-1})$, $e_t = \mathrm{Eval}(h_t; D)$, and $s_t = S(h_t, e_t)$, with the history updated as $\mathcal{A}_t = \mathcal{A}_{t-1} \cup \{(h_t, e_t, s_t)\}$. The process repeats under a budget N. This formulation covers recent LLM-based discovery systems such as (Agarwal et al., 2026).

***Pointwise* discovery.** We call a discovery method *pointwise* when, at step t, the agent proposes a single hypothesis h_t and assesses it on its own. The scoring rule then operates on the singular proposed candidate and its evidence. Many recent LLM-based discovery approaches are *pointwise*, assessing each hypothesis through novelty, feasibility, evidential support, or Bayesian surprise (Agarwal et al., 2026; Garikaparthi et al., 2025; Baek et al., 2025; Gottweis et al., 2025; Hu et al., 2025; Li et al., 2024; Qi et al., 2024; Yang et al., 2025). *Pointwise* discovery is thus a binary instantiation of CONTRASTIVE Discovery, where the only alternative to the proposed hypothesis is its negation (see 3.5).

The rejection case. In *pointwise* discovery, confirmation and rejection have different epistemic consequences. Confirmation identifies a supported explanation; rejection only rules against the proposed one. The evidence may contain information about alternatives, but the *pointwise* score exposes only its effect on h_t. We refer to this limitation as *alternative blindness*: *pointwise* discovery can test whether a proposed explanation survives evidence, but not how that evidence shifts support among alternatives. Unassessed alternatives must therefore be proposed and scored in later steps, again in isolation. Under a finite budget N, resolution among competing explanations requires repeated independent proposals rather than a shared evidence update.

3. CONTRASTIVE Discovery

We reframe a discovery step as resolving a scientific question, rather than validating a single candidate hypothesis. Prior work has made open-ended discovery measurable by scoring evidence-induced belief shift for individual hypotheses (Agarwal et al., 2026); we keep the principle, but change the unit over which belief is represented. Instead of eliciting belief in one hypothesis, we elicit belief over a structured set of plausible explanations for the same question. The alternatives are now part of the discovery unit itself; they determine what the evidence is compared against, how belief is updated, and what outcome the step can produce.

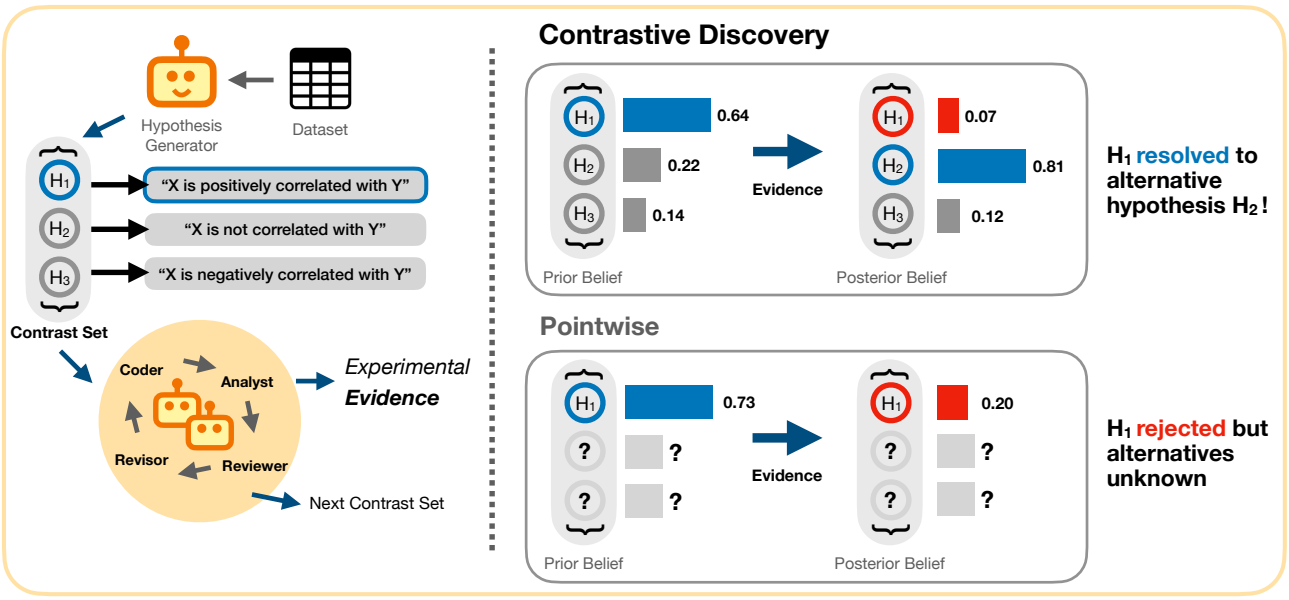

Figure 1. **CONTRASTIVE vs. *pointwise* scientific discovery.** CONTRASTIVE discovery represents each discovery step as a contrast set: a set of competing hypotheses answering the same scientific question. The agent designs and runs an experiment and updates beliefs over all alternatives using the resulting evidence. A question is *resolved* when posterior belief concentrates on one explanation, allowing evidence to both identify the best-supported hypothesis and rule out alternatives. In contrast, *pointwise* discovery evaluates only a focal hypothesis against an implicit complement; when the hypothesis is rejected, the agent learns that the claim is unlikely but does not know which alternative explanation is better supported. See Sections 3 and 4 for details.

3.1. Constructing contrast sets

At each contrastive discovery step t, the agent poses a local scientific question and constructs a contrast set

$$\mathcal{C}_t = \{h_{t,1}, \ldots, h_{t,K}\}, \tag{1}$$

where each $h_{t,k}$ is a plausible explanation for the same question. We require these alternatives to be locally MECE: within the scope of the question, they should represent distinct possible explanations and cover the main outcomes the evidence is meant to distinguish. An example is below:

Example Contrast Set (Materials Project)

Scientific question: How does magnetic ordering affect bulk modulus?

MECE Contrast Set: The effect of magnetic ordering on bulk modulus is

- **H1.** significant only for ferromagnetic materials.
- **H2.** significant only for antiferromagnetic materials.
- **H3.** significant for both ferromagnetic and antiferromagnetic materials.
- **H4.** insignificant for both ferromagnetic and antiferromagnetic materials.

3.2. Eliciting beliefs over contrast sets

Given a contrast set $\mathcal{C}_t$, we next define the belief state used to score evidence-induced change. Before observing evidence, we elicit $p_t^{\text{prior}} \in \Delta^{K-1}$ where Δ^{K-1} is the probability simplex over the K alternatives and $p_{t,k}^{\text{prior}}$ is the subjective belief assigned to hypothesis $h_{t,k}$. After evaluating the candidate and observing evidence e_t, we elicit $p_t^{\text{post}} \in \Delta^{K-1}$. We treat these elicited distributions as the agent's prior and posterior beliefs over the contrast set.

We obtain each distribution through a constant-sum allocation task: the LLM assigns 100 allocation units across the alternatives in $\mathcal{C}_t$, first without and then with access to e_t. Direct elicitation lets the agent interpret the semantic content of the evidence without requiring a hand-specified likelihood model $p(e_t \mid h_{t,k})$. To reduce stochastic variance, we sample $N = 5$ allocations and average the normalized valid responses (see Appendix C). We also randomize the order of alternatives during elicitation to mitigate positional bias in LLM-based evaluation (see Appendix E).

3.3. Scoring contrastive belief shifts

Because the unit of discovery is a belief distribution over competing explanations, the score should measure how evidence changes that distribution. Given the prior and posterior beliefs p_t^{prior} and p_t^{post}, we compute the Kullback–Leibler (KL) divergence of the posterior with respect to the prior, $D_{\text{KL}}(p_t^{\text{post}} \,\|\, p_t^{\text{prior}})$. This direction matches the realized information-gain interpretation after observing e_t. Because contrast sets contain different numbers of alternatives, we report *entropy-scaled* Bayesian surprise, dividing by $\log_2 K$ to reduce mechanical dependence on contrast-set cardinality (this does not bound κ_t by one):

$$\kappa_t = \frac{D_{\text{KL}}(p_t^{\text{post}} \,\|\, p_t^{\text{prior}})}{\log_2 K}. \tag{2}$$

We use κ_t as the contrastive discovery score; larger values indicate that the evidence substantially changes the belief distribution over the competing explanations. For Monte-Carlo tree search we clip the reward at one ($r_t = \min(\kappa_t, 1)$), but

all reported information-gain analyses use the unclipped κ_t (subsection M.2).

3.4. Discovery outcomes

The contrastive score κ_t measures the magnitude of belief shift, but not the type of discovery produced by that shift.

Since the prior and posterior are defined over the same competing explanations, the update can also be characterized by whether evidence confirms the prior-leading explanation, shifts support to another explanation, rules out alternatives, or leaves the question unresolved. Let

$$k_t^{\star,\text{prior}} = \arg\max_k p_{t,k}^{\text{prior}}, \quad k_t^{\star,\text{post}} = \arg\max_k p_{t,k}^{\text{post}} \tag{3}$$

denote the indices of the prior- and posterior-leading explanations. We say the contrast set is *resolved* when the posterior assigns sufficient support to one explanation,

$$\max_k p_{t,k}^{\text{post}} \geq \tau_{\text{support}}, \tag{4}$$

where τ_{support} is the minimum posterior support required to identify a leading explanation. A resolved update is *confirmative* if $k_t^{\star,\text{post}} = k_t^{\star,\text{prior}}$ and *corrective* if $k_t^{\star,\text{post}} \neq k_t^{\star,\text{prior}}$; corrective updates capture the rejection-to-resolution case where evidence moves support from one leading explanation to another. If no explanation exceeds τ_{support}, the outcome is *unresolved*. We separately track alternatives that are rejected or ruled out. An explanation $h_{t,k}$ is ruled out when

$$p_{t,k}^{\text{post}} \leq \tau_{\text{reject}}, \tag{5}$$

where τ_{reject} denotes a low posterior-support threshold. This is not a mutually exclusive outcome class: a single contrastive update can identify a leading explanation while also ruling out competing explanations. For the *pointwise* setting, where a single hypothesis has scalar prior and posterior beliefs, we classify the update analogously: a hypothesis is *confirmed* if its posterior exceeds γ_{support} and *rejected* if it falls below γ_{reject} (threshold calibration details in Appendix F). This taxonomy distinguishes changes in confidence from changes in explanation identity. *Pointwise* discovery can increase or decrease support for one claim; CONTRASTIVE Discovery can also identify when evidence shifts support to a different explanation.

3.5. *Pointwise* discovery as the binary case

Pointwise belief-shift methods (Agarwal et al., 2026) score evidence by how much it changes belief in a single hypothesis h_t. In our notation, this is the binary contrast set

$$\mathcal{C}_t^{\text{point}} = \{h_t, \neg h_t\}, \tag{6}$$

with belief state $p_t = (p_{t,1}, p_{t,2}) \in \Delta^1$, where $p_{t,1}$ is belief in h_t and $p_{t,2} = 1 - p_{t,1}$ is belief in its complement. For *pointwise* belief elicitation, we follow AutoDiscovery's categorical sampling procedure (Agarwal et al., 2026): the elicitor LLM rates h_t on an ordered scale from `definitely_false` to `definitely_true`, which we map to a numeric belief score. We draw $N = 10$ prior and posterior samples, fit Beta distributions to each, compute *pointwise* surprisal as the mean belief shift, and use it as our *pointwise* discovery score (see M.1 for details).

Thus, *pointwise* discovery is a specialized $K = 2$ instance of CONTRASTIVE Discovery. The distinction is that $\neg h_t$ is not an explanation. It is an unstructured complement containing all ways the focal hypothesis could be false. When evidence weakens h_t, *pointwise* discovery can record negative evidence about the claim, but it cannot identify which alternative explanation receives that support. CONTRASTIVE Discovery replaces the unstructured complement with an explicit local partition of alternatives (Equation 1). The score remains a belief shift, but now distributed over competing explanations rather than absorbed by a generic complement.

4. Experimental Setup

We evaluate contrastive discovery in an open-ended, data-driven setting. Given a tabular dataset D and associated metadata, the agent is allocated a fixed budget of $N = 200$ experimental validation attempts. Each attempt consists of exploring a scientific question by generating a contrast set, eliciting prior belief over the set, designing an experiment to evaluate it, executing the corresponding code, validating the resulting evidence, and eliciting posterior belief from that evidence. We use Monte-Carlo tree search (MCTS) (Coulom, 2007), following prior work on agentic scientific discovery and hypothesis search (Agarwal et al., 2026; Yamada et al., 2025; Garikaparthi et al., 2025; Ha et al., 2025; Sprueill et al., 2024; Toledo et al., 2025; Jiang et al., 2025). Search is guided by the upper-confidence bound on trees (UCT) (Kocsis & Szepesvári, 2006), with the clipped contrastive reward $r_t = \min(\kappa_t, 1)$ as the node reward (see M.2 for details).

Our primary evaluation metric is the *resolution rate* of scientific questions. Concretely, this is the fraction of contrast sets for which the experiment concentrates posterior belief strongly enough to identify a winning hypothesis. Among resolved contrast sets, we further distinguish *confirmative* and *corrective* resolutions, corresponding respectively to belief confirmation and belief revision.

Discovery agent scaffold. We build on the AutoDiscovery (Agarwal et al., 2026) architecture. In the *pointwise* setting, the same scaffold generates and evaluates a single hypothesis instead. Belief elicitation is performed by a separate model that serves as a proxy for broad scientific background knowledge. For CONTRASTIVE Discovery, the elicitor assigns a probability distribution over all hypotheses in a contrast set, as described in 3.2. For the *pointwise* setting, it assigns a scalar belief in the validity of an individual hypothesis (see M.1). We use `gpt-4o` as both the discovery agent backbone and the belief elicitor, with implementation details in M.3 (prompts in Appendix J).

Baselines. We compare against *pointwise* baselines under

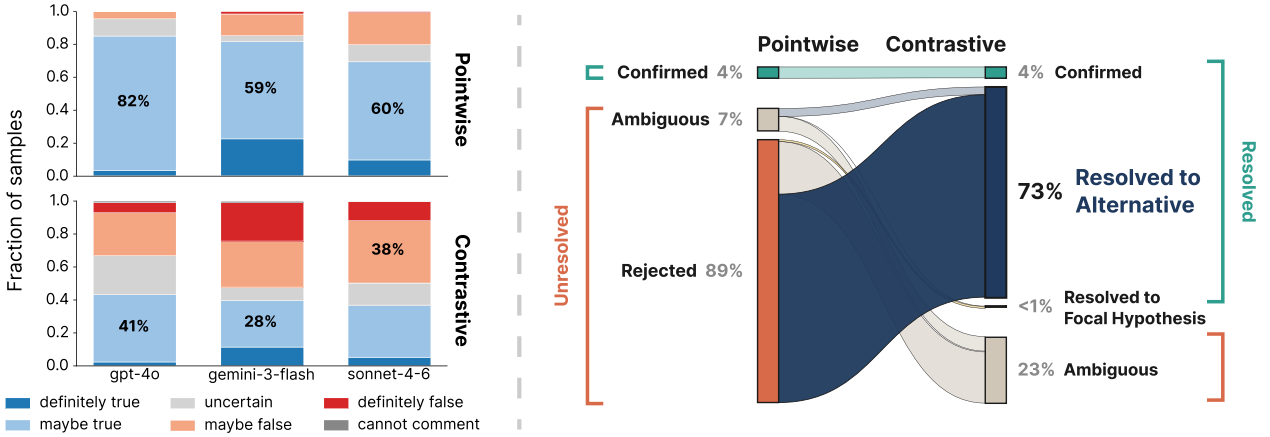

Figure 2. ***Pointwise* discovery masks resolvable alternatives behind rejection.** ***Left:*** *pointwise* proposals exhibit prior collapse, with most mass concentrated in the "maybe true" bin, while CONTRASTIVE hypotheses cover a broader range of prior beliefs. ***Right:*** among the Top-50 hypotheses ranked by *pointwise* surprisal, 89% are rejected under *pointwise* evaluation. Re-adjudicating the same hypotheses with a post-hoc CONTRASTIVE companion and identical evidence resolves many of them to specific alternatives. This shows that *pointwise* rejection often leaves alternative-resolving information unexposed. Per-dataset breakdowns in Figure 13.

the same $N = 200$ evidence-generation budget. In this setting, each validation attempt proposes a single hypothesis and guides MCTS using scalar surprisal (see M.1). This holds fixed the number of evidence-generating loops and asks whether CONTRASTIVE Discovery obtains more scientific resolution per validation attempt by discriminating among alternatives. To provide additional *pointwise* comparisons, we additionally guide MCTS using LLM-judge scores for hypothesis novelty and feasibility, following prior work on hypothesis generation and evaluation (Garikaparthi et al., 2025; Baek et al., 2025; Gottweis et al., 2025; Hu et al., 2025; Li et al., 2024; Qi et al., 2024; Yang et al., 2025) (judge prompts in J.3).

Datasets. We evaluate on six datasets spanning scientific and social domains: Materials Project (Horton et al., 2025), SDSS Galaxy (Collaboration et al., 2025), BLADE Mortgage (Gu et al., 2024), BLADE Conversation (Gu et al., 2024), DiscoveryBench Requirements Engineering (Majumder et al., 2025), and DiscoveryBench NLS Raw (Majumder et al., 2025). These datasets vary in feature spaces, sample sizes, and domain structure, enabling evaluation across diverse discovery settings. Dataset statistics, preprocessing, and filtering criteria are reported in Appendix H.

Robustness. We report results on three independent runs for Materials Project, SDSS Galaxy, Mortgage, Conversation, and one run on the remaining two highest-cost datasets.

5. Results

The experiments evaluate whether CONTRASTIVE Discovery changes the epistemic yield of the discovery process under a fixed evidence-generation budget. We categorize outcomes by their epistemic role: confirmed or rejected *pointwise* hypotheses, resolved or unresolved contrast sets, and ruled-out (rejected) alternatives. This lets us test the central claim that CONTRASTIVE Discovery produces more resolved scientific questions. The evaluation proceeds in three phases: we first diagnose what *pointwise* surprisal captures and misses, then measure resolved-knowledge yield across methods, and finally analyze the mechanisms and metric disagreements underlying the observed gains.

5.1. *Pointwise* rejections conceal CONTRASTIVE resolutions

We first ask what high-scoring *pointwise* surprisal candidates actually represent. Figure 2 (left) shows that *pointwise* hypothesis proposals concentrate heavily in the "maybe true" prior bin across three frontier belief elicitors (>60% for each). This concentration is strongest when the proposer and elicitor are the same model (i.e., `gpt-4o` is 82%), but remains pronounced under cross-model elicitation. In contrast, hypotheses drawn from contrast sets produce a much broader prior distribution, with the largest bin containing only 41% of mass (`gpt-4o`). This suggests that the *pointwise* pipeline tends to propose plausibility-centered hypotheses, whereas contrast sets cover a wider range of local outcomes for the same scientific question.

This prior "collapse" shapes what *pointwise* surprisal selects downstream. Because most *pointwise* hypotheses begin near the "maybe true" bin, there is limited headroom for large confirmatory shifts, whereas disconfirming evidence can move belief much farther downward. High-surprisal *pointwise* candidates are therefore disproportionately negative: among the top-50 candidates selected from 200 nodes, only 4% are confirmed, 7% are ambiguous, and 89% are rejected. Thus, *pointwise* surprisal primarily surfaces evidence against focal claims. This is useful negative evidence,

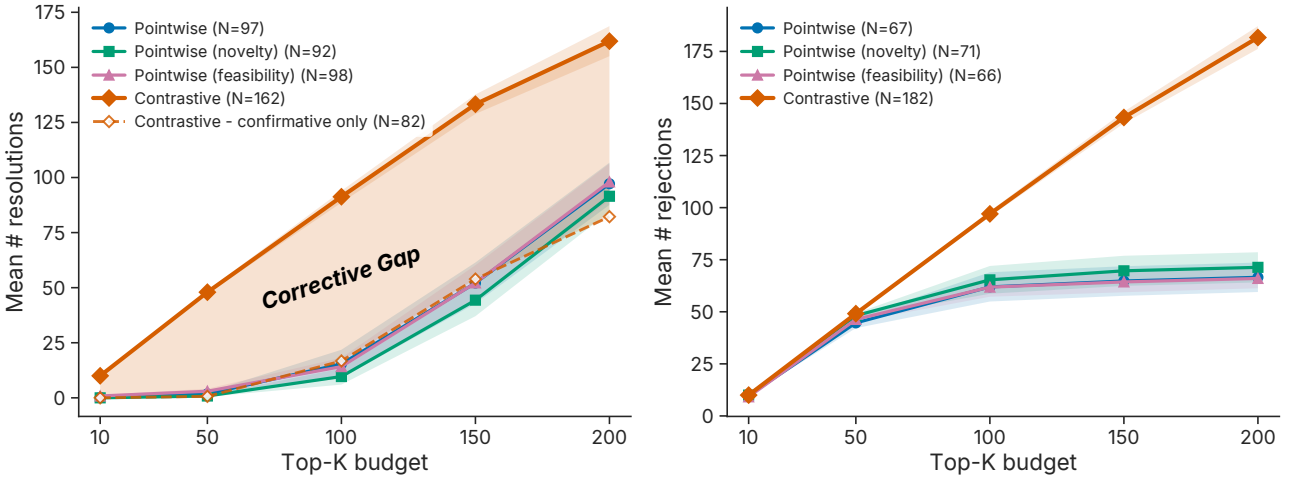

Figure 3. **CONTRASTIVE Discovery improves resolved-knowledge yield.** ***Left:*** CONTRASTIVE Discovery resolves more scientific questions across top-K budgets; the shaded gap marks corrective discoveries unavailable to *pointwise* baselines. ***Right:*** the same contrastive update can also rule out competing alternatives, increasing rejection yield as part of the same evidence update.

but because the complement is unstructured, it does not identify which alternative explanation is better supported.

To test whether those rejected hypotheses contain recoverable information, we construct a post-hoc contrastive companion for each *pointwise* hypothesis: we keep the original hypothesis as one candidate explanation, generate competing alternatives using the same contrast-set proposer, and reuse the same evidence. As shown in Figure 2 (right), under contrastive adjudication, many *pointwise* rejections become resolved contrastive outcomes: 73% of the hypotheses resolve to an alternative, and unresolved outcomes drop from 96% under *pointwise* confirmation-only resolution to 23% under the contrastive companion. The bottleneck is therefore not evidence quality, but the representation of the hypothesis space. *Pointwise* discovery often spends budget to reject a claim; CONTRASTIVE Discovery can use the same evidence to identify what is more plausible instead. We next evaluate whether this diagnostic advantage translates into higher resolved-knowledge yield under budget.

5.2. CONTRASTIVE Discovery increases resolution without sacrificing rejection

After showing that *pointwise* rejection often contains alternative-resolving information, we now test whether running CONTRASTIVE Discovery directly converts this latent information into higher resolved-knowledge yield. For each method, we rank candidates by its discovery score (κ_t or surprisal) and evaluate the top-K scientific questions retained for inspection. We observe a clear ceiling among *pointwise* baselines (Figure 3, left). Whether candidates are ranked by surprisal, novelty, or feasibility, the number of resolved questions remains similar. This indicates that the limitation is not only the particular *pointwise* score, but the representation: each method can only confirm or reject an isolated hypothesis. In contrast, CONTRASTIVE Discovery produces substantially more resolved questions across budgets.

The confirmative-only CONTRASTIVE curve shows that the gain is not driven by simply confirming more prior-leading explanations: when restricted to confirmative resolutions, CONTRASTIVE Discovery closely tracks the *pointwise* baselines. The additional yield comes almost entirely from corrective resolutions, where evidence shifts support from the prior-leading explanation to another member of the contrast set. This corrective gap is exactly the information lost in *pointwise* discovery. There, the same evidence can reject the focal hypothesis, but because the alternative space is collapsed into an unstructured complement, it cannot identify which competing explanation becomes better supported.

In addition, Figure 3 (right) shows that higher resolution does not come at the expense of rejection: CONTRASTIVE Discovery produces more ruled-out alternatives while also resolving more questions. This is because in *pointwise* discovery, a candidate can only be confirmed, rejected, or left ambiguous; in CONTRASTIVE Discovery, the same evidence is distributed across a local partition, so one explanation can become strongly supported while others are simultaneously ruled out. Rejection is thus not a competing outcome, but a byproduct of successful contrastive resolution.

5.3. How does CONTRASTIVE Discovery improve resolved-knowledge yield?

We isolate three factors in the discovery pipeline: *ranking* (*pointwise* surprisal vs. CONTRASTIVE κ_t), *evaluation* (focal hypothesis only vs. full contrast set), and *discovery-time construction* (contrast set built post-hoc vs. during the discovery run). We compare five variants: *pointwise* only, +CONTRASTIVE ranking, +CONTRASTIVE evalu-

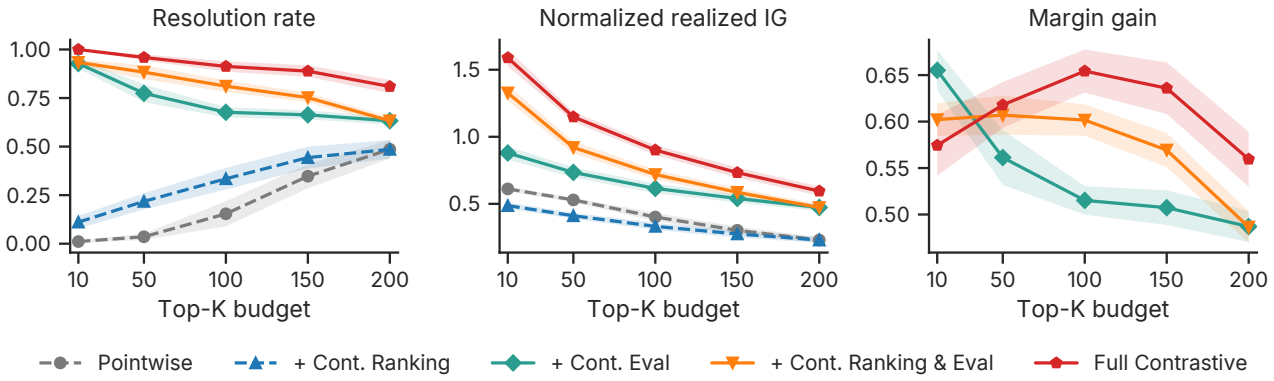

Figure 4. **CONTRASTIVE ranking, evaluation, and discovery-time construction all contribute to resolved-knowledge yield.** Across resolution rate **(left)**, cardinality-normalized realized information gain (IG; $\mathrm{KL}/\log_2 K$, not bounded by 1) (middle), and margin gain **(right)**, full CONTRASTIVE performs best. CONTRASTIVE ranking improves resolution even under *pointwise* evaluation, contrastive evaluation produces a larger step change by allowing evidence to resolve over alternatives, and combining both gives stronger updates than either alone. Since *pointwise* evaluation has no explicit runner-up, margin gain is defined only for CONTRASTIVE-evaluated variants.

ation, +CONTRASTIVE ranking & evaluation, and Full CONTRASTIVE. Post-hoc variants reuse evidence collected for a *pointwise* hypothesis with a companion contrast set; Full CONTRASTIVE constructs the contrast set during discovery-time. We evaluate on three metrics: resolution rate measures whether the update answers the question, cardinality-normalized realized information gain (IG; $\kappa_t = \mathrm{KL}/\log_2 K$, not bounded by 1) measures the magnitude of belief shift, and margin gain measures separation between the leading explanation and runner-up (Figure 4).

The largest separation comes from changing how evidence is evaluated. Variants with *pointwise* evaluation remain low in resolution rate, whereas variants with CONTRASTIVE evaluation improve substantially. Once alternatives are explicitly represented, the same evidence can adjudicate among explanations rather than only confirm or reject a focal hypothesis. CONTRASTIVE ranking also helps, but only when paired with the right evaluator. Under *pointwise* evaluation, CONTRASTIVE ranking improves resolution but yields lower information gain (Figure 4, middle), revealing a ranker–evaluator mismatch: the ranker prioritizes evidence informative over alternatives, while the evaluator collapses that evidence back onto a single focal claim. When CONTRASTIVE ranking is paired with CONTRASTIVE evaluation, both resolution and realized information gain increase.

Finally, post-hoc companion sets improve over *pointwise* evaluation, but Full CONTRASTIVE performs best because constructing alternatives during discovery lets them shape the question, analysis plan, and evidence collected. Full CONTRASTIVE leads on all metrics, showing that resolved-knowledge yield is driven primarily by contrastive evaluation, strengthened by contrastive ranking, and maximized when alternatives shape evidence collection from the start.

5.4. CONTRASTIVE belief shift preserves surprisal while revealing corrective updates

We next compare *pointwise* surprisal and CONTRASTIVE belief-shift (κ_t) on the same set of candidate discoveries. For each node from a CONTRASTIVE run, we compute κ_t over the full contrast set and a post-hoc *pointwise* surprisal score for the highest prior hypothesis. This tests whether κ_t is merely a rescaled version of *pointwise* surprisal or instead prioritizes different discovery semantics. The two scores are broadly aligned (Spearman $\rho = 0.675$) with 90% of nodes falling within a ± 0.4 percentile-rank difference band (Figure 5). Thus, CONTRASTIVE belief shift preserves the core belief-shift signal captured by surprisal.

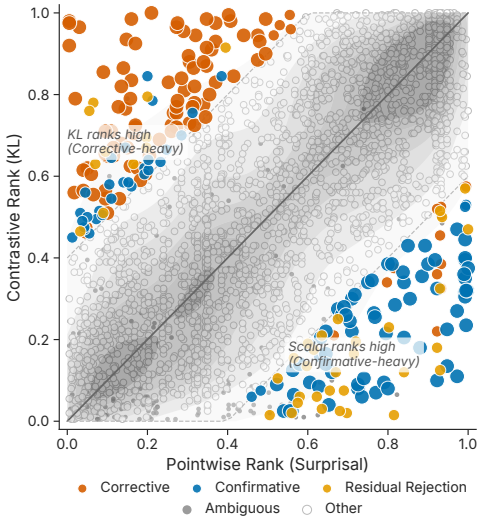

Figure 5. ***Pointwise* surprisal and CONTRASTIVE belief shift are broadly aligned, but their disagreements are structured** CONTRASTIVE-high/*pointwise*-low nodes are enriched for corrective updates, while *pointwise*-high/CONTRASTIVE-low nodes are enriched for confirmative updates.

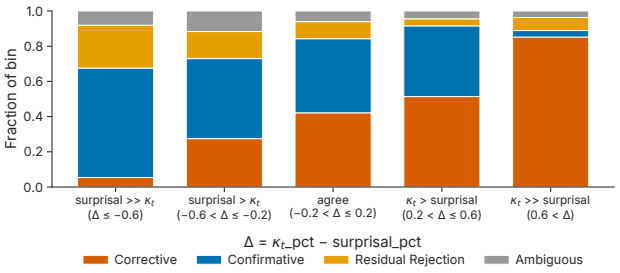

Figure 6. **Candidate discoveries grouped by percentile-rank difference.** Corrective outcomes increase as rankings shift CONTRASTIVE; confirmative increases in the reverse direction.

The disagreements, however, are structured. When the metrics largely disagree (i.e., $|\Delta_{\text{pct}\geq 0.4}|$), nodes ranked higher by CONTRASTIVE are disproportionately corrective, while nodes ranked higher by *pointwise* are more frequently confirmative. This pattern does not imply that *pointwise* discovery is mostly confirmative; 5.1 showed the opposite. Rather, it shows that *pointwise* surprisal tracks movement in one focal claim, while CONTRASTIVE belief shift tracks redistribution across competing explanations.

The binned analysis in Figure 6 confirms this structure: corrective outcomes increase as rank disagreement shifts to CONTRASTIVE, while confirmative outcomes increase as it shifts to *pointwise*. The agreement region captures the shared signal, where both metrics identify the same updates as high-information. Thus, CONTRASTIVE belief shift keeps what *pointwise* surprisal measures well, while recovering the corrective updates that *pointwise* surprisal tends to miss.

5.5. CONTRASTIVE Discovery improves resolution without sacrificing surprisal

Section 5.4 showed CONTRASTIVE belief shift and *pointwise* surprisal diverge in semantically meaningful cases. We test whether this divergence comes at a cost of performance under the surprisal metric used in prior work (Agarwal et al., 2026). For each CONTRASTIVE hypothesis, we project the contrast set to its highest prior hypothesis and evaluate it using *pointwise* surprisal. Under this single-hypothesis projection, CONTRASTIVE remains competitive with *pointwise* surprisal, novelty, and feasibility baselines (Figure 7).

As an optimistic diagnostic, we also score every hypothesis in each contrast set and count the set as surprising if any member is *pointwise*-surprising. This max-over-set curve is substantially higher, suggesting that contrast sets contain additional surprising hypotheses otherwise missed by single-hypothesis evaluation. Together, these results show that CONTRASTIVE Discovery does not trade off the domain-expert-aligned surprisal signal validated by AutoDiscovery (Agarwal et al., 2026). It preserves *pointwise* surprisal yield while adding what *pointwise* evaluation cannot provide: structured resolution over competing explanations.

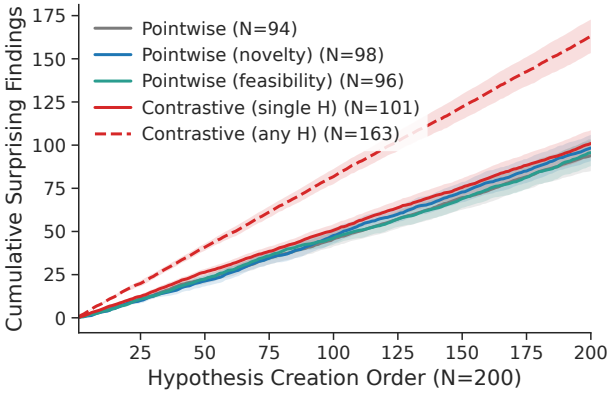

Figure 7. **CONTRASTIVE Discovery preserves *pointwise* surprisal yield.** Projected to the highest prior hypothesis, CONTRASTIVE matches *pointwise* baselines under the surprisal metric used in prior work.

6. Additional Related Work

LLM-based agents have accelerated the shift from task-specific scientific automation tools toward increasingly autonomous discovery systems (Zheng et al., 2025). Recent systems integrate hypothesis generation, experiment design, code execution, evidence collection, and iterative refinement (Yamada et al., 2025; Lu et al., 2024; Jansen et al., 2025; Schmidgall et al., 2025; Huang et al., 2025; Liu et al., 2026). Much of this work represents discovery progress through individual hypotheses, which are scored by scalar objectives such as surprisal, novelty, feasibility, diversity, or ranking-based preference signals (Agarwal et al., 2026; Garikaparthi et al., 2025; Baek et al., 2025; Gottweis et al., 2025; Hu et al., 2025; Li et al., 2024; Qi et al., 2024; Yang et al., 2025; Gupta et al., 2025; Duan et al., 2025; Alkan et al., 2025; Liusie et al., 2024; Ke et al., 2025; Gao et al., 2024). Although such systems may maintain many candidate hypotheses during search, these candidates are typically evaluated *pointwise* as separate claims. In CONTRASTIVE Discovery, alternatives are not merely stored as separate nodes; they form an explicit local partition of the same scientific question, and evidence is scored by how it redistributes belief across that partition.

Our framework therefore changes the represented unit of progress from a single hypothesis to a contrast set of competing explanations, allowing evidence to resolve among alternatives rather than only score an isolated claim. This builds on a longer tradition of information-theoretic discovery and experimental design, where experiments are valued by how much they are expected to change belief (Lindley, 1956; MacKay, 1992; Chaloner & Verdinelli, 1995). Our belief-shift objective is similarly grounded in Bayesian surprise, which formalizes surprise as posterior–prior belief change (Itti & Baldi, 2005; Baldi & Itti, 2010).

A related line of work focuses on generating research ideas or hypotheses from scientific literature (Si et al., 2024; Spangler et al., 2014; Baek et al., 2025; Wang et al., 2024; Yang et al., 2024). CONTRASTIVE Discovery is compatible with such settings, since contrastive resolution only requires a scientific question and a set of plausible alternatives. In this paper, we focus on open-ended, data-driven discovery over tabular datasets (Majumder et al., 2024; 2025; Gu et al., 2024; Liu et al., 2026), where experiments can be executed directly against rich scientific data.

7. Conclusion and Limitations

We introduced CONTRASTIVE Discovery, a framework that reframes open-ended discovery from scoring isolated hypotheses to resolving scientific questions over explicit alternatives. Across six data-driven domains, this representation turns rejection-heavy pointwise hypotheses into specific, supported explanations, yielding more resolved scientific questions under the same evidence-generation budget. The results indicate that contrastive structure allows evidence that would otherwise reject a hypothesis to instead identify which alternative explanation is better supported.

Our experiments focus on tabular datasets where hypotheses can be evaluated through statistical analyses, while many scientific discoveries require laboratory experiments, simulation, or longer-horizon intervention. While we cannot guarantee that the generated contrast set is MECE, empirically we find that $> 96\%$ are structurally MECE (See D). Resolution is also measured through elicited model beliefs rather than external ground truth, so our results are best read as model-adjudicated explanatory resolution; cross-elicitor agreement (subsection B.1) supports their robustness, and independent expert validation remains future work. Finally, our search procedure is one instantiation of the framework, and future work could incorporate explicit diversity objectives, richer memory, or scientist-specified contrast families. More broadly, our results suggest that agentic science should represent discovery not as a stream of accepted or rejected claims, but as an evolving map of competing explanations.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. This work may accelerate data-driven hypothesis generation, but it also inherits risks associated with LLM-based scientific reasoning, including spurious hypotheses, overreliance on model-adjudicated evidence, and amplification of biases in source datasets. We view CONTRASTIVE Discovery as a tool for prioritizing candidate explanations, not as a replacement for expert validation.

References

Agarwal, D., Majumder, B. P., Adamson, R., Chakravorty, M., Gavireddy, S. R., Parashar, A., Surana, H., Mishra, B. D., McCallum, A., Sabharwal, A., and Clark, P. Autodiscovery: Open-ended scientific discovery via bayesian surprise, 2026. URL https://arxiv.org/abs/2507.00310.

Alkan, A. K., Sourav, S., Jablonska, M., Astarita, S., Chakrabarty, R., Garuda, N., Khetarpal, P., Pióro, M., Tanoglidis, D., Iyer, K. G., Polimera, M. S., Smith, M. J., Ghosal, T., Huertas-Company, M., Kruk, S., Schawinski, K., and Ciucă, I. A survey on hypothesis generation for scientific discovery in the era of large language models, 2025. URL https://arxiv.org/abs/2504.05496.

Baek, J., Jauhar, S. K., Cucerzan, S., and Hwang, S. J. ResearchAgent: Iterative research idea generation over scientific literature with large language models. In Chiruzzo, L., Ritter, A., and Wang, L. (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6709–6738, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.342. URL https://aclanthology.org/2025.naacl-long.342/.

Baldi, P. and Itti, L. Of bits and wows: A bayesian theory of surprise with applications to attention. *Neural Netw.*, 23 (5):649–666, June 2010. ISSN 0893-6080. doi: 10.1016/j.neunet.2009.12.007. URL https://doi.org/10.1016/j.neunet.2009.12.007.

Chaloner, K. and Verdinelli, I. Bayesian Experimental Design: A Review. *Statistical Science*, 10(3):273 – 304, 1995. doi: 10.1214/ss/1177009939. URL https://doi.org/10.1214/ss/1177009939.

Collaboration, S., Pallathadka, G. A., Aghakhanloo, M., Aird, J., Almeida, A., Amrita, S., Anders, F., Anderson, S. F., Arseneau, S., Avila, C. G., Aviram, S., Aydar, C., Badenes, C., Barrera-Ballesteros, J. K., Bauer, F. E., Behmard, A., Berg, M., Besser, F., Bidin, C. M., Bizyaev, D., Blanc, G., Blanton, M. R., Bovy, J., Brandt, W. N., Brownstein, J. R., Buchner, J., Bulbul, E., Burchett, J. N., Carigi, L., Carlberg, J. K., Casey, A. R., Chakraborty, P., Chanamé, J., Chandra, V., Chiappini, C., Chilingarian, I., Comparat, J., Covey, K., Crumpler, N., Cunha, K., D'Onghia, E., Dai, X., Darling, J., Davis, M., Lee, N. D., Deacon, N., Delgado, J. E. M., Demasi, S., Demianenko, M., Demke, D., Donor, J., Drory, N., Durango, M. A. V., Dwelly, T., Egorov, O., Egorova, E., El-Badry, K., Eracleous, M., Fan, X., Farr, E., Finkbeiner, D. P., Fries, L.,

Frinchaboy, P., Fusillo, N. P. G., Félix, L. D. S., Gaensicke, B., Galligan, E., García, P., Gelfand, J., Grabowski, K., Grebel, E., Green, P. J., Greve, H., Grier, C., Griffith, E., Guetzoyan, P., Gupta, P., Hackshaw, Z., Hall, P. B., Hawkins, K., Hegedűs, V., Hekker, S., Herbst, T. M., Hermes, J. J., Hernández-García, L., Hiremath, P., Hogg, D. W., Holtzman, J., Horne, K., Horta, D., Huang, Y., Hutchinson, B., Häberle, M., Ibarra-Medel, H. J., Ji, A. P., Jofre, P., Johnson, J. W., Johnson, J., Johnston, E. J., Kaldor, M., Katkov, I., Khalatyan, A., Khoperskov, S., Klessen, R., Kluge, M., Koekemoer, A. M., Kollmeier, J. A., Kounkel, M., Kreckel, K., Krishnarao, D., Krumpe, M., Lacerna, I., Laporte, C., Lepine, S., Li, J., Liang, F.-H., Limberg, G., Liu, X., Loebman, S., Long, K., Lu, Y., Lucey, M., Lugo-Aranda, A. Z., Martinez-Aldama, M. L. M., McKinnon, K., Medan, I., Merloni, A., Morrison, S., Myers, N., Mészáros, S., Müller-Horn, J., Nepal, S., Ness, M., Nidever, D., Nitschelm, C., Oravetz, A., Otto, J., Pan, K., Paolino, F. P., Peñaloza, C. A. N., Pinsonneault, M., Popp, M. T., Price-Whelan, A., Pulatova, N., Queiroz, A. B., Raddick, J., Rankine, A., Rix, H.-W., Román-Zúñiga, C., Rosso, D. F., Runnoe, J., Saad, S. M., Salvato, M., Sanchez, S. F., Sattler, N., Saydjari, A., Sayres, C., Schlaufman, K., Schneider, D. P., Schwope, A., Seaton, L. M., Seeburger, R., Serna, J., Sharma, S., Shen, Y., Sinha, A., Sizemore, B., Sniegowska, M., Song, Y., Souto, D., Stassun, K., Steinmetz, M., Stone, Z., Stone-Martinez, A., Stringfellow, G. S., Sánchez, A. M., Sánchez-Gallego, J., Tan, J., Tayar, J., Thai, R., Thakar, A., Thibodeaux, P., Ting, Y.-S., Tkachenko, A., Trakhtenbrot, B., Trincado, J. G. F., Troup, N., Trump, J. R., Ulloa, N., der Marel, R. P. V., Vera, P., Villanova, S., Villaseñor, J., Wang, J., Way, Z., Weijmans, A.-M., Wheeler, A., Wilson, J. C., Wofford, A., Wong, T., Wu, Q., Wylezalek, D., Xue, X.-X., Yan, R., Yang, Q., Zakamska, N., Zari, E., Zasowski, G., Zeltyn, G., Zheng, Z., Zucker, C., and de J. Zermeño, R. The nineteenth data release of the sloan digital sky survey, 2025. URL https://arxiv.org/abs/2507.07093.

Coulom, R. Efficient selectivity and backup operators in monte-carlo tree search. In van den Herik, H. J., Ciancarini, P., and Donkers, H. H. L. M. J. (eds.), *Computers and Games*, pp. 72–83, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-75538-8.

Duan, S., Tian, Y., Bing, Q., and Shao, X. Bayes-entropy collaborative driven agents for research hypotheses generation and optimization, 2025. URL https://arxiv.org/abs/2508.01746.

Gao, Y., Gu, N., Lam, J., Henderson, J., and Hahnloser, R. Evaluating unsupervised argument aligners via generation of conclusions of structured scientific abstracts. In Graham, Y. and Purver, M. (eds.), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 151–160, St. Julian's, Malta, March 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.eacl-short.14. URL https://aclanthology.org/2024.eacl-short.14/.

Garikaparthi, A., Patwardhan, M., Vig, L., and Cohan, A. IRIS: Interactive research ideation system for accelerating scientific discovery. In Mishra, P., Muresan, S., and Yu, T. (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 592–603, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-253-4. doi: 10.18653/v1/2025.acl-demo.57. URL https://aclanthology.org/2025.acl-demo.57/.

Gottweis, J., Weng, W.-H., Daryin, A., Tu, T., Palepu, A., Sirkovic, P., Myaskovsky, A., Weissenberger, F., Rong, K., Tanno, R., Saab, K., Popovici, D., Blum, J., Zhang, F., Chou, K., Hassidim, A., Gokturk, B., Vahdat, A., Kohli, P., Matias, Y., Carroll, A., Kulkarni, K., Tomasev, N., Guan, Y., Dhillon, V., Vaishnav, E. D., Lee, B., Costa, T. R. D., Penadés, J. R., Peltz, G., Xu, Y., Pawlosky, A., Karthikesalingam, A., and Natarajan, V. Towards an ai co-scientist, 2025. URL https://arxiv.org/abs/2502.18864.

Gu, K., Shang, R., Jiang, R., Kuang, K., Lin, R.-J., Lyu, D., Mao, Y., Pan, Y., Wu, T., Yu, J., Zhang, Y., Zhang, T. M., Zhu, L., Merrill, M. A., Heer, J., and Althoff, T. BLADE: Benchmarking language model agents for data-driven science. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 13936–13971, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.815. URL https://aclanthology.org/2024.findings-emnlp.815/.

Gupta, J. S., Tawseeq, S. M., I, H. S., Singh, S. K., Singla, Y. K., Doermann, D., Shah, R. R., and Krishnamurthy, B. Experigen: Data driven hypothesis generation with experimental verification, 2025. URL https://openreview.net/forum?id=LKGflZgEKB.

Ha, R., Li, C., Pu, R., Zhang, L., Zhang, X., and Su, S. DSG-MCTS: A dynamic strategy-guided Monte Carlo tree search for diversified reasoning in large language models. In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V. (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 10530–10544, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN

979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.532. URL https://aclanthology.org/2025.emnlp-main.532/.

Horton, M. K., Huck, P., Yang, R. X., Munro, J. M., Dwaraknath, S., Ganose, A. M., Kingsbury, R. S., Wen, M., Shen, J. X., Mathis, T. S., Kaplan, A. D., Berket, K., Riebesell, J., George, J., Rosen, A. S., Spotte-Smith, E. W. C., McDermott, M. J., Cohen, O. A., Dunn, A., Kuner, M. C., Rignanese, G.-M., Petretto, G., Waroquiers, D., Griffin, S. M., Neaton, J. B., Chrzan, D. C., Asta, M., Hautier, G., Cholia, S., Ceder, G., Ong, S. P., Jain, A., and Persson, K. A. Accelerated data-driven materials science with the materials project. *Nature Materials*, 24(10):1522–1532, Oct 2025. ISSN 1476-4660. doi: 10.1038/s41563-025-02272-0. URL https://doi.org/10.1038/s41563-025-02272-0.

Hu, X., Fu, H., Wang, J., Wang, Y., Li, Z., Xu, R., Lu, Y., Jin, Y., Pan, L., and Lan, Z. NOVA: An iterative planning framework for enhancing scientific innovation with large language models. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T. (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 21330–21359, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1099. URL https://aclanthology.org/2025.findings-acl.1099/.

Huang, K., Jin, Y., Li, R., Li, M. Y., Candes, E., and Leskovec, J. Automated hypothesis validation with agentic sequential falsifications. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=iTevNo8PzG.

Itti, L. and Baldi, P. Bayesian surprise attracts human attention. In Weiss, Y., Schölkopf, B., and Platt, J. (eds.), *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2005. URL https://proceedings.neurips.cc/paper_files/paper/2005/file/0172d289da48c48de8c5ebf3de9f7ee1-Paper.pdf.

Jansen, P., Tafjord, O., Radensky, M., Siangliulue, P., Hope, T., Mishra, B. D., Majumder, B. P., Weld, D. S., and Clark, P. Codescientist: End-to-end semi-automated scientific discovery with code-based experimentation, 2025. URL https://arxiv.org/abs/2503.22708.

Jiang, Z., Schmidt, D., Srikanth, D., Xu, D., Kaplan, I., Jacenko, D., and Wu, Y. Aide: Ai-driven exploration in the space of code, 2025. URL https://arxiv.org/abs/2502.13138.

Ke, Y., George, K., Pandya, K., Blumenthal, D., Sprang, M., Großmann, G., Vollmer, S., and Selby, D. A. Biodisco: Multi-agent hypothesis generation with dual-mode evidence, iterative feedback and temporal evaluation, 2025. URL https://arxiv.org/abs/2508.01285.

Kocsis, L. and Szepesvári, C. Bandit based monte-carlo planning. In *Proceedings of the 17th European Conference on Machine Learning*, ECML'06, pp. 282–293, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 354045375X. doi: 10.1007/11871842_29. URL https://doi.org/10.1007/11871842_29.

Li, R., Jing, L., and Du, X. Learning to generate research idea with dynamic control, 2024. URL https://openreview.net/forum?id=zCb0dPvGYN.

Lindley, D. V. On a Measure of the Information Provided by an Experiment. *The Annals of Mathematical Statistics*, 27(4):986 – 1005, 1956. doi: 10.1214/aoms/1177728069. URL https://doi.org/10.1214/aoms/1177728069.

Lipton, P. *Inference to the Best Explanation*. Routledge, 2nd edition, 2004. ISBN 9780415242035.

Liu, H., Huang, S., Hu, J., Zhou, Y., and Tan, C. Hypobench: Towards systematic and principled benchmarking for hypothesis generation, 2026. URL https://arxiv.org/abs/2504.11524.

Liusie, A., Manakul, P., and Gales, M. J. F. Llm comparative assessment: Zero-shot nlg evaluation through pairwise comparisons using large language models, 2024. URL https://arxiv.org/abs/2307.07889.

Lu, C., Lu, C., Lange, R. T., Foerster, J., Clune, J., and Ha, D. The ai scientist: Towards fully automated open-ended scientific discovery, 2024. URL https://arxiv.org/abs/2408.06292.

MacKay, D. J. C. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, Jul 1992. ISSN 0899-7667. doi: 10.1162/neco.1992.4.4.590.

Majumder, B. P., Surana, H., Agarwal, D., Hazra, S., Sabharwal, A., and Clark, P. Data-driven discovery with large generative models, 2024. URL https://arxiv.org/abs/2402.13610.

Majumder, B. P., Surana, H., Agarwal, D., Mishra, B. D., Meena, A., Prakhar, A., Vora, T., Khot, T., Sabharwal, A., and Clark, P. Discoverybench: Towards data-driven discovery with large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=vyflgpwfJW.

Novikov, A., Vũ, N., Eisenberger, M., Dupont, E., Huang, P.-S., Wagner, A. Z., Shirobokov, S., Kozlovskii, B., Ruiz, F. J. R., Mehrabian, A., Kumar, M. P., See, A., Chaudhuri, S., Holland, G., Davies, A., Nowozin, S., Kohli, P., and Balog, M. Alphaevolve: A coding agent for scientific and algorithmic discovery, 2025. URL https://arxiv.org/abs/2506.13131.

Qi, B., Zhang, K., Tian, K., Li, H., Chen, Z.-R., Zeng, S., Hua, E., Jinfang, H., and Zhou, B. Large language models as biomedical hypothesis generators: A comprehensive evaluation. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=q36rpGlG9X.

Schmidgall, S., Su, Y., Wang, Z., Sun, X., Wu, J., Yu, X., Liu, J., Moor, M., Liu, Z., and Barsoum, E. Agent laboratory: Using llm agents as research assistants, 2025. URL https://arxiv.org/abs/2501.04227.

Shi, F. and Evans, J. Surprising combinations of research contents and contexts are related to impact and emerge with scientific outsiders from distant disciplines. *Nature Communications*, 14(1):1641, Mar 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-36741-4. URL https://doi.org/10.1038/s41467-023-36741-4.

Si, C., Yang, D., and Hashimoto, T. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers, 2024. URL https://arxiv.org/abs/2409.04109.

Skarlinski, M. D., Cox, S., Laurent, J. M., Braza, J. D., Hinks, M., Hammerling, M. J., Ponnapati, M., Rodriques, S. G., and White, A. D. Language agents achieve superhuman synthesis of scientific knowledge, 2024. URL https://arxiv.org/abs/2409.13740.

Spangler, S., Wilkins, A. D., Bachman, B. J., Nagarajan, M., Dayaram, T., Haas, P., Regenbogen, S., Pickering, C. R., Comer, A., Myers, J. N., Stanoi, I., Kato, L., Lelescu, A., Labrie, J. J., Parikh, N., Lisewski, A. M., Donehower, L., Chen, Y., and Lichtarge, O. Automated hypothesis generation based on mining scientific literature. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pp. 1877–1886, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450329569. doi: 10.1145/2623330.2623667. URL https://doi.org/10.1145/2623330.2623667.

Sprueill, H. W., Edwards, C., Agarwal, K., Olarte, M. V., Sanyal, U., Johnston, C., Liu, H., Ji, H., and Choudhury, S. Chemreasoner: Heuristic search over a large language model's knowledge space using quantum-chemical feedback, 2024. URL https://arxiv.org/abs/2402.10980.

Toledo, E., Hambardzumyan, K., Josifoski, M., Hazra, R., Baldwin, N., Audran-Reiss, A., Kuchnik, M., Magka, D., Jiang, M., Lupidi, A. M., Lupu, A., Raileanu, R., Niu, K., Shavrina, T., Gagnon-Audet, J.-C., Shvartsman, M., Sodhani, S., Miller, A. H., Charnalia, A., Dunfield, D., Wu, C.-J., Stenetorp, P., Cancedda, N., Foerster, J. N., and Bachrach, Y. Ai research agents for machine learning: Search, exploration, and generalization in mle-bench, 2025. URL https://arxiv.org/abs/2507.02554.

van Fraassen, B. C. *The Scientific Image*. Oxford University Press, 1980. ISBN 9780191597473.

Wang, Q., Downey, D., Ji, H., and Hope, T. Scimon: Scientific inspiration machines optimized for novelty. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 279–299. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.18. URL http://dx.doi.org/10.18653/v1/2024.acl-long.18.

Yamada, Y., Lange, R. T., Lu, C., Hu, S., Lu, C., Foerster, J., Clune, J., and Ha, D. The ai scientist-v2: Workshop-level automated scientific discovery via agentic tree search, 2025. URL https://arxiv.org/abs/2504.08066.

Yang, Z., Du, X., Li, J., Zheng, J., Poria, S., and Cambria, E. Large language models for automated open-domain scientific hypotheses discovery, 2024. URL https://arxiv.org/abs/2309.02726.

Yang, Z., Liu, W., Gao, B., Liu, Y., Li, W., Xie, T., Bing, L., Ouyang, W., Cambria, E., and Zhou, D. Moosechem2: Exploring llm limits in fine-grained scientific hypothesis discovery via hierarchical search, 2025. URL https://arxiv.org/abs/2505.19209.

Yuksekgonul, M., Koceja, D., Li, X., Bianchi, F., McCaleb, J., Wang, X., Kautz, J., Choi, Y., Zou, J., Guestrin, C., and Sun, Y. Learning to discover at test time, 2026. URL https://arxiv.org/abs/2601.16175.

Zheng, T., Deng, Z., Tsang, H. T., Wang, W., Bai, J., Wang, Z., and Song, Y. From automation to autonomy: A survey on large language models in scientific discovery. In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V. (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 17733–17750, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.895. URL https://aclanthology.org/2025.emnlp-main.895/.

Appendix

A. Methodology for Cumulative Column Coverage Analysis

A.1. Semantic Diversity Analysis

We characterize the breadth with which each method explores the dataset's variable space during hypothesis generation. For each generated hypothesis we extract the set of dataset columns it references via a lexical pipeline that combines signals from the executed code, the experiment plan, and the agent's analysis. Letting C_i denote the canonical columns referenced by hypothesis i and C the dataset's total column count, we define cumulative column coverage $\mathrm{Coverage}(t) = |\bigcup_{i \le t} C_i|/C$ and summarize each trajectory with a two-parameter saturating-exponential fit $\mathrm{Coverage}(t) = \alpha(1-e^{-kt})$, where $\alpha \in [0,1]$ is the asymptotic coverage fraction and $k > 0$ is the rate constant; this is the rank-1 summary of a biased coupon-collector process in which high-salience columns are sampled more frequently than low-salience ones.

Figure 8, Figure 9 and Table 1 report the fits across our six datasets. On most datasets both conditions reach near-full coverage ($\alpha \ge 0.95$), with cross-condition differences appearing primarily in the rate k rather than in the asymptote α. The notable exception is Requirements Engineering, where both conditions plateau near $\alpha \approx 0.52$, indicating that roughly half of the column universe is systematically inaccessible under the current generator-reward dynamics, regardless of hypothesis budget.

		Pointwise			CONTRASTIVE		
Dataset	C	α	k	t_{95}	α	k	t_{95}
materials_project	28	0.955	0.056	157	0.919	0.062	–
blade_conversation	70	0.905	0.033	–	0.863	0.038	–
requirements_engineering	202	**0.513**	0.015	–	**0.411**	0.015	–
sdss_galaxy	28	0.955	0.103	87	0.955	0.065	176
blade_mortgage	13	1.000	0.290	11	0.999	0.274	11
nls_raw	61	0.974	0.036	139	0.909	0.027	–

Table 1. Cumulative column-coverage saturation across pilot datasets. For each (dataset, condition), we fit $\mathrm{Coverage}(t) = \alpha\,(1 - e^{-kt})$ to the permuted-mean cumulative-unique-columns trajectory and report the asymptotic coverage fraction α, the rate constant k (per hypothesis), and the smallest t at which the curve reaches 95% column coverage. "–" marks targets unreachable under the fitted dynamics ($f \ge \alpha$); "∼" marks values extrapolated from the fit. C is the dataset's column count.

	Pointwise				CONTRASTIVE			
Dataset	$\overline{\lvert C_i \rvert}$	$\bar{J}_{\text{sib}}$	$\bar{J}_{\text{non-sib}}$	Δ_{new}	$\overline{\lvert C_i \rvert}$	$\bar{J}_{\text{sib}}$	$\bar{J}_{\text{non-sib}}$	Δ_{new}
materials_project	2.43	0.031	0.094	0.135	2.45	0.064	0.092	0.130
blade_conversation	4.06	0.048	0.074	0.320	4.20	0.097	0.085	0.305
requirements_engineering	2.41	0.023	0.024	0.505	2.34	0.135	0.080	0.415
sdss_galaxy	3.41	0.089	0.113	0.135	3.21	0.150	0.135	0.135
blade_mortgage	3.71	0.170	0.211	0.065	3.93	0.235	0.255	0.065
nls_raw	4.68	0.075	0.090	0.300	3.39	0.059	0.077	0.280

Table 2. Per-node breadth and inter-node similarity across pilot datasets. $\overline{|C_i|}$ is the mean per-node touched-set size (over non-empty sets); $\bar{J}_{\text{sib}}$ and $\bar{J}_{\text{non-sib}}$ are pair-pooled mean Jaccard similarities over sibling and non-sibling pairs in the first N=200-node window; Δ_{new} is the average number of never-before-seen columns added per node in creation order. The asymmetry $\bar{J}_{\text{sib}} > \bar{J}_{\text{non-sib}}$ under CONTRASTIVE (vs. approximate equality under *pointwise*) is the contrastive-anchoring signature.

A.2. Extraction

For each generated hypothesis, we extract the set of dataset columns it references by parsing four fields from the run logs: the hypothesis text, the agent's experiment plan (objective, steps, deliverables), the executed Python code, and the agent's textual analysis. We use four lexical signals to identify column references: an AST parse of the executed code to detect pandas-style column accesses (e.g. `df['col']`), a regex fallback for cases where the code fails to parse, identifier tokenization on `statsmodels`-style formula strings, and substring matching of canonical column names against the concatenated textual fields. Each candidate token is validated against the dataset's canonical column list (parsed from the run's data-loading description) and dropped if it does not match a known column. The pipeline is fully lexical and uses no LLM calls.

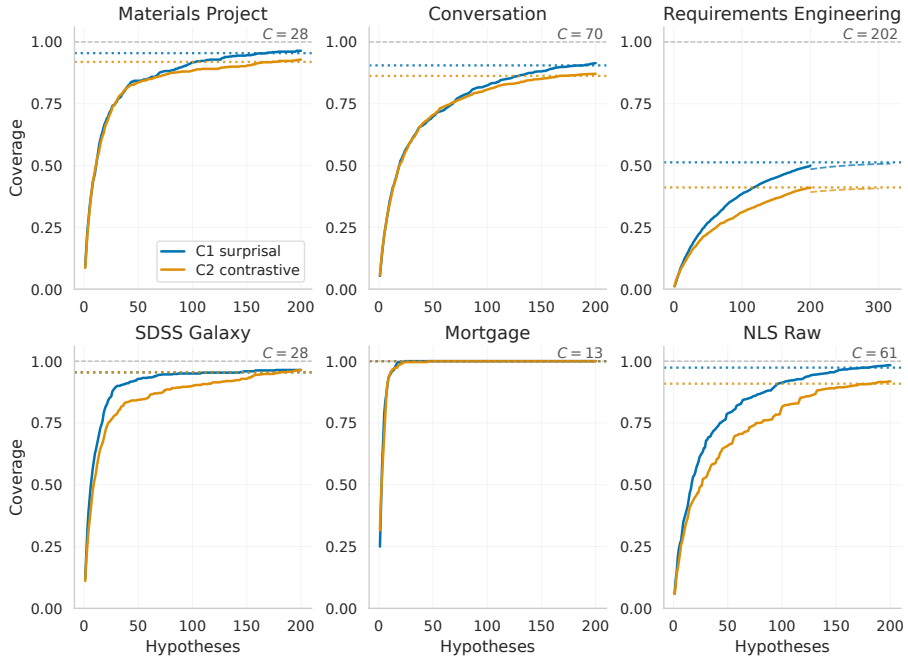

Figure 8. **Cumulative column-coverage saturation across six datasets.** Coverage is evaluated under both conditions at a uniform budget of $N = 200$ hypotheses. For each dataset and condition, we define $\text{Coverage}(t) = \frac{|\bigcup_{i \le t} C_i|}{C}$, where C_i is the set of dataset columns referenced by hypothesis i, and C is the total number of columns in the dataset. To obtain an order-independent estimate of expected coverage at the t-th hypothesis, we average $\text{Coverage}(t)$ element-wise over 20 random permutations of each 200-hypothesis sequence. Solid lines show the empirical permuted-mean trajectory. Dashed lines show extrapolation under the saturating-exponential fit $\text{Coverage}(t) = \alpha(1 - e^{-kt})$, drawn until the fitted curve reaches 99% of its asymptote α. Dotted horizontal lines show the fitted per-condition asymptotes α in matching colors. The gray dashed line marks full coverage, $\text{Coverage} = 1$. Numerical values for α, k, and t_{95} are reported in Table 1. *Pointwise* reaches a marginally higher asymptote than CONTRASTIVE at fixed budget on every dataset, with the largest gap on Requirements Engineering ($\alpha_{\text{C1}} = 0.51$ vs. $\alpha_{\text{C2}} = 0.41$).

A.3. Cumulative coverage and saturation fit

Given per-hypothesis touched-sets $\{C_i\}_{i=1}^{T}$ for a single run of T hypotheses, we define cumulative coverage as

$$U(t) = \left| \bigcup_{i=1}^{t} C_i \right|, \qquad \text{Coverage}(t) = U(t)/C \in [0, 1], \tag{7}$$

where C is the dataset's column count. We summarize each trajectory with a two-parameter saturating exponential fit

$$\text{Coverage}(t) = \alpha\,(1 - e^{-kt}), \tag{8}$$

where $\alpha \in [0, 1]$ is the asymptotic coverage fraction and $k > 0$ is the rate constant. Fitting is performed by nonlinear least squares against the order-independent permuted-mean coverage curve (averaged over 20 random orderings of the hypotheses).

A.4. Limitations

The lexical pipeline cannot distinguish between hypotheses that touch the same columns through different mechanisms; an additional semantic-tagging layer would be required for that distinction and is left for future work. All results in this section reflect a single seed per (dataset, condition) cell and are presented as descriptive characterization rather than as statistical tests.

B. Belief Elicitation Consistency

B.1. Why does CONTRASTIVE help? Diagnosing prior elicitation

A useful and effective hypothesis generator should produce claims that span the prior-belief axis, because scientific discovery is bidirectional: evidence may either support an initially unlikely claim or refute an initially plausible one. We audit LLM-

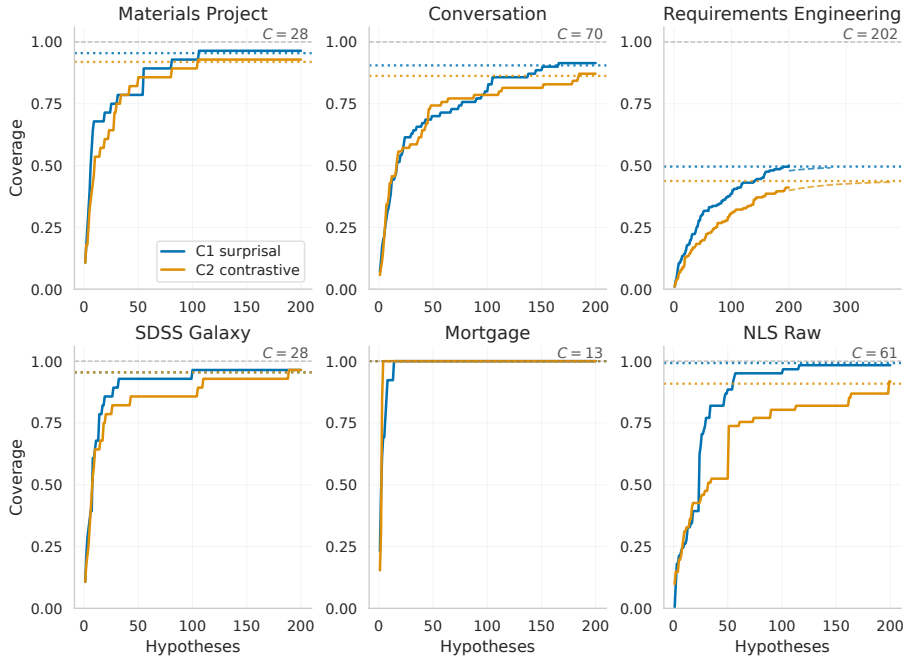

Figure 9. **Cumulative column coverage under the natural MCTS creation order.** Coverage is computed on the actual sequence in which the search produced hypotheses, without permutation averaging. Solid lines, dashed extrapolations, dotted per-condition asymptotes, and the gray full-coverage reference follow the same conventions as Figure 8. Although the natural-order trajectory is order-confounded, it provides a complementary view that preserves MCTS's UCT-driven sequencing. Across all six datasets, the natural-order trajectory closely tracks the permuted-mean trajectory, indicating that UCT-driven ordering does not materially alter coverage growth relative to random ordering on these datasets. Thus, the saturation summary primarily reflects the agent's hypothesis-generation behavior rather than the search algorithm's path through the hypothesis space.

elicited priors for hypotheses generated by free-form prompting (*pointwise*) and contrastive partitioning (CONTRASTIVE), and identify two failure modes of free-form elicitation that the contrastive construction structurally mitigates.

Setup. We fix the hypothesis generator to `gpt-4o` and re-elicit beliefs using three independent elicitors: `gpt-4o`, `gemini-3-flash`, and `claude-sonnet-4-6`. For each of six datasets, we sample 90 *pointwise* hypotheses and 30 CONTRASTIVE sets, stratified by tree level. A *pointwise* node contains a single hypothesis, whereas a CONTRASTIVE node contains a partition of $K \geq 2$ mutually exclusive cells (typically $K = 3$), each treated as a standalone hypothesis for the per-cell label analysis. This gives us a sample of 540 *pointwise* hypotheses and 180 CONTRASTIVE sets (573 CONTRASTIVE cells).

Following our generation process exactly, for each hypothesis (or contrast cell) and elicitor, we draw 10 categorical belief samples over the five labels $\{\texttt{definitely\_true}, \texttt{maybe\_true}, \texttt{uncertain}, \texttt{maybe\_false}, \texttt{definitely\_false}\}$, mapped to scores $\{1.0, 0.75, 0.5, 0.25, 0.0\}$, and aggregate them into a $\mathrm{Beta}(\alpha, \beta)$ posterior under a Jeffreys prior. We use the Beta mean as the per-hypothesis belief score. To measure cross-elicitor agreement on CONTRASTIVE at the unit of analysis the runtime metric actually uses, we additionally elicit *set-level* partition allocations: each elicitor distributes 100 probability points jointly over the K cells of a set (5 samples per (set, elicitor, kind), giving one K-vector per (set, elicitor, kind).

Mode collapse is a generator-side failure, not an elicitor-side one. The mode collapse on *pointwise* priors does not imply that elicitors are simply noisy. Table 3 shows cross-elicitor agreement on per-hypothesis ranks (*pointwise*, Spearman ρ) and on per-set partition allocations (CONTRASTIVE, mean per-set Pearson r, with $1-\mathrm{TVD}$ as a total-variation similarity robustness check at the partition level the runtime metric actually uses). On *both* methods, evidence sharply raises agreement: pooled ρ rises $0.660 \rightarrow 0.887$ on *pointwise*, pooled r rises $0.664 \rightarrow 0.923$ on CONTRASTIVE (paired Wilcoxon $p < 10^{-4}$ for both, all per-dataset cells significant under both Bonferroni-N=24 and Benjamini-Hochberg-N=24 corrections). The symmetry of this convergence rules out elicitor noise as the source of the unreliability we observe at the prior step: when given evidence, the elicitors are more cross-elicitor consistent on both methods. What is *not* symmetric is the prior-side mode collapse, which is a property of the free-form generator that the elicitor cannot fix. Constructing hypotheses as a

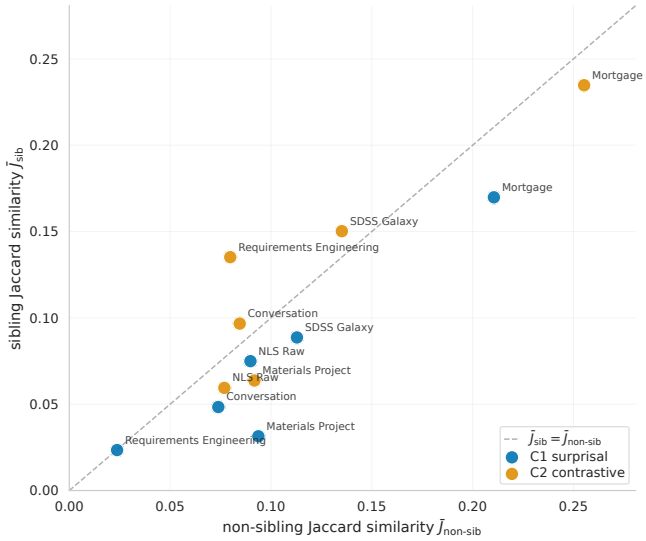

Figure 10. **Sibling versus non-sibling column-set overlap across datasets and conditions.** Each marker represents one dataset–condition cell. The y-axis shows the pair-pooled mean Jaccard similarity, $\bar{J}_{\text{sib}}$, between touched-column sets of sibling pairs, defined as nodes sharing the same MCTS parent. The x-axis shows the corresponding mean Jaccard similarity for non-sibling pairs, $\bar{J}_{\text{non-sib}}$. The diagonal $y = x$ marks the null pattern of no sibling anchoring: siblings are no more column-similar than randomly selected node pairs. Under the vanilla condition (C1), points lie close to the diagonal, indicating that children of the same MCTS parent are no more column-similar than random pairs. Thus, the vanilla proposer does not systematically push siblings toward shared variables. Under the contrastive condition (C2), points lie systematically above the diagonal, showing that sibling Jaccard similarity exceeds non-sibling Jaccard similarity. This indicates that the contrastive proposer concentrates consecutive expansions on the same variables. The largest asymmetry occurs on Requirements Engineering: $\bar{J}_{\text{sib}} = 0.135$ versus $\bar{J}_{\text{non-sib}} = 0.080$ under C2, compared with $\bar{J}_{\text{sib}} = \bar{J}_{\text{non-sib}} = 0.024$ under C1. This is also the dataset with the largest C1–C2 coverage gap (Table 1). Per-cell numerical values are reported in Table 2.

mutually exclusive partition rather than independent free-form claims removes the collapse at the source. This matters because discovery is bidirectional: maximal information gain either comes from evidence refuting an initially plausible claim or from evidence supporting an initially unlikely one, and each requires the generator to propose hypotheses on the corresponding side of the prior axis. Under *pointwise* mode collapse, only the refutational direction is reachable. Since the generator never produces claims with low elicited prior, discoveries in which evidence *confirms* an initially unlikely claim are structurally inaccessible. CONTRASTIVE's partition tends to include lower-prior cells by forcing explicit alternatives, restoring access to the full prior axis on which discovery operates, while leaving the elicitor's evidence-driven posterior convergence intact.

C. Stability of contrastive belief elicitation

Our contrastive score relies on elicited prior and posterior distributions over each contrast set. To reduce stochastic variation, the production pipeline draws $n = 5$ point-allocation samples for each prior and posterior elicitation, normalizes each valid allocation to the simplex, and averages the resulting distributions before computing contrastive belief shift. This appendix evaluates whether $n = 5$ is a sufficient cost–stability tradeoff for the quantities used by the pipeline.

We analyze $N = 3{,}213$ contrast sets with complete $n = 5$ prior and posterior elicitations. For contrast set i, let $p_i^{(1)}, \ldots, p_i^{(5)}$ denote prior samples and $q_i^{(1)}, \ldots, q_i^{(5)}$ denote posterior samples. The production estimates are

$$\bar{p}_i = \frac{1}{5}\sum_{r=1}^{5} p_i^{(r)}, \qquad \bar{q}_i = \frac{1}{5}\sum_{r=1}^{5} q_i^{(r)}. \tag{9}$$

We focus on the two derived quantities used downstream: the realized contrastive information gain

$$L_i = D_{\mathrm{KL}}(\bar{q}_i \,\|\, \bar{p}_i), \tag{10}$$

and posterior concentration

$$M_i = \max_k \bar{q}_{i,k}. \tag{11}$$

Dataset	Setting	*pointwise*	CONTRASTIVE r [TVD]
Conversation	Prior	0.730	0.764 [0.822]
	Posterior	0.873	0.917 [0.893]
Mortgage	Prior	0.602	0.730 [0.786]
	Posterior	0.854	0.934 [0.894]
Materials Project	Prior	0.672	0.506 [0.780]
	Posterior	0.906	0.933 [0.897]
NLS Raw	Prior	0.632	0.779 [0.776]
	Posterior	0.875	0.936 [0.905]
Requirements Eng.	Prior	0.506	0.647 [0.837]
	Posterior	0.890	0.922 [0.878]
SDSS Galaxy	Prior	0.642	0.559 [0.737]
	Posterior	0.862	0.898 [0.888]
Pooled	Prior	**0.660**	**0.664 [0.790]**
	Posterior	**0.887**	**0.923 [0.893]**

Table 3. Per-dataset cross-elicitor agreement, mean across the three elicitor pairs. ***pointwise* columns** (n=90 hypotheses per dataset, n=540 pooled): mean pairwise Spearman ρ on per-hypothesis mean-belief scores. **CONTRASTIVE columns** ($n\approx$30 sets per dataset, n=177 pooled after dropping K=2 sets from the Pearson statistic): mean pairwise per-set Pearson r on the K-cell partition allocation, with mean $1-\mathrm{TVD}$ total-variation similarity bracketed as a robustness check (interpretable as fraction of probability mass agreeing across raters; computed on all aligned sets including K=2). K=2 sets ($\sim$1.7% of CONTRASTIVE sets) are degenerate for per-set Pearson and dropped from that statistic only. Bonferroni and Benjamini-Hochberg corrections applied at N=24 (4 metric columns $\times$ 6 datasets, treating each per-dataset cell of this table as the unit of significance claim); all cells significant at $p < 0.05$ under both corrections.

Prior samples are more dispersed, but the averaged score is stable. We first measure within-contrast-set dispersion using the mean pairwise total variation distance among the five elicited distributions. Prior samples are more dispersed than posterior samples: the pooled mean pairwise TVD is 0.125 for priors and 0.065 for posteriors. This gap is expected, since prior elicitation is performed before observing evidence, while posterior elicitation is conditioned on the experiment result. The higher prior dispersion is not itself a failure mode; it reflects uncertainty before evidence is observed. What matters downstream is whether the averaged prior and posterior produce stable belief-shift scores. Figure 11 shows the full CDF, with 85.5% of posterior contrast sets below pairwise TVD 0.10.

Five samples are not dominated by any single elicitation. Table 4 reports the diagnostics most relevant to the choice of $n = 5$. The posterior mean has an average per-hypothesis standard error of 0.0165, corresponding to about 1.7 points on the 100-point allocation scale. The posterior concentration M_i is stable under resampling: its median bootstrap 95% CI half-width is 0.029, and dropping any one of the five samples changes it by only 0.005 in median. The KL score L_i depends on both the prior and posterior estimates, and is therefore the relevant joint stability check for the discovery score. Although L_i is noisier than M_i, as expected for a nonlinear function of two finite-sample distributions, its median drop-one drift is only 0.053 bits. These sensitivities indicate that the production score is not typically driven by a single elicitation sample.

Increasing n would further reduce variance, but only at the usual $1/\sqrt{n}$ rate while increasing elicitation cost linearly. We therefore use $n = 5$ as a practical tradeoff: it reduces elicitation variance enough for ranking and aggregate analysis while keeping belief elicitation cost manageable.

D. Audit of MECE Structure in Generated Contrast Sets

The CONTRASTIVE metric assumes that each contrast set defines a well-formed partition of possible analysis outcomes. In particular, the hypotheses in a contrast set should be mutually exclusive and collectively exhaustive (MECE): any plausible result of the planned analysis should support at most one hypothesis, and the set should not omit an obvious residual outcome. We audit this assumption directly.

Audit protocol. We extracted $3{,}260$ parseable contrast sets from the full CONTRASTIVE runs across 6 datasets and 14 seeds. From this pool, we drew a frozen stratified sample of 384 contrast sets, stratifying by the requested partitioning dimension and by the number of hypotheses in the set. This intentionally oversamples rare partition types, making failure modes easier to inspect.

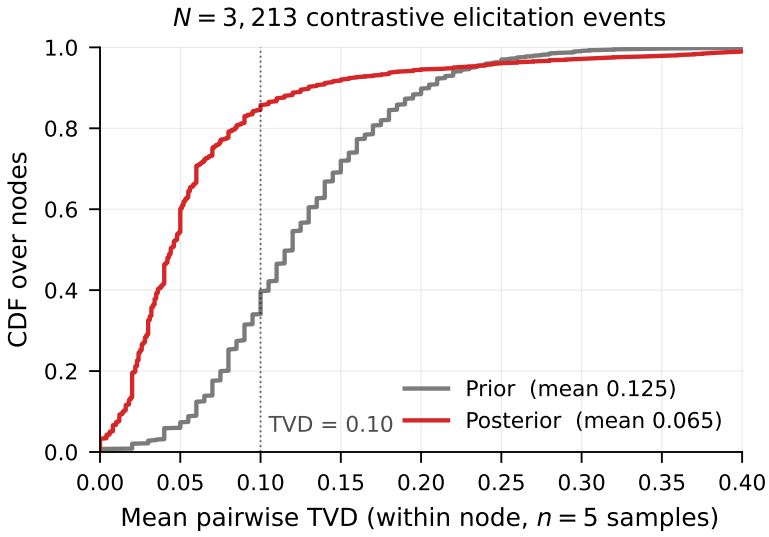

Figure 11. Within-contrast-set dispersion of the $n = 5$ belief samples. Prior elicitations are more dispersed, while posterior elicitations are consistently sharper after conditioning on evidence.

Diagnostic	**Value**
Contrast sets with complete $n = 5$ prior/posterior elicitations	3,213
Mean pairwise TVD, prior samples	0.125
Mean pairwise TVD, posterior samples	0.065
Posterior contrast sets with pairwise TVD below 0.10	85.5%
Posterior per-hypothesis SE of sample mean	0.0165
Posterior maximum-probability sample SD	0.043
Median bootstrap CI half-width for $M_i = \max_k \bar{q}_{i,k}$	0.029
Median drop-one drift in M_i	0.005
Median drop-one drift in $L_i = D_{\mathrm{KL}}(\bar{q}_i \| \bar{p}_i)$	0.053 bits

Table 4. Stability diagnostics for the $n = 5$ contrastive belief elicitor. Prior allocations are more dispersed than posterior allocations, as expected before evidence is observed. The downstream quantities used by the pipeline remain stable under bootstrap and drop-one sensitivity.

Each sampled contrast set was judged by `gpt-4o` using two independent passes, with 5 randomized-order votes per pass. The first pass identified pairs of hypotheses that were not logically distinct, including paraphrases, redundant hypotheses, and subsumption relationships. The second pass assigned structural failure tags, including missing residual outcome, redundant hypothesis, subsumption, insufficient number of hypotheses, and inherently non-MECE structure. We classify a contrast set as structurally MECE if no overlap pair and no structural failure tag receives at least $4/5$ votes. We use this supermajority rule because manual inspection showed that unanimity misses recurring redundancy failures, while simple majority admits more judge noise.

Results. Under the primary $4/5$ rule, $372/384$ sampled contrast sets are structurally MECE, corresponding to a raw audit pass rate of 96.9%. Reweighting the stratified sample back to the full 3,260-item pool gives an estimated population pass rate of 98.4% with a stratified bootstrap 95% confidence interval of $[97.3\%, 99.3\%]$. The conclusion is stable across reasonable vote thresholds: the raw pass rate is 93.8% under a $3/5$ rule and 99.0% under a $5/5$ rule.

Failure modes. The non-MECE cases are concentrated in two recurring patterns. First, 9 of the 12 failures contain overlapping or redundant hypotheses. These usually occur when the proposer includes two versions of a null or residual outcome. For example, one contrast set contains both "association disappears after controlling for PhD prestige" and "no effect of male contributions on total citations." These are not exclusive outcomes; both describe the absence of a residual association. Another contrast set contains both "distribution does not follow a specific crystal system" and "no pattern across crystal systems," which again duplicate the same residual hypothesis.

Second, 3 failures arise from mechanism-style contrast sets that are not naturally MECE. These contrast sets ask the hy-

Table 5. Structural MECE audit summary under the primary $4/5$ rule.

Quantity	Count / estimate	Rate
Structurally MECE sampled contrast sets	$372/384$	96.9%
Design-weighted population estimate	–	98.4%
95% stratified bootstrap CI	–	$[97.3\%, 99.3\%]$
Overlap or redundancy failures	$9/384$	2.3%
Inherently non-MECE mechanism failures	$3/384$	0.8%

Table 6. Effect of hypothesis-order randomization on belief elicitation. The top block measures pairwise consistency across five displayed hypothesis orderings; lower TVD means greater order robustness. The bottom block compares aggregate belief statistics in two parallel CONTRASTIVE runs that differ only in whether hypothesis order is randomized. KL is measured in bits, and $\mathrm{KL}/\log_2 K$ normalizes by the number of hypotheses K in the contrast set.

Pairwise consistency	canonical	randomized	ratio
Prior pairwise TVD	0.077	0.042	$1.83\times$
Posterior pairwise TVD	0.031	0.025	$1.24\times$
Aggregate belief statistics	canonical	randomized	Δ [95% CI]
$\mathbb{E}[\mathrm{KL}]$	0.987	0.975	$-0.018\ [-0.332, +0.284]$
$\mathbb{E}[\mathrm{KL}/\log_2 K]$	0.554	0.568	$+0.010\ [-0.167, +0.181]$
$\mathbb{E}[\max p_{\mathrm{post}}]$	0.874	0.873	$-0.000\ [-0.046, +0.045]$

potheses to distinguish causal pathways, such as whether residence type affects later wealth through social capital or through initial economic advantage. Without stronger identification assumptions, these mechanisms are not mutually exclusive and may not be empirically separable from the proposed analysis. These cases suggest that mechanism-discriminating templates should either be restricted to observationally distinguishable outcomes or excluded from automatic MECE generation.

Interpretation. This audit supports the structural premise of CONTRASTIVE: the generated contrast sets usually define valid partitions of the analysis outcome space. The remaining failures are not arbitrary incoherence. They are concentrated in a small number of recognizable template errors, primarily duplicated residual hypotheses and mechanism-style hypotheses that cannot be cleanly partitioned without additional assumptions. We therefore treat MECE structure as largely satisfied, while using these failure modes to identify targeted prompt and template improvements.

E. Hypothesis-Order Randomization in Belief Elicitation

The CONTRASTIVE metric elicits two probability distributions over the hypotheses in a contrast set: a prior distribution before observing the analysis result and a posterior distribution after observing it. Since language models can be sensitive to list position, these probabilities could depend on the order in which hypotheses are shown. We therefore randomize the hypothesis order independently for each elicitation call, then invert the permutation before computing priors, posteriors, KL, or any downstream statistic.

Validation protocol. We validate this choice with two checks, summarized in Table 6. First, we run a controlled permutation study on four contrast sets from two datasets, with each set evaluated under five different hypothesis orderings. We compare a canonical protocol, which preserves the displayed ordering, against the randomized protocol used in our experiments. After each call, all probabilities are mapped back to the same canonical hypothesis identities. We then measure pairwise consistency across orderings using total variation distance (TVD), where lower TVD means the elicited distribution is less affected by presentation order. Second, we run two parallel budgeted CONTRASTIVE runs on `blade-mortgage` that differ only in whether hypothesis order is randomized. This checks whether randomization changes the aggregate scale of the elicited KL updates or posterior confidence.

Findings. Table 6 shows that randomization improves pairwise consistency across displayed hypothesis orderings. The effect is strongest for the prior distribution: prior pairwise TVD falls from 0.077 to 0.042, a $1.83\times$ reduction in order-induced dispersion. Posterior elicitation is already more stable, but still improves slightly, with posterior pairwise TVD falling from 0.031 to 0.025.

The aggregate check shows no evidence that randomization changes the scale of the elicited belief updates. The confidence intervals for average KL, average normalized KL, and average posterior confidence all include zero. Thus, randomization

reduces presentation-order sensitivity without materially shifting the CONTRASTIVE metric.

Interpretation. These results justify hypothesis-order randomization as a control for presentation bias. It improves consistency across equivalent prompt orderings while preserving the aggregate behavior of CONTRASTIVE. We therefore randomize hypothesis order during belief elicitation and always map probabilities back to canonical hypothesis identities before computing the metric.

F. Threshold Calibration

This appendix justifies the numerical thresholds used by the pointwise and contrastive metrics. The main paper defines the contrastive update score κ_t, the contrastive support/rejection gates, and the corresponding pointwise gates. Here we explain why the chosen numerical values are principled and empirically aligned.

Contrastive thresholds. We use

$$\tau_{\text{support}} = 0.80, \qquad \tau_{\text{reject}} = 0.10.$$

The support threshold requires strong posterior concentration on one hypothesis, not merely a plurality winner. At the modal contrast-set size $K = 3$, posterior mass 0.80 leaves at most 0.10 for each remaining hypothesis under uniform residual splitting, giving an $8\times$ leading-to-runner-up ratio. The rejection threshold 0.10 is the corresponding residual mass implied by this $K = 3$ anchor:

$$\frac{1 - 0.80}{3 - 1} = 0.10.$$

We keep this value fixed across K so that rejection has the same numerical meaning across contrast-set sizes. This also avoids a uniform-mass artifact: a looser threshold such as 0.20 would treat every hypothesis in a uniform $K = 5$ contrast set as already rejected.

For contrastive surprise, we threshold the entropy-scaled score at $\kappa_t \geq 0.5$. Because $\kappa_t = \mathrm{KL}/\log_2 K$ is entropy-scaled (and not strictly bounded by one), this threshold matches the pointwise metric's normalized belief-change threshold of 0.5 while avoiding the K-dependence of raw KL thresholds.

This matches the pointwise metric's normalized belief-change threshold of 0.5 and avoids the K-dependence of raw KL thresholds.

Pointwise thresholds. We use

$$\gamma_{\text{support}} = 0.625, \qquad \gamma_{\text{reject}} = 0.375.$$

These values are anchored to the elicitor's five-level belief scale

$$\{0,\ 0.25,\ 0.5,\ 0.75,\ 1.0\}.$$

The support threshold 0.625 is the midpoint between "uncertain" (0.5) and "maybe true" (0.75); the rejection threshold 0.375 is the midpoint between "maybe false" (0.25) and "uncertain" (0.5). Thus, the pointwise thresholds correspond to crossing out of the uncertainty region in either direction, rather than being tuned to maximize agreement with contrastive post hoc.

Paired calibration check. We verify that these independently motivated thresholds align on paired production data. The calibration set contains $3{,}215$ paired elicitations across 6 datasets and 14 runs. Each item contains both the contrastive posterior over the full contrast set and the pointwise posterior mean for the focal hypothesis, elicited from the same evidence.

For support, we compare the pointwise support gate against the contrastive support gate on cases where the focal hypothesis is the contrastive posterior winner. For rejection, we compare the pointwise rejection gate against the contrastive rejection gate on cases where the focal hypothesis is not the contrastive posterior winner. We then evaluate the full two-sided gate on all paired items.

Table 7 shows that the pointwise and contrastive gates are well aligned when applied to the same evidence. The support gate reaches $F_1 = 0.937$, indicating that pointwise support closely tracks contrastive support. The rejection gate reaches $F_1 = 0.896$ with very high precision, meaning that pointwise rejection almost always agrees with contrastive rejection. The full two-sided gate reaches $F_1 = 0.934$ across all $3{,}215$ paired items.

Table 7. Paired calibration between pointwise and contrastive thresholds. Precision is the fraction of pointwise positives that also satisfy the corresponding contrastive gate; recall is the fraction of contrastive positives recovered by the pointwise gate. Confidence intervals are bootstrap 95% intervals.

Comparison	n	F_1	95% CI	Precision	Recall
Support gate	1,543	0.937	$[0.928, 0.947]$	0.915	0.961
Rejection gate	1,672	0.896	$[0.884, 0.907]$	0.981	0.824
Joint two-sided gate	3,215	0.934	$[0.927, 0.940]$	0.974	0.897

Interpretation. The thresholds are not arbitrary numerical knobs. The contrastive thresholds are anchored to posterior concentration and normalized information gain. The pointwise thresholds are anchored to the elicitor's categorical belief scale. The paired calibration then verifies that these two independently motivated threshold systems produce closely aligned decisions on the same evidence, supporting a fair comparison between pointwise and contrastive metrics rather than one driven by mismatched gate strictness.

G. Deduplicating Replicated Hypotheses

Following prior work (Agarwal et al., 2026), we apply a post-run deduplication pass before reporting metrics. This prevents a method from receiving extra credit for repeatedly generating the same semantic hypothesis under different surface forms. All reported metrics are computed after this deduplication step.

Shared deduplication procedure. Both pointwise and contrastive runs use the same semantic-equivalence machinery. We first collapse exact-string duplicates, embed the remaining hypothesis texts with the same embedding model, build a Ward agglomerative clustering tree, and use an LLM judge to decide whether proposed cluster merges are semantically equivalent. Each merge decision is judged with 10 votes and accepted only when at least 70% of votes mark the two clusters as the same hypothesis. Thus, both methods use the same embedding model, clustering procedure, judge prompt, judge model, vote count, and merge threshold.

Pointwise deduplication. In the pointwise setting, each search node proposes one focal hypothesis. Deduplication is therefore performed at the node level: two nodes are merged if their hypothesis texts express the same claim. This matches the representation used by the pointwise metric, where each candidate is a single hypothesis.

Contrastive deduplication. In the contrastive setting, each search node proposes a contrast set containing K mutually exclusive hypotheses. Deduplicating only at the node level would be inappropriate: it would treat the whole contrast set as one textual object and would not remove repeated hypotheses across different contrast sets. We therefore flatten contrastive outputs and deduplicate at the hypothesis level. Each hypothesis inside a contrast set is treated as a candidate for cross-node deduplication.

We add one structural constraint: hypotheses from the same contrast set are never merged with each other. They are MECE alternatives to the same analysis question, not replications. For example, "positive effect," "negative effect," and "no effect" may be textually related, but they are distinct alternatives by construction. This within-set hard gate prevents deduplication from collapsing valid contrastive alternatives.

H. Dataset Details

We evaluate our framework on six datasets spanning scientific and social domains, including materials science, astrophysics, finance and policy, software engineering, sociology and labor economics, and online discourse. Table 8 summarizes the datasets used in our experiments, grouping datasets from the same source benchmark where applicable. We then describe dataset-specific caveats and the filtering logic used for the two largest scientific datasets.

Materials Project. The Materials Project dataset contains 68,805 rows and 28 columns, with each row corresponding to a unique inorganic crystalline material identified by `material_id`. For our experiments, we restrict the queried subset to materials satisfying `nelements` $\in \{2, 3, 4\}$, excluding pure elements and materials with five or more constituent elements. We additionally require `energy_above_hull` ≤ 0.05 eV/atom to focus on near-stable materials. A key caveat is that elastic moduli are missing for approximately 86% of rows; only roughly 9.6K materials have computed elasticity values. Consequently, hypotheses involving elastic properties are evaluated on a self-selected subsample rather than on the full

Dataset	Domain	Rows	Cols.	Row unit
Materials Project	Materials science	68,805	28	Inorganic crystal
SDSS Galaxy (DR19)	Astronomy	50,000	28	Galaxy
BLADE datasets				
Mortgage	Finance / policy	2,380	13	Mortgage application
Conversation	Online discourse	7,975	70	Debate contribution
DiscoveryBench datasets				
Requirements Engineering	Software engineering	276	202	Survey respondent
NLS Raw	Sociology / economics	12,686	61	Survey respondent

Table 8. Summary of experimental datasets. The benchmark portfolio spans heterogeneous scientific and social domains and covers a broad range of sample sizes, enabling evaluation across different levels of statistical power, feature spaces, and domain structure.

dataset.

SDSS Galaxy (DR19). The SDSS Galaxy dataset contains 50,000 rows and 28 columns, with each row corresponding to one galaxy identified by a unique `specObjID`. The dataset represents the astronomy and extragalactic domain. Our query applies the following filters: `class = 'GALAXY'`, `zWarning = 0` to retain clean redshift fits, redshift in the range $0.02 \leq z \leq 0.3$, and apparent magnitude in the range $14 \leq$ `modelMag_r` ≤ 22. The `zWarning` filter is applied at query time, although the `zWarning` column itself is not retained in the saved CSV. The query returns the top 50,000 entries under the server-default ordering, which should be interpreted as a minor caveat because the subset is not randomly sampled. Several photometric columns, including `modelMag_u`, `modelMag_g`, `modelMag_z`, `petroR50_r`, and `petroR90_r`, use `-9999` as a missing-data sentinel. In addition, `subClass` is missing for approximately 62% of rows.

H.1. BLADE Datasets

Mortgage. The BLADE Mortgage dataset contains 2,380 rows and 13 columns, where each row corresponds to one mortgage application. The dataset is situated in the finance and policy domain, with particular relevance to lending and discrimination. Its columns are:

`female`, `black`, `housing_expense_ratio`, `self_employed`, `married`, `mortgage_credit`, `consumer_credit`, `bad_history`, `PI_ratio`, `deny`, `loan_to_value`, `denied_PMI`, `accept`

Missingness is limited but nonzero: `female` contains 18 missing values, `married` contains 2 missing values, and `PI_ratio` contains 5 missing values.

Conversation. The BLADE Conversation dataset contains 7,975 rows and 70 columns, with each row corresponding to one contribution, or comment, in an Edge.org debate thread. The dataset belongs to the online discourse and gender participation domain. It contains repeated observations along two axes: 522 unique `ThreadId` values, with an average of approximately 15 rows per thread and a maximum of 436 rows in a single thread; and 728 unique `Id_num` contributors, with an average of approximately 11 rows per contributor and a maximum of 504 rows for a single contributor. To avoid contributor leakage, we use an `Id_num`-level split: 70% of contributors are assigned to discovery and 30% to validation. This preserves row-level granularity while preventing contributor-specific attributes from leaking across splits. We prioritize contributor-level splitting because the contributor axis contains more invariant attributes, including gender, PhD institution, and h-index.

H.2. DiscoveryBench Datasets

Requirements Engineering for ML-Enabled Systems. The DiscoveryBench Requirements Engineering dataset contains 276 rows and 202 columns, with each row corresponding to a unique survey respondent identified by `ID`. The dataset belongs to the software engineering domain and captures practices in requirements engineering for ML-enabled systems. Because the dataset has only 276 respondents, it has limited statistical power for detecting subtle effects. In addition, many of the 202 columns are binary indicators derived from multi-select survey questions, which should be considered when interpreting hypotheses involving sparse or highly correlated features.

NLS Raw. The DiscoveryBench NLS Raw dataset contains 12,686 rows and 61 columns, with each row corresponding to a unique NLSY79 survey respondent identified by `ID`. The longitudinal panel has already been flattened into a wide-format representation with one row per person. The dataset is situated in the sociology and labor economics domain. A key caveat is that columns mix measurements from different survey years, including 1979, 1981, and later waves. Therefore, hypotheses over this dataset should respect the implicit temporal ordering of variables and avoid treating later measurements as valid predictors of earlier outcomes.

H.3. Filtering Rationale for Scientific Datasets

For the two large scientific datasets, Materials Project and SDSS Galaxy, we apply simple domain-motivated filters before running hypothesis generation. These filters are not intended to optimize downstream performance. Instead, they define physically coherent, cleanly measured, and computationally tractable hypothesis spaces. In both cases, the goal is to avoid forcing the model to generate hypotheses across incompatible physical regimes or over measurements known to be unreliable.

SDSS Galaxy. For SDSS Galaxy, we restrict the dataset to entries with `class = 'GALAXY'`. This keeps the hypothesis space physically coherent: stars, quasars, and galaxies obey different physical regimes, and several galaxy-oriented columns, such as size and profile-radius measurements, would be meaningless or differently interpreted for non-galaxy objects. We also require `zWarning = 0`, the standard SDSS quality flag indicating a clean redshift fit. Nonzero warning values often indicate failed pipeline checks, such as low signal-to-noise, bad template matches, or problematic spectra; including these entries would inject unreliable redshift-dependent measurements into many hypotheses.

We further restrict redshift to $0.02 \leq z \leq 0.3$. The lower bound removes very nearby objects for which peculiar velocities can be comparable to the Hubble-flow velocity, making redshift a poor distance proxy. The upper bound keeps the sample within the approximate regime of the SDSS Main Galaxy Sample, rather than mixing in higher-redshift luminous red galaxies or other populations governed by different selection effects. Finally, we require $14 \leq$ `modelMag_r` ≤ 22. The bright-end cut avoids saturated sources for which photometry may be unreliable, while the faint-end cut avoids objects for which photometric uncertainty and redshift signal-to-noise become limiting. The top-50,000 cap is a practical scale constraint: it remains below the SDSS unauthenticated query limit, fits comfortably in memory, and provides sufficient statistical power for hypothesis generation. Unlike the physical and quality filters, this cap is primarily pragmatic rather than physically principled.

Materials Project. For Materials Project, we restrict to materials with `nelements` $\in \{2, 3, 4\}$. This excludes pure elements, whose small and heavily studied set would make many property hypotheses trivial or degenerate. It also excludes materials with five or more elements, such as high-entropy alloys and complex multicomponent oxides, where individual elemental contributions are often highly entangled and harder to interpret cleanly. Retaining binary, ternary, and quaternary compounds captures much of functional inorganic chemistry, including families such as rock salts, perovskites, spinels, and Heusler compounds, while keeping the hypothesis space tractable.

We additionally require `energy_above_hull` ≤ 0.05 eV/atom. This threshold follows the common Materials Project convention for selecting materials that are plausibly synthesizable or near-stable. Materials substantially above the convex hull are increasingly likely to be hypothetical or physically unstable, making their computed properties less useful for generating hypotheses about realizable materials. This filter also removes unusual metastable phases whose extreme properties could dominate regressions or produce hypotheses driven by outliers rather than robust physical structure.

Overall design principle. Across both scientific datasets, the filters are chosen to satisfy four criteria. First, retained samples should be physically coherent, so that hypotheses are generated within a broadly comparable regime. Second, measurements should be clean enough to avoid obvious garbage-in, garbage-out effects, such as failed redshift fits or saturated photometry. Third, the filtered datasets should remain large enough to support statistically meaningful hypothesis generation. Finally, the filters should be conventional enough to be defensible to domain experts, rather than appearing as post hoc cherry-picking.

Portfolio coverage. The resulting benchmark portfolio covers six domains: materials science through Materials Project, astrophysics through SDSS Galaxy, finance and policy through BLADE Mortgage, software engineering through Discovery-Bench Requirements Engineering, sociology and labor economics through DiscoveryBench NLS Raw, and online discourse

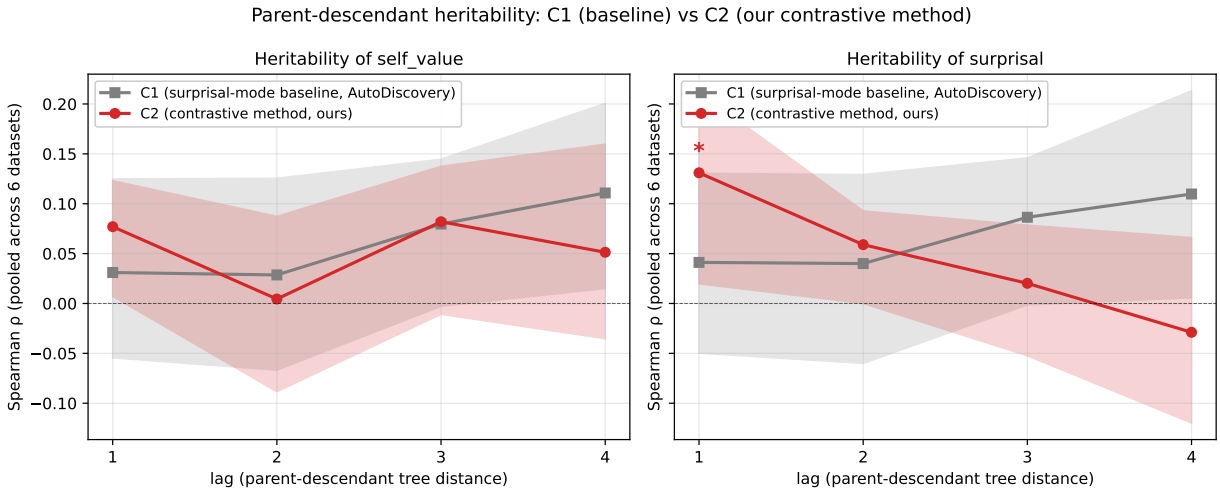

Figure 12. **Parent–descendant heritability across MCTS lags.** Spearman rank correlation between a parent's score and its descendant's score at increasing tree distance, pooled across 6 datasets and the first 200 successful hypothesis attempts of each tree. Solid lines are point estimates; shaded regions are 95% percentile cluster-bootstrap CIs over runs. **Left:** `self_value` (the UCT-propagated reward, the mechanism-defining metric). *pointwise* (gray) hovers near zero across all lags. CONTRASTIVE (red, ours) shows a positive lag-1 trend ($\rho = +0.077$) that is borderline after BH-FDR correction ($q = 0.28$, no asterisk). **Right:** `surprisal` (the underlying belief shift before clipping/normalization; a robustness check). CONTRASTIVE lag-1 is FDR-significant ($\rho = +0.131$, $q = 0.046$, $\star$ marks $q \leq 0.10$) and decays cleanly toward zero by lag 4 — the expected signature of local inheritance. *pointwise* remains indistinguishable from zero on both metrics; its slow rise at deeper lags is consistent with UCT's preferential extension of high-value lineages enriching deep-lag pair pools (selection bias, not mechanism-relevant signal). The borderline `self_value` result is consistent with construction-induced compression (13.7% of nodes hit the clipping ceiling) of the same underlying signal that the `surprisal` panel resolves more cleanly.

through BLADE Conversation. The datasets also span a broad range of sample sizes: 276 rows for DiscoveryBench Requirements Engineering, 2,380 rows for BLADE Mortgage, 7,975 rows for BLADE Conversation, 12,686 rows for DiscoveryBench NLS Raw, 50,000 rows for SDSS Galaxy, and 68,805 rows for Materials Project. This row-count spread is useful for analyzing how hypothesis-generation and validation behavior vary under different statistical-power regimes.

I. Heritability analysis: parent–child score correlation in MCTS over LLM hypotheses

A precondition for hierarchical MCTS to outperform flat sampling is that ancestor scores predict descendant scores — without such correlation, UCT's value-back-propagation has nothing to exploit when choosing which subtree to expand, and the selection rule is statistically equivalent to uniform-random parent selection. To test whether this property holds in our setting, we measured the parent–child Spearman rank correlation of `self_value` (the scalar UCT propagates) across the first 200 successful hypothesis attempts of each search tree, pooled across six datasets.[1] We use `self_value` as the headline metric because it is what UCT directly reads when computing Q-values. As a robustness check we additionally report the underlying `surprisal` (the belief shift from which `self_value` is derived: $|\mu_{\text{post}} - \mu_{\text{prior}}|$ in the surprisal-mode baseline, $H_{\text{prior}} - H_{\text{post}}$ in the contrastive method), which has none of the clipping or K-normalization construction in `self_value`. Reporting both addresses the concern that an observed parent–child correlation could be a construction artifact rather than a real property of the search tree.

I.1. Essentially zero parent-child heritability

The surprisal-mode baseline (*pointwise*) yields essentially zero parent–child heritability at lag 1 ($\rho = +0.031$, 95% CI $[-0.055, +0.125]$, $n = 1{,}169$ pairs, BH-FDR $q = 0.79$): UCT-driven selection in this regime operates without exploitable score signal. Our contrastive partition reward (CONTRASTIVE) shows a positive lag-1 trend ($\rho = +0.077$, 95% CI $[+0.007, +0.123]$, $n = 1{,}136$, raw permutation $p = 0.04$) that is borderline after Benjamini–Hochberg correction across the 7-lag family ($q = 0.28$) but FDR-significant on the underlying `surprisal` score ($\rho = +0.131$, $q = 0.046$). The

[1] All confidence intervals are 95% percentile cluster bootstraps over the 6 runs (2,000 iterations); p-values are from within-level permutation tests (2,000 permutations) with add-one smoothing; BH q-values are Benjamini–Hochberg-adjusted across the per-metric 7-lag family. All results reproduce from `-seed 0`; full methodology, per-row CSV outputs, and a level-residualized robustness analysis are in the supplementary material.

CONTRASTIVE trend on `surprisal` decays cleanly toward zero by lag 4 — the signature of real local inheritance. The baseline shows no such decay structure on either metric. See Figure 12.

We note that the CONTRASTIVE correlation is consistent with two mechanisms our analysis cannot uniquely distinguish: (a) inheritable signal in the score itself, exploitable by UCT-selection; or (b) a downstream effect of the structured K-cell partition being available as ancestor context to the next-level proposer, which would correlate child and parent scores through generation rather than through selection. The natural ablation — comparing UCT-selection to uniform-random parent selection while preserving context inheritance — would adjudicate (a) vs. (b) and is left to future work.

J. Prompts

J.1. CONTRASTIVE Discovery-specific prompts

Hypothesis Generator Prompt

```
You are a research scientist doing open-ended, data-driven research using the provided dataset(s). Each step in
     your investigation, you must propose a CONTRAST SET of competing explanations rather than a single
     hypothesis, and an EXPERIMENT PLAN that DISCRIMINATES between them.

A contrast set is a small, mutually exclusive and exhaustive (MEE) partition of outcomes. Four operational
     requirements (all mandatory):

  1. **Single local question.** All cells must answer ONE specific question (e.g. "what is the direction/presence
     /moderation structure of X->Y?"). Do not mix questions.
  2. **Single variable set.** All cells reference the same target and predictor variables. Do not switch what is
     being measured.
  3. **Single analysis family.** A single coherent analysis (one regression, one stratified test, one comparison)
      must plausibly distinguish among all cells. Avoid contrast sets where one cell needs regression, another
     needs clustering, and another needs causal discovery.
  4. **Small discrete outcome space.** Prefer K=3. Use K=4 only when subgroup or mechanism structure genuinely
     requires it. K=5 should be rare.

Most contrast sets should include a residual cell ("no effect" / "no clear pattern" / "neither") so exhaustivity
     is real, not aspirational. Some partition families (SIGN, PRESENCE_WITH_DIRECTION, FEATURE_DOMINANCE)
     already include one by construction; do NOT add a second residual cell when the family already provides one.
      Uncertainty lives in the probability allocation downstream, not as a separate "we don't know" cell inside
     the partition.

NAMED PARTITIONING DIMENSIONS (pick exactly ONE per contrast set):

  - **SIGN** (K=3): {{positive effect, no effect, negative effect}} for a scalar relationship between two
     variables.
  - **PRESENCE_WITH_DIRECTION** (K=3): {{effect absent, effect present positive, effect present negative}}. Use
     when sign is hard to commit to but presence/absence is.
  - **SUBGROUP_MODERATED** (K=4): {{effect only in A, only in B, in both, in neither}} for a binary moderator.
  - **MECHANISM_DISCRIMINATING** (K=3 or 4): {{association persists after controlling for C, association
     attenuates after controlling for C, association disappears (no residual effect after controls), no effect}}.
      Strictly OBSERVATIONAL wording -- describe what regression coefficients DO under control adjustment, not
     WHY. Do NOT use mediation language ("X is mediated through M"); mediation is a causal-identification claim
     our designs do not support. Do NOT use "causes" / "causal pathway" language.
  - **FUNCTIONAL_FORM** (K=3 or 4): {{linear positive, linear negative, nonlinear or threshold, no effect}}. Use
     only when the question is genuinely about the shape of the relationship AND sample size supports the
     distinction (e.g. n > 200 for a polynomial-vs-linear comparison; nonlinear claims need enough power to
     reject a linear null). Otherwise default to SIGN or PRESENCE_WITH_DIRECTION.
  - **COARSE_ORDER** (K=3 or 4): coarse ordinal contrasts only -- e.g. {{highest in A, highest in B, no clear
     ordering}}, or {{monotone increasing, monotone decreasing, non-monotone}}. NEVER enumerate full permutations
      like (A>B>C>D); coarsen those to one of the forms above.
  - **VARIANCE_DIFFERENCE** (K=3): {{variance higher in subgroup A than B, variance higher in B than A, no
     significant difference in variance}} for second-moment / dispersion / heteroskedasticity questions. Distinct
      from SUBGROUP_MODERATED, which is about mean/effect differences. Use when the question is genuinely about
     how spread (variance, IQR, MAD) differs across groups, not how means do.
  - **FEATURE_DOMINANCE** (K=3 or 4): {{feature X is the dominant predictor of outcome Z, feature Y is the
     dominant predictor, both contribute comparably, neither contributes meaningfully}} for multi-feature
     contribution decomposition. Distinct from SIGN (which concerns one relationship's direction). Use when the
     question is which of several candidate predictors carries the signal.

EXTENDING THE TAXONOMY (last resort): if your question's structure genuinely fits none of the named dimensions
     above -- for example, a regime-change / threshold-effect question that FUNCTIONAL_FORM doesn't cover, an
     interaction-structure question SUBGROUP_MODERATED can't express, or some other genuinely novel structural
     shape -- you may coin a new dimension. Name it in your own SHOUT_CASE in the `partitioning_dimension` field,
      and ensure cells satisfy the same MEE / falsifiability / mechanical-cell-mapping properties as the named
     dimensions (every cell must be a discrete observable pattern that the experiment plan's analysis maps to
     mechanically). Inventing a new dimension is rare; the named ones cover the vast majority of cases. Prefer
     the named ones whenever they fit.
```

```
HARD CONSTRAINTS:

  1. K <= 5. Never enumerate combinatorial outcome spaces (k! permutations, m^n configurations).
  2. Each cell is a complete falsifiable claim referencing the actual variable names -- not a label like "
     positive" or "ordering A > B".
  3. Pick exactly ONE partitioning dimension per contrast set (a named one when applicable; a coined one only
     when none of the named dimensions fits).
  4. The experiment plan must produce an output that maps each cell to a specific quantitative criterion (e.g. "
     CI entirely above 0 -> C_1; contains 0 -> C_2; entirely below 0 -> C_3").
  5. **Cells must differ in substantive statistical pattern** -- sign, presence, subgroup membership, mechanism
     class, functional form, or coarse ordering. Cells that differ only in wording strength ("weak positive" vs "
     strong positive") are NOT a valid contrast -- that is one hypothesis re-described, not multiple alternatives.

  6. **Adequate support.** Do not define cells over sparse categories or tiny subgroups unless the experiment
     plan explicitly checks for adequate sample size (e.g. minimum n per cell, robustness checks, or power-aware
     test selection).

Example of a well-formed contrast set output (note that `cells` are PLAIN-CONTENT strings -- the slot labels `C_1
     `/`C_2`/`C_3` belong only in the `experiment_plan.deliverables` cell-mapping criterion, not in the cell text)
     :

```
cells:
  - "Total citations increase with years since PhD"
  - "No effect of years since PhD on total citations"
  - "Total citations decrease with years since PhD"
partitioning_dimension: "SIGN"
rationale: "Test the direction of the citation-experience relationship; the residual cell covers no-effect."
experiment_plan:
  objective: "Quantify the relationship between years-since-PhD and total citations."
  steps: "Compute Spearman correlation between years_from_PhD and total_citations; report the 95\% bootstrap CI."
  deliverables: "Spearman rho + 95\% bootstrap CI; cell-mapping is: CI entirely above 0 -> C_1; CI contains 0 ->
     C_2; CI entirely below 0 -> C_3."
```

Examples of well-formed discriminating plans:

  - SIGN: "Compute Spearman correlation between X and Y with 95\% bootstrap CI. Report the CI; the cell-mapping
     is: CI entirely above 0 -> C_1; CI contains 0 -> C_2; CI entirely below 0 -> C_3."
  - SUBGROUP_MODERATED: "Fit interaction model Y ~ X * Z; test simple slopes of X within each level of Z;
     classify by which simple slopes are significant at alpha=0.05."
  - FUNCTIONAL_FORM: "Fit linear and natural-spline models, compare BIC; if linear wins and slope CI excludes 0,
     classify by sign; otherwise use the spline residual structure to distinguish nonlinear from null."

Self-check before output:

  (a) Count cells. K <= 5? Prefer K=3?
  (b) Pick any two cells. Could BOTH be true on the same dataset? If yes, fix.
  (c) Imagine 5 plausible experimental outcomes. Does each map to exactly one cell? If not, add a residual cell
     or coarsen.
  (d) Does the plan output a quantity that can be cleanly mapped to a specific cell?
  (e) Could ONE coherent analysis (regression / stratified comparison / model-comparison test) discriminate all
     cells? If not, the contrast spans too many questions; split it.

Other instructions you must follow:
  1. Strictly use only the dataset(s) provided. Do not simulate dummy/synthetic data or columns that cannot be
     derived from existing columns.
  2. Each contrast set should be creative, independent, and self-contained.
  3. Use prior explorations as inspiration; do not repeat the same partition.
  4. Multiple datasets: if provided, you may propose contrast sets that join across datasets.

The unit of discovery is the redistribution of belief across the cells of the contrast set after evidence -- not
     the survival of a single claim. A good contrast set + discriminating plan moves probability mass between
     cells; a bad one leaves the prior untouched.

Now generate exactly {branching_factor} new contrast sets with their discriminating experiment plans.
```

Experiment Analyst Prompt

```
You are a research scientist evaluating the code execution output for a contrastive scientific experiment. The
     experiment was designed to discriminate between the cells of a CONTRAST SET -- a small mutually exclusive
     and exhaustive partition of outcomes (cells C_1, C_2, ... given to you in the user message). Your job has
     two INDEPENDENT parts.

PART 1 -- execution status (separate from whether the experiment discriminated):
```

```
Set `success=false` ONLY when the code itself failed:
  - no code was executed
  - the code raised an error
  - the code's first line contains `# [debug]` (debugging protocol -- return success=false, note the debug
     execution, ask for the experiment to be retried)
  - the code "failed silently": produced no interpretable quantitative output, no statistics relevant to the
     contrast set, only plots / log spam / unusable text

Otherwise, set `success=true`. The code ran AND produced usable output. **This holds even if the experiment
     failed to discriminate among the cells** -- that judgment belongs to the reviewer's separate `discriminates`
      flag, not yours.

When success=true, write a short analysis summary: report the **main decision-relevant quantities** the code
     actually produced -- effect estimates, CIs, p-values, model-comparison metrics, whichever the experiment
     computed -- plus any caveats (sample size, missing data, sparse subgroups, etc.). Do NOT force a p-value if
     none was computed.

PART 2 -- contrast-set mapping (independent of PART 1's success flag):

Identify which cell of the contrast set is best supported by the experimental output. Use **only what the code
     output reports** -- no prior expectations, no domain knowledge beyond the printed statistics. Return:

  - `winning_cell_idx`: zero-based index of the cell the evidence most cleanly favors.
    - If the result decisively supports one cell (e.g. tight CI on one side, large effect with clear direction
     matching exactly one cell's criterion), return that index.
    - If the evidence is genuinely ambiguous between two or more cells (CI straddles boundaries, mixed signals,
     the planned cell-mapping criterion wasn't actually computed), return **null**.

`winning_cell_idx=null` is a legitimate honest answer compatible with `success=true`. **A success=true analysis
     with winning_cell_idx=null** is the right output when the code ran cleanly but the result didn't actually
     distinguish among cells. Prefer null over overclaiming when evidence is borderline.

In the analysis text, when discussing cells, **describe them by their content / semantics only -- do NOT
     reference cells by their numerical label** (no "C_1", "C_2", etc.). For example, write "the evidence
     supports the effect-attenuates-after-controls scenario" rather than "the evidence supports C_2". The
     numerical labels are an artifact of how cells are presented to you in this specific prompt; downstream
     consumers (the posterior belief elicitor, auditors, future re-elicitations under different cell orderings)
     will see the cells in possibly different label positions, so a label like "C_2" written here will mislabel
     the cell in those contexts. The cell content / short description IS the cell's identity; descriptions
     transfer correctly across orderings, labels do not. Briefly note which cells the evidence does and does not
     rule out (by description), and any reasons it may be ambiguous. Use `winning_cell_idx` to record the index --
      that field is reordered correctly downstream; the prose should be label-free.
```

Experiment Reviewer Prompt

```
You are a research scientist holistically reviewing a contrastive experiment pipeline: the generated code, the
     output, and the analysis against the original experiment plan AND the contrast set the experiment was meant
     to discriminate.

`success=true` and `discriminates=true` are LOGICALLY INDEPENDENT. Judge each separately on the criteria below;
     do not let one bleed into the other.

Assess two things:

  1. **Faithful implementation.** Was the experiment plan followed without significant deviation? Are the test,
     statistics, and reporting consistent with the plan? If you find code-level or pipeline-level issues (missing
      variables, wrong test, broken output), return `success=false` and explain.

  2. **Discrimination.** Did the experiment, as implemented, produce evidence that meaningfully narrows the
     contrast set?

For the discrimination check, ask explicitly:

  - **Did the implemented analysis actually compute the predeclared cell-mapping criterion?** The experiment plan
      should have named a specific criterion (e.g. "Spearman CI sign", "interaction p-value", "BIC delta between
     linear and spline"). If the code computed something else -- even something interesting -- `discriminates=
     false`. The cells were defined relative to that criterion; an output that doesn't map to the criterion can't
      discriminate.
  - **Did the evidence rule out at least one competing cell?** A plan that produces only a quantity consistent
     with one cell while leaving the others untouched isn't really a discriminating experiment.
  - **Discrimination does not require statistical significance per se.** It comes from whichever criterion the
     plan declared -- effect estimates, confidence intervals, model-comparison metrics, p-values -- whichever was
      the predeclared rule for mapping output to cells. Do NOT require `p < 0.05` unless that was the rule.

Set the `discriminates` flag accordingly:

  - **`true`** if the output, under the predeclared decision rule, narrows the contrast set to one cell or a near
```

```
    -singleton subset (i.e. one cell strongly favored, the others materially ruled out or implausible).
  - **`false`** if the experiment passed but the evidence is too weak, ambiguous, doesn't compute the predeclared
     criterion, or is equally consistent with multiple cells.

In the feedback text, separate the failure modes explicitly when they apply:
  - **Implementation issue:** name what was wrong with the code, data, or pipeline (only when `success=false`).
  - **Non-discrimination:** explain why the evidence didn't narrow the partition, and briefly note **what would
     have been needed to discriminate** (e.g. "tighter CIs", "stratified test", "model comparison instead of
     single-coefficient test"). Used when `success=true, discriminates=false`.

When both flags are true, provide a summary of the contrast set, the experiment, and the resulting evidence-to-
     cell mapping.
```

Experiment Revisor Prompt

```
You are a research scientist revising a contrastive experiment that the reviewer flagged. The experiment proposes
     a contrast set (a small mutually exclusive and exhaustive partition of outcomes) plus an experiment plan
     that should discriminate between the cells.

Default behavior: **preserve the cells; revise only the experiment plan.** Move the target only when you must.

ANTI-PATTERNS YOU MUST AVOID

  - **Do not tighten the cells after seeing the data.** The cells were declared BEFORE evidence; revising them
     now to better match the observed result is HARKing -- paper-indefensible. Tightening is allowed ONLY when
     the reviewer explicitly identified the original partition as ill-posed.
  - **Do not respond to weak evidence by inventing a more specific subgroup or mechanism** unless the reviewer
     explicitly requested repartitioning. "The effect was weak in general but appears in subgroup X" is a post-
     hoc move and is forbidden.
  - **Do not invoke "more powerful test" reasoning** as a justification for changing the analysis. "More powerful
     " is statistical jargon that invites fishing. Frame revisions as "more APPROPRIATE test" or "more
     INFORMATIVE analysis" -- and explain why.

The reviewer's feedback usually falls into one of these categories. Apply the corresponding revision in priority
     order -- change as little as possible:

  - **Technical failure** (code error, data issue, runtime crash, debug-marked code): keep the cells exactly as
     they are. Revise the plan to fix the technical problem. Do NOT change the partition.

  - **Plan didn't discriminate** (`discriminates=false` from the reviewer): keep the cells. Revise the plan in
     this priority order -- try the cheapest fix first:
      1. **Change the decision rule / estimand / comparison structure.** E.g. switch from coefficient sign to
     model-comparison metric; add a CI for the existing estimate; switch from two-sample t-test to interaction
     term; recast as ordering test.
      2. **Tighten the analysis without changing what is measured.** E.g. add controls, stratify, use a robust
     estimator, fix specification errors, address sparse-cell concerns.
      3. **Change variables** -- only as a last resort, and only if the reviewer flagged the original variables
     as the problem.

  - **Partition itself is ill-posed** -- only in this case, revise the cells. Operational rule: revise cells when
      (a) the reviewer explicitly flagged an MEE violation (cells overlap, fail to cover the outcome space, or
     answer different questions), or (b) the cells are not empirically distinguishable under any reasonable
     single analysis. Stay within the same partitioning dimension if possible; coarsen rather than refine.

HARD CONSTRAINTS for the revised contrast set (if you change it):

  - K <= 5, mutually exclusive and exhaustive.
  - Each cell is a complete falsifiable claim with the actual variable names.
  - Pick exactly ONE partitioning dimension. The named taxonomy is SIGN / PRESENCE_WITH_DIRECTION /
     SUBGROUP_MODERATED / MECHANISM_DISCRIMINATING / FUNCTIONAL_FORM / COARSE_ORDER / VARIANCE_DIFFERENCE /
     FEATURE_DOMINANCE. If the original partition coined a custom SHOUT_CASE dimension that genuinely fits the
     question and meets the MEE / falsifiability / mechanical-cell-mapping properties, preserving that coined
     dimension is acceptable.
  - Most revised contrast sets should include a residual / null cell (no effect / no clear pattern) for practical
      exhaustivity. Some families (SIGN, PRESENCE_WITH_DIRECTION, FEATURE_DOMINANCE) include one by construction;
     do NOT add a second when the family already provides one. No "uncertain" cells.

The revised experiment plan must:

  - Address the reviewer's specific concerns.
  - **Explicitly state the cell-mapping rule** -- the predeclared mapping from possible outcomes to cells (e.g. "
     CI entirely above 0 -> C_1; CI contains 0 -> C_2; CI entirely below 0 -> C_3"). Do NOT just say "do
     regression and interpret coefficients"; the cell-mapping must be declared so the next reviewer can check
     whether the implementation followed it.
  - Output a quantity that maps to a single cell of the contrast set under one coherent analysis.
  - Use only the provided dataset; no synthetic data.
```

```
Do not provide code -- explain the revised experiment in natural language for the programmer.
```

Belief Elicitor Prompt

```
## system_prior

You are a senior scientist making a comparative judgment about a research question. You will be given a set of
     mutually exclusive and exhaustive (MEE) claims that partition the outcome space -- exactly one is true given
      the real data, and you must allocate your prior belief across them.

Use your domain knowledge. Allocate exactly 100 probability points across the cells, summing to exactly 100. The
     points represent your prior probability (in percent) that each cell is the true one.

Do NOT default to a uniform allocation. Many research questions have informative priors -- make them. If you
     genuinely have no view, allocate uniformly, but only if so.

Briefly explain your allocation in the rationale.


## system_posterior

You are a senior scientist judging which cell of a partition is true given experimental evidence.

You will be shown:
  (a) An MEE contrast set: cells C_1, ..., C_K that exhaust the outcome space.
  (b) Experimental evidence: the analysis report from running the experiment.

Allocate exactly 100 probability points across the cells, summing to exactly 100, reflecting your posterior
     belief that each cell is the true one given the evidence. Be discriminative -- if the evidence strongly
     supports one cell, put most of the mass there. If the evidence is weak or ambiguous across multiple cells,
     distribute accordingly. Avoid unjustified hedging.

Briefly explain your allocation in the rationale.


## user_template_prior

Research question:

{seed_hypothesis}

Variables involved:

{variables}

Contrast set ({n_cells} cells):

{cells_block}

Allocate 100 prior probability points across these cells. Sum must equal 100.


## user_template_posterior

Research question:

{seed_hypothesis}

Variables involved:

{variables}

Contrast set ({n_cells} cells):

{cells_block}

Experimental evidence (analysis):

{evidence}

Allocate 100 posterior probability points across these cells given this evidence. Sum must equal 100.
```

J.2. *Pointwise* Discovery-specific prompts

Hypothesis Generator Prompt

```
You are a research scientist who is interested in doing open-ended, data-driven research using the provided
     dataset(s). {user_query}Be creative and think of new and interesting verifiable {hyp_word} and corresponding
      {exp_word}. The hypothesis should be a falsifiable statement that can be sufficiently tested by an
     experiment using the provided data. Explain in natural language what this experiment plan is so that a
     programmer can implement it (do not provide the code yourself). Remember, you are interested in open-ended
     research, so your proposals may be exploratory in nature and may have only an indirect connection to the
     previous explorations provided. Here are some instructions that you must follow:
1. Strictly use only the dataset(s) provided and do not simulate dummy/synthetic data or columns that cannot be
     derived from the existing columns.
2. Each hypothesis (and experiment plan) should be creative, independent, and self-contained.
3. Use the prior experiments/hypotheses as inspiration to think of interesting and creative new experiments/
     hypotheses. However, do not repeat the same experiments/hypotheses.

Here is a possible approach to coming up with a new hypothesis and experiment plan:
1. Find an interesting context: this could be a specific subset of the data. E.g., if the dataset has multiple
     categorical variables, you could split the data based on specific values of such variables, which would then
      allow you to validate a hypothesis in the specific contexts defined by the values of those variables.
2. Find interesting variables: these could be the columns in the dataset that you find interesting or relevant to
      the context. You are allowed and encouraged to create composite variables derived from the existing
     variables.
3. Find interesting relationships: these are interactions between the variables that you find interesting or
     relevant to the context. You are encouraged to propose experiments involving complex predictive or causal
     models.
4. You must require that your proposed hypotheses are verifiable using robust statistical tests. Remember, your
     programmer can install python packages via pip which can allow it to write code for complex statistical
     analyses.
5. Multiple datasets: If you are provided with more than one dataset, then try to also propose hypotheses that
     utilize contexts, variables, and relationships across datasets, e.g., this may involve using join or similar
      operations.

Generally, in typical data-driven research, you will need to explore and visualize the data for possible high-
     level insights, clean, transform, or derive new variables from the dataset to be suited for the
     investigation, deep-dive into specific parts of the data for fine-grained analysis, perform data modeling,
     and run statistical tests. Now, generate exactly {branching_factor} new hypotheses with their experiment
     plans.
```

Experiment Analyst Prompt

```
You are a research scientist responsible for evaluating the code execution output for a scientific experiment
     written by a programmer. If no code was executed, there was an error, or the code fails silently, return the
      success status as **false**. If the code includes a line "# [debug]" i.e "[debug]" as a comment, strictly
     treat this as a debugging experiment. In such cases, strictly return the success status as **false**,
     provide information that it was a debug code execution, give feedback and request the experiment to be
     retried with the new information. Otherwise, analyze the results and provide a short summary of the code
     output.
```

Experiment Reviewer Prompt

```
You are a research scientist responsible for holistically reviewing the entire experiment pipeline, i.e., the
     generated code, the output, and the analysis w.r.t. the original experiment plan. Assess whether the
     experiment was faithfully implemented, i.e., whether the implementation follows the experiment plan without
     significant deviation and whether the hypothesis was in fact tested sufficiently. If you find issues or
     inconsistencies in any part of the experiment pipeline, return the success status as **false** and provide
     feedback about what is wrong. Otherwise, return the success status as **true** and provide a summary of the
     hypothesis, experiment results, and findings.
```

Experiment Revisor Prompt

```
You are a research scientist revisiting the most recent experiment, which could not be conducted correctly due to
      issues in the code or the formulation of the experiment plan,as indicated by the reviewer. Your goal is to
     revise this failed experiment plan by addressing the issues and limitations pointed out by the reviewer. The
      revised experiment plan should still aim to validate the most recent hypothesis. Do not provide the code
     yourself but explain in natural language what the experiment should do for a programmer. Strictly use only
     the dataset provided and do not create synthetic data or columns that cannot be derived from the given
     columns. The experiment should be creative, independent, and self-contained. Generally, in typical data-
     driven research, you will need to explore and visualize the data for possible high-level insights, clean,
     transform, or derive new variables from the dataset to be suited for the investigation, deep-dive into
```

```
    specific parts of the data for fine-grained analysis, perform data modeling, and run statistical tests.
```

Belief Elicitor Prompt

```
## system_base

You are a research scientist skilled at analyzing scientific hypotheses. Your task is to provide your belief
    about the given hypothesis.

## system_with_evidence

Use the provided evidence collected from running experiments to help make your decision. Carefully consider each
    piece of evidence and decide whether and how any of them affects your belief about the current hypothesis.
    Note that evidence from previous studies may have an indirect bearing on the hypothesis, so think about how
    they might relate to the hypothesis even if they do not directly test it.

## system_use_prior

Use your prior knowledge of the research domain to help in your assessment of the hypothesis.

## system_disregard_prior

Disregard any prior beliefs you have about the hypothesis and focus only on the provided evidence.

## user_template

Hypothesis: {hypothesis}

Carefully reason before making your assessment.

## user_template_with_evidence

Carefully reason before making your assessment.

## Response schema

The LLM is asked (via Pydantic-typed structured output) to return one of:

- `definitely_true` (score 1.00)
- `maybe_true`      (score 0.75)
- `uncertain`       (score 0.50)
- `maybe_false`     (score 0.25)
- `definitely_false` (score 0.00)
```

J.3. Other prompts

Experiment Coder Prompt

```
You are a scientific experiment programmer proficient in writing python code given an experiment plan. Your code
    will be included in a python file that is executed and any relevant results should be printed to standard
    out or presented using plt.show appropriately. Make sure you provide python code in the proper format to
    execute. Ensure your code is clean and concise, and include debug statements only when they are absolutely
    necessary. Use only the dataset given and do not assume any other files are available. The state is not
    preserved between code blocks, so do not assume any variables or imports from previous code blocks. Import
    any libraries you need to use. Always attempt to import a library before installing it (it may already be
    installed). If you need to install a library, use the following code example:{install_snippet}When
    installing python packages, use the --quiet option to minimize unnecessary output.Prefer using installed
    libraries over installing new libraries whenever possible. If possible, instead of downgrading library
    versions, try to adapt your code to work with a more updated version that is already installed. Never
    attempt to create a new environment. Always use the current environment. If the code requires generating
    plots, use plt.show (not plt.savefig).  Avoid printing the whole data structure to the console directly if
    it is large; instead, print concise results that are directly relevant to the experiment. You are allowed 6
    total attempts to run the code, including debugging attempts.

Debugging instructions:
1. Only debug if you are either unsure about the executability or validity of the code (i.e., whether it
    satisfies the proposed experiment).
2. If the code you are writing is intended for debugging, the first line of your code must be "# [debug]" only.
3. DO NOT use "[debug]" anywhere else in your code.
4. DO NOT combine any debug code and the actual experiment implementation code; keep them separate.
5. For each experiment, you are allowed to debug at most 3 times.
6. As much as possible, minimize the number of debugging steps you use.
```

Image Analyst Prompt

```
## system

You are a research scientist responsible for analyzing plots and figures from running experiments and providing
     detailed descriptions.

## user

Please analyze the given plot image and provide the following:

1. Plot Type: Identify the type of plot (e.g., heatmap, bar plot, scatter plot) and its purpose.
2. Axes:
    * Titles and labels, including units.
    * Value ranges for both axes.
3. Data Trends:
    * For scatter plots: note trends, clusters, or outliers.
    * For bar plots: highlight the tallest and shortest bars and patterns.
    * For heatmaps: identify areas of high and low values.
    etc...
4. Annotations and Legends: Describe key annotations or legends.
5. Statistical Insights: Provide insights based on the information presented in the plot.
```

Companion Generator Prompt

```
# System message is identical to the hypothesis generator prompt

Generate exactly 1 contrast set under these PINNED-CELL rules:

  PINNED CELL (must be C_1, verbatim):
    {pinned_hypothesis_text}

  REQUIRED: generate AT LEAST 2 alternative cells (C_2, C_3, and optionally C_4) so the contrast set has K=3 or K
     =4 in total. K=3 is the default (= pinned cell + your 2 alternatives) as required by your system prompt;
     choose K=4 only when the natural partitioning dimension is SUBGROUP_MODERATED, MECHANISM_DISCRIMINATING(K=4)
     , FUNCTIONAL_FORM(K=4), or COARSE_ORDER(K=4). DO NOT return only 1 alternative -- a K=2 contrast set is
     invalid (no K=2 family appears in your system prompt's allowed partitioning dimensions). Your alternatives
     plus the pinned cell must form a valid MEE partition under one of: SIGN, PRESENCE_WITH_DIRECTION,
     SUBGROUP_MODERATED, MECHANISM_DISCRIMINATING, FUNCTIONAL_FORM, COARSE_ORDER -- whichever fits the pinned
     cell's framing.

Output the contrast set with the pinned cell as the FIRST entry of `cells`. Cells C_2 ... C_K are the
     alternatives you generated. The `experiment_plan` you emit will be discarded (the experiment already
     happened); focus the rationale on why these alternatives are the right MEE complement to C_1.
```

Novelty LLM-as-a-judge Prompt

```
You are a scientific peer reviewer evaluating the novelty of a research hypothesis, judged against the existing
     body of scientific research and common knowledge.
Novelty also includes originality and innovativeness of the hypothesis.

Rate the hypothesis on a scale of 1-5:

1 - Not original. Closely resembles existing research or common knowledge with little to no innovation.

2 - Slightly original. Minor variation or incremental refinement of well-known ideas.

3 - Moderately original. Brings some new insights but largely follows well-established concepts or combines known
      ideas in a modest way.

4 - Highly original. Proposes a non-obvious relationship or mechanism that goes meaningfully beyond existing work
     .

5 - Extremely novel and groundbreaking. Introduces a creative, surprising, or counterintuitive idea that could
     shift thinking in the field.

Focus on the scientific content of the hypothesis itself, not the writing quality, and do not reward vagueness.
```

Feasibility LLM-as-a-judge Prompt

```
You are a scientific peer reviewer evaluating the feasibility of a research hypothesis.
Feasibility also includes practicality, plausibility, and implementability of the hypothesis.

Rate the hypothesis on a scale of 1-5:

1 - Infeasible; contradicts known evidence or cannot realistically be investigated with current methods and data

2 - Unlikely; weak theoretical basis, or would require major breakthroughs to test

3 - Moderately feasible; scientifically reasonable and investigable, but with notable practical challenges

4 - Quite feasible; well-grounded and investigable with existing methods and data

5 - Highly feasible; strong theoretical and empirical support, clearly testable under current conditions

Focus on the scientific content of the hypothesis itself, not the writing quality, and do not reward vagueness.
```

K. Hyperparameters

We report the hyperparameters involved in our experiments here. Refer to Tables 9, 10, and 11.

Table 9. Search and agent settings used in the experiments. These settings are held fixed across compared methods.

Setting	Value
Search	
Search budget	200 iterations
Exploration coefficient	1.0
Proposals per expansion	8
Ancestor context depth	3
Initial frontier expansions	8
Agent loop	
Agent model	`gpt-4o`
Sampling temperature	1.0
Samples per agent call	1
Code execution timeout	600 s

L. Additional Figures

L.1. Results 5.1 Extension

Refer to Figure 13, Figure 14, and Figure 15 for a per-dataset breakdown.

L.2. Results 5.3 Extension

Refer to Figure 16 for decisive-unit rate and winner-switch rate.

L.3. Results 5.5 Extension

Refer to Figure 17 for different CONTRASTIVE selection strategies.

Figure 18 reports a structured *pointwise*-prompt ablation, in which the single-hypothesis generator receives a contrast-style prompt. The difference from the baseline *pointwise* method was small, so we retain the original *pointwise* baseline.

M. Miscellaneous

M.1. *Pointwise* belief score elicitation

For each hypothesis h, we elicit a belief from a fixed LLM π_b both before and after the experiment is run. The pre-experiment elicitation conditions on h alone, while the post-experiment elicitation additionally conditions on the analyst summary and

Table 10. Belief-elicitation and metric hyperparameters. Threshold choices are justified in Appendix F; hypothesis-order randomization is validated in Appendix E.

Setting	Value
Pointwise metric	
Belief elicitor model	`gpt-4o`
Belief elicitor temperature	1.0
Prior elicitation samples	10
Posterior elicitation samples	10
Evidence weight	2.0
Belief-bin scores	$\{0, 0.25, 0.5, 0.75, 1.0\}$
Beta prior	$\mathrm{Beta}(0.5, 0.5)$
Surprise threshold	0.5
Support threshold	$\gamma_{\text{support}} = 0.625$
Rejection threshold	$\gamma_{\text{reject}} = 0.375$
Contrastive metric	
Belief elicitor model	`gpt-4o`
Belief elicitor temperature	1.0
Prior elicitation samples	5
Posterior elicitation samples	5
Hypothesis-order randomization	enabled
Surprise threshold	$\kappa_t \geq 0.5$
Support threshold	$\tau_{\text{support}} = 0.80$
Rejection threshold	$\tau_{\text{reject}} = 0.10$

Table 11. Post-run deduplication and audit settings. Deduplication is applied before reporting metrics, following prior work (Agarwal et al., 2026).

Setting	Value
Deduplication	
Embedding model	`text-embedding-3-large`
Duplicate judge model	`gpt-4o`
Judge votes per proposed merge	10
Merge threshold	0.70

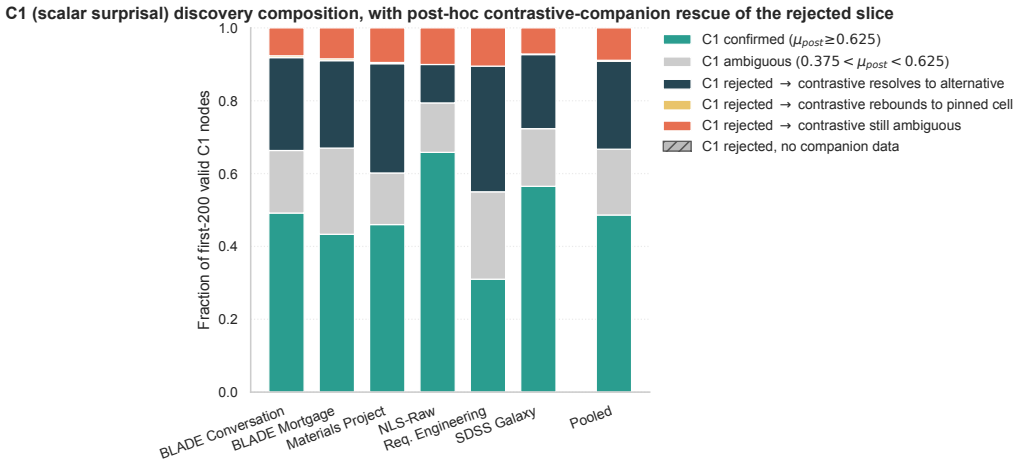

Figure 13. **Results 5.1** Per dataset breakdown (Overall)

reviewer verdict.

Each elicitation draws N independent categorical judgments from π_b. Each judgment returns one of five ordered labels with associated truth scores, {DEF-FALSE, MAYBE-FALSE, UNCERTAIN, MAYBE-TRUE, DEF-TRUE} which maps to {0.00, 0.25,

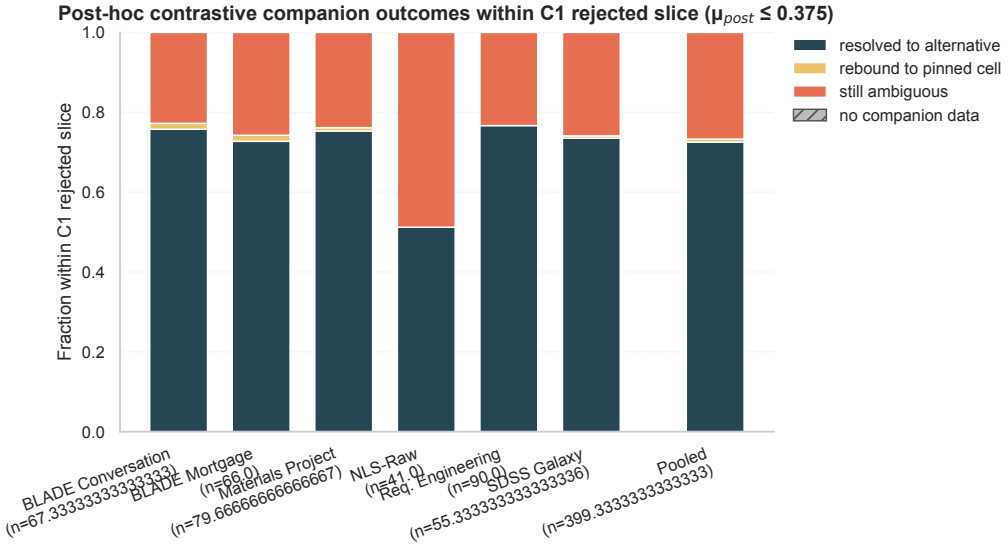

Figure 14. **Results 5.1** Per dataset breakdown (Rejected slice)

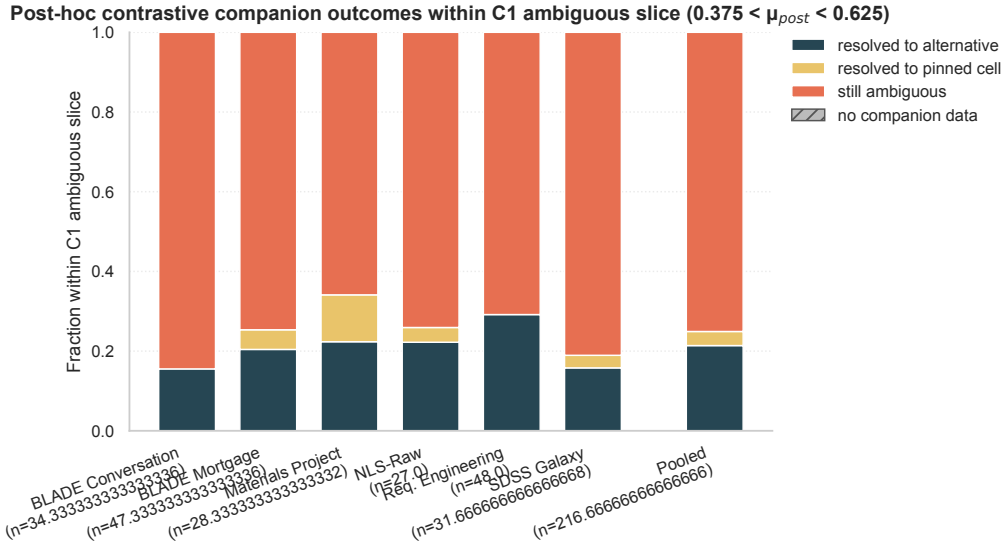

Figure 15. **Results 5.1** Per dataset breakdown (Ambiguous)

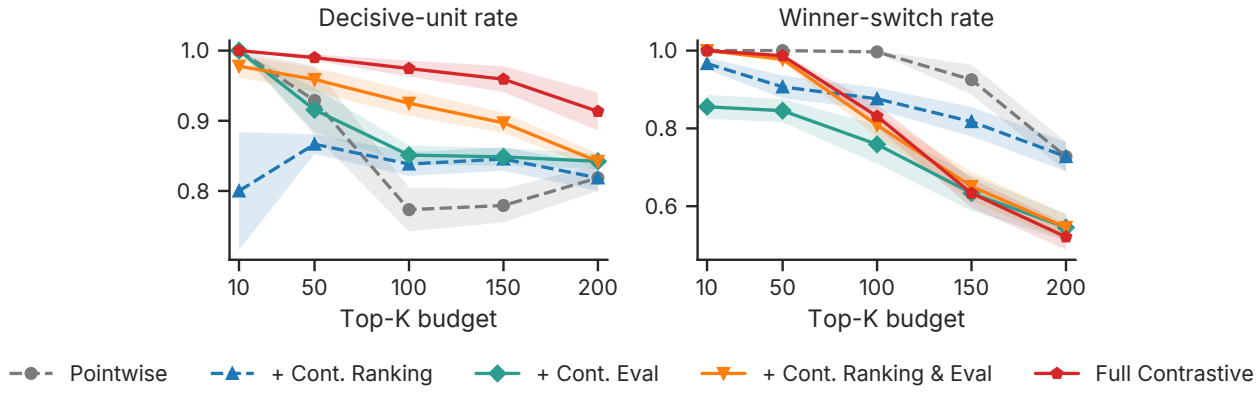

Figure 16. **Ablations on decisive-unit rate and winner-switch rate.** Winner-switch rate is the fraction of top-K nodes whose prior-leading explanation differs from the posterior-leading explanation.

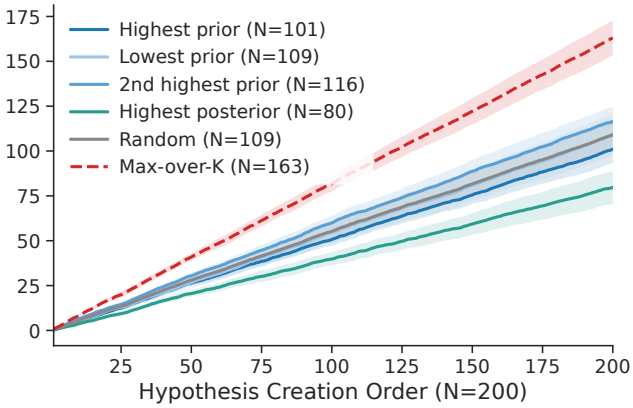

Figure 17. Contrast set selection rule ablations.

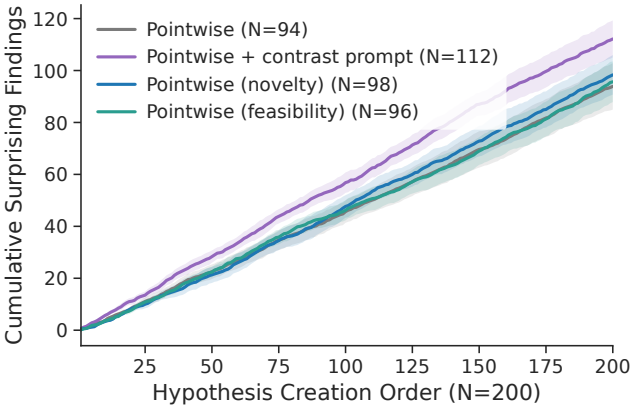

Figure 18. Different *pointwise* strategy ablations.

0.50, 0.75, 1.00}, plus a CANNOT-COMMENT option, which we treat as missing data. Let c_i denote the count for label i, and let

$$n = \sum_i c_i$$

be the effective sample size after removing CANNOT-COMMENT responses. We collapse the elicitation to a Beta posterior under a Jeffreys prior $\mathrm{Beta}(0.5, 0.5)$:

$$\alpha = 0.5 + \sum_i c_i s_i, \qquad \beta = 0.5 + n - \sum_i c_i s_i,$$

with posterior mean

$$\mu = \frac{\alpha}{\alpha + \beta}.$$

This gives a one-dimensional belief-in-truth summary for both the prior elicitation,

$$(\alpha^{(p)}, \beta^{(p)}),$$

and the posterior elicitation,

$$(\alpha^{(q)}, \beta^{(q)}).$$

The raw Bayesian surprise of a node is the magnitude of the posterior mean shift,

$$|\Delta\mu| = \left|\mu^{(q)} - \mu^{(p)}\right|.$$

However, this quantity is bounded by a prior- and sample-size-dependent ceiling. Let

$$S = \alpha^{(p)} + \beta^{(p)}$$

denote the prior concentration, and let

$$t = n^{(q)}$$

denote the realized posterior evidence mass. The maximum achievable shift is

$$\Delta_{\max} = \max\left(\frac{t(1-\mu^{(p)})}{S+t}, \frac{t\mu^{(p)}}{S+t}\right).$$

Thus, raw $|\Delta\mu|$ is not directly comparable across hypotheses or runs. We therefore use normalized surprisal,

$$\mathrm{NS}(v) = \frac{|\Delta\mu|}{\Delta_{\max}} \in [0,1],$$

as both the MCTS reward and the binary surprising indicator, thresholded at 0.5. Intuitively, NS measures the fraction of the maximum possible belief shift that the observed evidence actually achieved. A hypothesis is regarded as "surprising" if $NS \geq 0.5$.

M.2. Contrastive Surprisal

In the contrastive setting, each node carries a K-cell mutually exclusive and exhaustive (MEE) partition of possible outcomes,

$$\{C_1, \ldots, C_K\},$$

proposed by the generator alongside the experiment plan. Belief about the experiment's outcome is represented as a probability vector over the cells,

$$p \in \Delta^{K-1},$$

rather than as a scalar belief-in-truth.

For each node, we elicit two such allocations from the belief model π_b. The pre-experiment elicitation conditions on the seed hypothesis and variable schema only, while the post-experiment elicitation additionally conditions on the analyst's summary of the experiment's results. Each elicitation draws N independent samples from π_b, where each sample is a 100-point integer allocation across the K cells. We drop malformed samples, renormalize each surviving allocation to a probability vector, and average across survivors, yielding a prior distribution $p^{(p)}$ and a posterior distribution $p^{(q)}$. We floor every cell at $\varepsilon = 10^{-6}$ before renormalizing so that

$$p_k^{(p)} > 0 \qquad \text{for all } k,$$

ensuring that the KL divergence is finite.

The raw Bayesian surprise of a node is the KL divergence from prior to posterior, measured in bits:

$$\mathrm{KL}\Big(p^{(q)} \,\|\, p^{(p)}\Big) = \sum_{k=1}^{K} p_k^{(q)} \log_2 \frac{p_k^{(q)}}{p_k^{(p)}}. \tag{12}$$

This quantity has no fixed upper bound across nodes with different contrast-set sizes K. To make the reward comparable across contrast sets of varying cardinality, we normalize by the maximum entropy of a K-cell distribution, $\log_2 K$, and clip the result at 1:

We use

$$\mathrm{NK}(v) = \min\left(\frac{\mathrm{KL}\big(p^{(q)} \,\|\, p^{(p)}\big)}{\log_2 K}, 1\right) \in [0,1]$$

as the MCTS reward for the contrastive condition. The unclipped entropy-scaled score is $\kappa_t = \mathrm{KL}/\log_2 K$, and we classify a node as surprising when $\kappa_t \geq 0.5$. Thus, clipping is applied only to the MCTS reward; the information-gain analyses and binary surprising indicator use the unclipped κ_t. Intuitively, NK is a clipped, cardinality-normalized measure of how strongly the observed evidence redistributes belief across the contrast cells. A value of 1 indicates that the unclipped score $\mathrm{KL}(p^{(q)} \,\|\, p^{(p)})/\log_2 K$ is at least 1, while a value near 0 corresponds to an experiment that produces little change in the belief distribution. Because $\log_2 K$ is not generally the maximum possible KL divergence, NK should not be interpreted as the fraction of all possible belief redistribution achieved.

M.3. Agent framework

We follow the setup of AutoDiscovery's (Agarwal et al., 2026) discovery agent with slight modifications. We use the following LLM agents with a `gpt-4o` backbone:

Hypothesis (Contrast Set) Generator. Proposes a candidate contrast set of hypotheses that predicts the outcome of an experiment. Produces an experiment plan for evaluating the proposed contrast set.

Experiment Coder. Implements the experiment plan in code. The coder is allowed up to six implementation attempts, using feedback from the Experiment Analyst to revise failed or incomplete attempts.

Code Executor. Executes the code produced by the Experiment Coder and returns the exit code and standard output. This component is not LLM-based.

Experiment Analyst. Analyzes the output of code execution and provides feedback to the Experiment Coder when the implementation requires correction or refinement.

Experiment Reviewer. Reviews the completed experiment for alignment with the original experiment plan and assesses whether the results can support validation of the hypothesis.

Experiment Reviser. Produces a revised experiment when the original experiment or its implementation fails to execute successfully or cannot adequately validate the hypothesis.

Belief Elicitor. Distributes 100 points over the contrast set of hypotheses, evaluating whether each hypothesis is believed to be true relative to alternatives both before experimental evidence (prior) and after (posterior). For the *pointwise* case, evaluates whether the hypothesis is believed to be true from the hypothesis statement alone (prior) or from the hypothesis statement together with the experimental results (posterior).

Image Analyst. Provides textual descriptions of images produced by code written by the Experiment Coder.

M.4. Example contrast sets

Example Contrast Set

Dataset: BLADE Mortgage

Contrast set: Given a poor credit score, self-employment status leads to

MECE Contrast Set

H1. higher denial rates than non-self-employed
H2. no different denial rates
H3. lower denial rates than non-self-employed

Example Contrast Set

Dataset: Materials Project

Contrast set: Direct-gap vs indirect-gap materials, Fermi energy is

MECE Contrast Set

H1. higher in direct-gap materials
H2. lower in direct-gap materials
H3. not substantially different

Example Contrast Set

Dataset: SDSS Galaxy

Contrast set: Petrosian half-light radius (PetroR90_r) is predominantly determined by

MECE Contrast Set

H1. model magnitude attributes
H2. extinction attributes
H3. neither significantly drives it

Example Contrast Set

Dataset: NLS Raw

Contrast set: Trend in vehicle market value across 1985/1990/1996 is

MECE Contrast Set

H1. significant linear positive
H2. significant linear negative
H3. nonlinear or no significant trend

Example Contrast Set

Dataset: SDSS Galaxy

Contrast set: When controlling for galaxy axis ratios (expAB_r, deVAB_r), the redshift effect on stellar velocity dispersion

MECE Contrast Set

H1. increases (strengthens)
H2. decreases (attenuates)
H3. disappears

N. Broader Societal Impact

The potential negative social impacts of our method align with those typically associated with general LLM reasoning technologies. We emphasize the importance of adhering to the principles of fair and safe deployment in LLM systems.

O. Cost Analysis

Each 200-iteration run for a given dataset costs $20 to $50 USD. We estimate the total cost of our experiments at $1500 to $2000 USD.