SEARCHING FOR WHAT MATTERS: OPEN-ENDED DISCOVERY OF MULTIPLE HYPOTHESES WITH AGENTIC DATA EXPLORATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Discovering what drives an outcome from data is central to science and often requires deciding what is worth investigating before a specific hypothesis is known. Yet most existing hypothesis-generation settings substantially narrow this search by providing a research question, target relationship, or curated context. We introduce **Open-ended Hypothesis Discovery**, where a system receives only a large tabular dataset and an outcome and must discover multiple data-supported hypotheses without knowing in advance the relevant variables, functional forms, conditions, or number of underlying mechanisms. To study this problem, we introduce **OpenHypBench**, a benchmark built from structural causal models grounded in real-world data schemas, containing 351 planted hypotheses across 15 tasks and five domains, with 53–65 variables and 1M–10M rows per task. The causal model is hidden from the discovery method and retained only by the evaluator, allowing proposed hypotheses to be tested through interventions and separating observational discovery from causal validation. We further propose **HypAgent**, which treats the dataset as an interactive environment and performs iterative search by proposing candidate directions, triaging and refining promising hypotheses, and removing explained signal so that each accepted discovery redirects subsequent search. HypAgent achieves 0.47 precision, 0.51 recall, and 0.49 F1, compared with 0.35 F1 for the strongest baseline. Our analysis shows that many failures occur after a relevant variable has already been surfaced, suggesting that correctly characterizing a mechanism's functional form and conditions, rather than merely identifying relevant variables, remains a key challenge in open-ended discovery.[1]

1 INTRODUCTION

Data-driven discovery often begins before there is a hypothesis to test: one must first decide *what is worth investigating*. This remains difficult even as large language models and agents increasingly automate downstream work such as coding, data analysis, and hypothesis testing (Yang et al., 2024; Guo et al., 2024; Jing et al., 2025; Chen et al., 2025). Consider a large tabular e-commerce dataset. Purchase behavior may simultaneously depend on a discount threshold, delivery-time sensitivity, and a marketing effect that appears only for a particular customer segment. A scientist may not know beforehand that any of these mechanisms exist, which variables define them, or how many distinct mechanisms are present.

Most existing hypothesis-generation settings substantially narrow this search in advance by providing a research question, discovery goal, prediction target, treatment, or curated scientific context (Majumder et al., 2025; Yang et al., 2025; Liu et al., 2026; Zhou et al., 2024; Qi et al., 2024; Liu et al., 2025b). We instead study **Open-ended Hypothesis Discovery**: given only a tabular dataset D and an outcome Y, discover a set $H = \{h_1, \ldots, h_K\}$ of data-supported hypotheses without guidance about which variables to investigate, what functional relationships to consider, under what conditions they hold, or even how many distinct hypotheses K are present. Here, *open-ended* refers to the discovery target being unspecified while the input itself remains structured tabular data.

[1] Code: `https://anonymous.4open.science/r/ICLR-hyp-disc-272E/`.

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

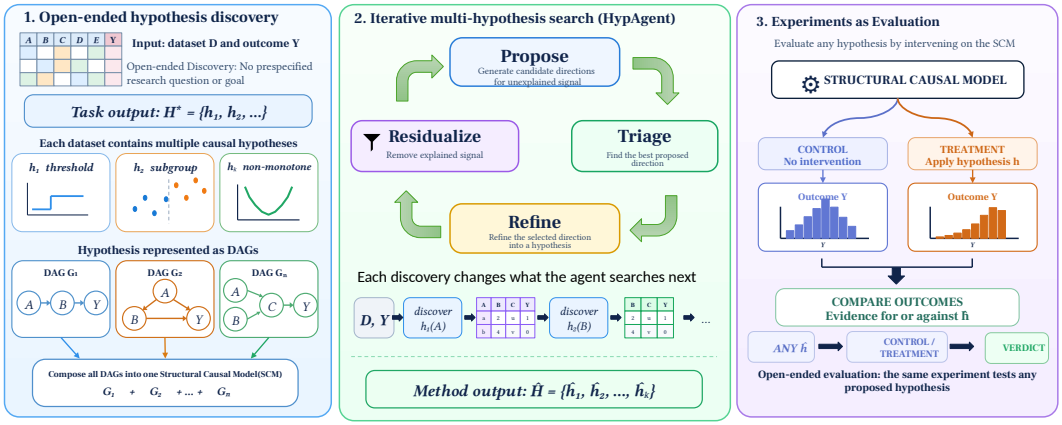


Figure 1: **Overview of Open-Ended Hypothesis Discovery. Left:** Given only a tabular dataset D and outcome Y, the goal is to discover multiple unknown hypotheses. **Middle:** HypAgent iteratively proposes, triages, refines, and residualizes hypotheses so that each discovery guides the next search. **Right:** OpenHypBench evaluates proposed hypotheses through interventions on the hidden structural causal model.

This setting creates a difficult search problem. The space of possible hypotheses grows rapidly with the number of variables, nonlinear relationships, interactions, thresholds, and subgroup conditions. Moreover, multiple mechanisms may coexist in the same data, so discovering one strong relationship does not imply that the discovery process is complete. A system must therefore do more than generate a plausible hypothesis once. Existing method families address only parts of this problem. Statistical and causal-discovery methods can operate over large datasets, but typically require the treatment or hypothesis family to be specified in advance, or return structural relationships rather than complete hypotheses (Agrawal et al., 1993; Chernozhukov et al., 2018; Wager & Athey, 2018; Spirtes et al., 1993; Chickering, 2002). LLM-based hypothesis generators are more expressive, but generally reason over examples or summaries placed in the model context (Zhou et al., 2024; Yang et al., 2025; Liu et al., 2025b). Data-science agents can execute code on large datasets, but are typically given an external objective, such as optimizing a predictive metric (Guo et al., 2024; Hong et al., 2025). Open-ended discovery instead requires the system itself to decide what is worth investigating next.

Evaluation poses a second challenge. Reference-set matching alone is insufficient because hypotheses involving similar variables may differ in functional form, direction, conditions, or effect magnitude, while a valid proposed hypothesis need not be phrased identically to a planted reference. To make this problem measurable, we introduce **OpenHypBench**, a benchmark built from structural causal models grounded in schemas and empirical variable distributions of real-world datasets. Each task contains multiple heterogeneous mechanisms composed into a structural causal model (SCM), yielding large datasets in which different effects can overlap, interact, or apply only under particular conditions. Across 15 tasks that span five domains, OpenHypBench contains 351 planted hypotheses, 53-65 variables per task, and datasets ranging from 1M to 10M rows. The underlying SCM is hidden from the discovery method and retained only by the evaluator. For a given proposed hypothesis, the evaluator performs the corresponding intervention on its variables and measures the resulting change in the outcome. Consequently, the discovery method uses observational data to generate and support candidate hypotheses, while OpenHypBench independently tests their interventional validity (Pearl, 2009; 2019).

To solve the Open-ended Hypothesis Discovery task, we introduce **HypAgent**, an agent designed to iteratively search for multiple hypotheses in datasets too large to place directly in an LLM context. Rather than treating the dataset as text, HypAgent treats it as an environment that can be queried programmatically. Each discovery cycle first proposes candidate directions, uses lightweight analyses over sampled observations to triage promising candidates, and then queries the full dataset to refine the selected direction into a more specific data-supported hypothesis. Accepted hypotheses are used to model and remove explained outcome variation before the next discovery cycle. The central principle is that *each accepted discovery changes what is searched next*. Inspired by match-

ing pursuit (Mallat & Zhang, 1993), this residual-driven procedure progressively redirects the search to the unexplained signal, as illustrated in Figure 1.

Across OpenHypBench, HypAgent achieves an F_1 of 0.49, compared to 0.35 for the strongest baseline. Its recall advantage persists under matched hypothesis-proposal budgets. Our analysis further shows that identifying a relevant variable is often insufficient because the majority of failures occur after a promising variable has already surfaced. These results suggest that open-ended hypothesis discovery requires both an efficient search over candidate mechanisms and careful refinement of the hypotheses that the search uncovers, while overall performance leaves substantial room for improvement. The contributions of this work are:

- **New Task:** We formulate **Open-ended Hypothesis Discovery**, where a system must discover multiple hypotheses from a large tabular dataset and outcome without being given the variables, functional forms, conditions or number of mechanisms to investigate.
- **New Benchmark:** We introduce **OpenHypBench**, comprising 351 planted hypotheses across 15 tasks and five domains, together with an interventional evaluation framework that tests the proposed hypotheses in addition to measuring recovery against the planted reference set.
- **Agentic Framework:** We propose **HypAgent**, an iterative residual-driven agent that treats the dataset as an interactive environment and uses each accepted discovery to redirect the subsequent search towards unexplained outcome variation. Our experiments further identify mechanism characterization, rather than merely surfacing relevant variables, as a key remaining challenge.

2 Related Work

Benchmarks for hypothesis discovery. Automated scientific discovery has historically been studied in constrained settings where the hypothesis space is defined in advance. Early systems recovered physical laws or causal structures from data but operated over fixed variable sets and predefined hypothesis families (King et al., 2009; Schmidt & Lipson, 2009), and causal discovery benchmarks inherited this structure by evaluating recovery of a target graph over a known variable set (Sachs et al., 2005; Scutari, 2010). More recent benchmarks broaden the range of expressible hypotheses but still evaluate against predefined targets: findings drawn from published scientific work (Majumder et al., 2025; Liu et al., 2025a), expert-authored questions or known solutions (Gu et al., 2024; Chen et al., 2025; Jing et al., 2025; Jansen et al., 2024), or hypotheses generated from provided context (Qi et al., 2024; Liu et al., 2026). Chen et al. (2026) evaluate hypothesis sets within finite and exactly enumerable spaces. These designs do not directly evaluate an arbitrary supported proposal that falls outside the predefined target set, which is required in our open-ended setting. Table 1 summarizes previous hypothesis discovery benchmarks in the context of open-ended hypothesis discovery.

Table 1: Comparison of existing hypothesis discovery benchmarks. *Scale* denotes the maximum number of rows in a benchmark dataset. *New targets* indicates that the evaluated data and target effects are newly generated rather than published targets.

Benchmark	Scale	Discovery setting	Open-ended evaluation	New targets	Causal
Context-based generation (Qi et al., 2024)	3K	Curated context	✗	✗	✗
Data-driven analysis (Gu et al., 2024)	∼100K	Prespecified question	✗	✗	✗
Hypothesis suites (Liu et al., 2025a; Majumder et al., 2025)	∼100K	Prespecified goal	✗	✗	✗
Causal suites (Sachs et al., 2005; Scutari, 2010)	10K	Prespecified goal	✗	✗	✓
OpenHypBench (ours)	**10M**	**Open-ended**	✓	✓	✓

Methods for hypothesis generation. Prior methods divide along two axes, whether they generate hypotheses in open-ended settings and whether they *scale without placing data in context*. LLM-based hypothesis generators (Zhou et al., 2024; Liu et al., 2025b) typically receive a posed question or scientific context and reason over examples or summaries placed in the prompt. LLM data-science agents such as DS-Agent (Guo et al., 2024) and Data Interpreter (Hong et al., 2025) can execute code over external datasets, but they optimize a fixed, externally specified objective rather than deciding what is worth investigating. Statistical pipelines rank associations or predictive features, while causal-structure methods such as PC and GES estimate graphical relationships over a specified variable set (Spirtes et al., 1993; Chickering, 2002). Neither directly returns the complete hypotheses required here, including their conditions, directions, and magnitudes.

3 PROBLEM FORMULATION

Let $D = \{(X_i, Y_i)\}_{i=1}^{n}$ denote an observational dataset with features $X = \{X_1, \ldots, X_p\}$ and a designated outcome Y. Hypothesis h is a testable claim about how intervening in one or more variables affects the outcome under specified conditions. We represent it as $h = (X_h, \psi, d, c, m_h)$, where $X_h \subseteq X$ contains the variables that participate in the mechanism, $T_h \subseteq X_h$ denotes the variables intervened on to test the claim, ψ specifies their structural relationship with the outcome, such as a monotone, thresholded, context-dependent or mediated relationship, $d \in \{+, -, \text{mixed}\}$ denotes the claimed direction of the effect, with mixed indicating that the direction changes across the regions or conditions specified by ψ and c. The term c specifies the condition or population for which the claim is valid and $m_h \geq 0$ denotes the claimed absolute change in the probability of the outcome. For example,

$$h_1 = (\{\texttt{recency}\}, \texttt{threshold}(\texttt{recency} > 47), +, \text{all customers}, 0.06)$$

states that, for all customers, purchase recency above 47 days increases the probability of churn by six percentage points. Together, the components of h distinguish hypotheses that involve similar variables but make different claims about their conditions, direction, or magnitude.

Definition 1 (Open-ended Hypothesis Discovery)**.** *Given an observational dataset D and a designated outcome Y, the goal of Open-ended Hypothesis Discovery is to produce a set of testable causal hypotheses $\widehat{H} = \mathcal{A}(D, Y) = \{\hat{h}_1, \ldots, \hat{h}_{\widehat{K}}\}$ about distinct mechanisms affecting Y. The method is not given the number of mechanisms, intervention variables, participating variables, structural relationships, conditions, directions, or effect magnitudes to search for.*

For each task, $H^\star = \{h_1^\star, \ldots, h_{K^\star}^\star\}$ denotes the set of $K^\star$ planted mechanisms hidden from the discovery method. Because Open-ended Hypothesis Discovery seeks multiple hypotheses rather than a single answer, we distinguish *mechanism identity* from *hypothesis correctness*. Two hypotheses correspond to distinct mechanisms when they differ in their variable set X_h or structural relationship ψ, or when their conditions c are not equivalent. Differences in wording, direction, or effect magnitude alone do not define a new mechanism. Instead, direction and magnitude determine whether a claim about that mechanism is correct. Section 4 describes how OpenHypBench instantiates this task using observational datasets with multiple hidden causal mechanisms and an interventional evaluator.

4 OPENHYPBENCH

Evaluating Open-ended Hypothesis Discovery requires more than matching generated hypotheses against a fixed answer set. A proposal may make a valid claim that is not identical to a planted reference, while a hypothesis involving the correct variables may still be wrong in its structural form, conditions, direction, or magnitude. At the same time, real observational datasets do not provide interventional ground truth for arbitrary proposed claims. We therefore construct OpenHypBench as a collection of controlled data-generating worlds in which the underlying causal mechanisms are known to the evaluator but hidden from the discovery method.

4.1 CONSTRUCTING MULTI-MECHANISM DATA-GENERATING WORLDS

Each OpenHypBench task begins from a public tabular dataset. We retain its variable vocabulary, data types, semantics, and empirical marginal distributions, but construct new causal mechanisms and generate new observations specifically for the benchmark. This preserves realistic variable semantics and marginal ranges while allowing the causal structure and interventional ground truth to be controlled. OpenHypBench contains 15 tasks spanning census, e-commerce, education, finance, and health, with three distinct source schemas per domain. Tasks contain 53–65 visible features and 1M–10M generated observations, and the benchmark contains 351 planted mechanisms in total. Table 2 summarizes the domains and source schemas; Appendix C provides the complete task inventory, including outcomes, licenses, scale, feature counts, and individual mechanism counts.

To instantiate heterogeneous discovery challenges, OpenHypBench uses seven hypothesis families: monotone threshold, non-monotone, derived-feature, interaction-reversal, subpopulation-specific, Simpson's-reversal, and mediation. These families define the mechanisms used to construct the SCM, and discovery methods are not required to predict a family label. Appendix B defines each family and reports its frequency. Candidate mechanisms are proposed using Claude Opus 4.8 with the prompt reported in Appendix E.1. Each retained hypothesis is represented using the formulation in Section 3 and instantiated as an individual causal DAG, as illustrated in Figure 2. The individual

Domain	Source schemas	Feat.	Mech.
Census	ACS, Adult, CPS	63–64	73
E-commerce	H&M, Instacart, Online Retail	61–63	68
Education	OULAD, Student Performance, xAPI	58–65	70
Finance	Give Me Some Credit, LendingClub, Taiwan	58–65	70
Health	BRFSS, Diabetes130, NHANES	53–65	70
Total	15 tasks	53–65	351

Table 2: Summary of OpenHypBench. Each domain contains 3 tasks with 1M–10M rows. Dataset citations and licenses are reported in Appendix C.

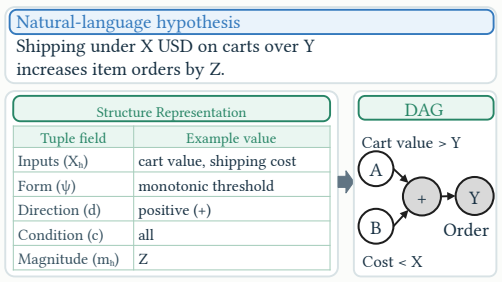


Figure 2: Example mapping from a natural-language hypothesis to the structured representation used by OpenHypBench and its corresponding mechanism DAG.

DAGs are then composed into a single task-level structural causal model (SCM). Specifications containing directed cycles, undefined parents, missing moderators, or incompatible operations on variable types are rejected before data generation.

Variables in the resulting SCM serve different roles, including causes, confounders, moderators, mediators, background variables, decoys, and null variables. In particular, decoys may correlate with causal variables through shared upstream structure while having no direct outcome mechanism, creating plausible but non-causal directions for discovery methods to distinguish from true mechanisms. The SCM jointly generates all observations in a task. Root variables are sampled from the empirical distributions of the corresponding source variables, and remaining variables are generated in topological order according to structural assignments $V_j := f_j(\mathrm{Pa}(V_j), \epsilon_j)$, where $\mathrm{Pa}(V_j)$ denotes the parents of V_j and ϵ_j captures exogenous variation. A mechanism may be active only under particular conditions or within specific regions of the feature space. Consequently, multiple heterogeneous mechanisms may coexist in the same dataset while all observations arise from one shared SCM. Importantly, the planted mechanisms participate directly in the data-generating process rather than being attached to generated observations as post hoc annotations.

4.2 Interventional Evaluation of Proposed Hypotheses

The SCM used to generate each task is hidden from every discovery method and retained only by the OpenHypBench evaluator. Methods produce hypotheses in the common representation defined in Section 3 or their outputs are parsed into that representation. For each proposed hypothesis, the evaluator constructs a deterministic experiment plan specifying the intervention variables, values, and any conditions needed to test the stated claim. Appendix D describes proposal normalization, deduplication, and recall matching.

Let T_h denote the intervention variables of hypothesis h. Executing $do(T_h = t)$ replaces the structural assignments of those variables with the intervention values t, removes their incoming edges, and regenerates their descendants in topological order while leaving the remainder of the SCM unchanged (Pearl, 2019). Each hypothesis is evaluated using 60,000 paired samples under the required intervention conditions. For a given hypothesis, the resulting causal contrast can be written as $\tau_h = \mathbb{E}[Y \mid do(T_h = t_1), c] - \mathbb{E}[Y \mid do(T_h = t_0), c]$, where c denotes any population restriction specified by the hypothesis. Here, τ_h is the signed causal contrast measured by the evaluator, whereas m_h is the nonnegative magnitude claimed in the hypothesis representation. Conditions used for such comparisons must involve variables that are not descendants of T_h. Experiment plans conditioning on descendants of an intervened variable are rejected. Other structural forms induce corresponding form-specific experiments and contrasts, such as evaluating interaction reversals across moderator values or non-monotone effects across multiple treatment regions. The resulting interventional response is compared with the structural form, direction, condition, and magnitude of the claim. Thus, the discovery method reasons only from observational data, while causal support is assessed separately through interventions on the hidden SCM.

Evaluation Metrics: We evaluate methods using *precision*, *recall*, and F_1 score. These quantities capture complementary aspects of open-ended discovery: precision measures whether the distinct hypotheses proposed by a method are supported by intervention, while recall is the fraction of the $K^\star$ planted mechanisms recovered by the method. Before computing precision, proposals are deduplicated

cated according to the mechanism-identity rule in Section 3, retaining the first proposal encountered for each identity without using interventional support to select the representative. A supported proposal need not match a planted hypothesis and may therefore receive precision credit even if it lies outside the planted reference set. Recall measures recovery of the set of planted mechanisms $H^\star$. The supported proposals are first shortlisted against the planted mechanisms using their intervention variables, interventional response structure, conditions, and magnitude of the effect. Claude Opus 4.8 then performs the final mechanism-equivalence judgment. Multiple proposals matched to the same planted mechanism count only once. Unless otherwise stated, a claimed effect magnitude m_h is considered correct when it differs by at most 10% from the corresponding interventional effect magnitude. Refer to Appendix D for further details on implementation.

5 HypAgent: Residual-Driven Hypothesis Search

In practical settings, Open-ended Hypothesis Discovery requires reasoning over datasets far larger than an LLM's context window. Directly ingesting such data is infeasible, while subsampling, chunking, or reducing the data to aggregate statistics can disturb the distributional structure and miss dependencies important for discovery. HypAgent therefore enables the LLM to decide what data to investigate, while analyses are executed programmatically on sampled observations. We implement this through a read-eval-print loop in which the LLM writes code to query the data instead of ingesting entire data in context.

A key challenge in generating hypotheses in an open-ended setting is that strong effects can repeatedly dominate the search, causing a method to rediscover variations of hypotheses it has already identified, while weaker mechanisms remain unexplored. We therefore design HypAgent around a simple principle: *each confirmed discovery should change what is searched next*. We operationalize the search using a matching-pursuit-inspired procedure (Mallat & Zhang, 1993). Matching pursuit decomposes a signal by repeatedly identifying a component that explains part of it, removing its contribution, and continuing on the residual. In our setting, the signal is the outcome Y, and the components are candidate hypotheses. Each round consists of three stages: *residualization* removes the effects already discovered, *triage* identifies which proposed candidate best explains the remaining signal, and *refinement* recovers conditional structure and validates the resulting hypothesis. The agent operates only on observational data and does not access the hidden SCM or the interventional evaluator used by OpenHypBench. Refer to Algorithm 1 for pseudocode of HypAgent.

Residualization. To address the challenge of recovering hypotheses beyond those that dominate the search, during each iteration HypAgent removes the variation explained by confirmed hypotheses, thereby changing the search target after each discovery. Let S_k denote the hypotheses confirmed after round k. Each hypothesis $h \in S_k$ contributes a representation $\phi_h(X)$ of its named variables, and $\Phi(S_k)$ denotes their combined representation. Appendix D.1.1 specifies the variable encodings. Using the class $\mathcal{F}_{\text{lin}}$ of linear predictors, we define $\hat{f}_k = \arg\min_{f \in \mathcal{F}_{\text{lin}}} \|Y - f(\Phi(S_k))\|_2^2, \qquad \widetilde{Y}_k = Y - \hat{f}_k(\Phi(S_k))$. The next iteration operates on $\widetilde{Y}_k$ rather than Y. Consequently, hypotheses already captured by S_k contribute less signal to subsequent search, shifting attention toward mechanisms that explain previously uncovered variation. Residualization removes fitted observational variation rather than a known causal mechanism. It therefore redirects search toward signal not yet represented by the accepted hypotheses, without assuming a unique causal decomposition.

Proposal & Triage. The space of candidate hypotheses that explain the residual is too large to traverse systematically. HypAgent leverages the world knowledge of LLMs to propose a set of candidate hypotheses using the dataset and the hypotheses confirmed so far. Rather than spending resources on each candidate hypothesis, HypAgent first performs a lightweight data-driven triage by querying the data to determine which candidate hypothesis explains the largest fraction of the remaining outcome variation. This step provides exploration in the discovery process by asking which broad direction explains most of the residual variation.

For each candidate variable T, we compute its empirical squared correlation ratio with the current residual, $s(T) = \frac{\mathrm{Var}\big(\mathbb{E}[\widetilde{Y}_k|\mathrm{bin}(T)]\big)}{\mathrm{Var}(\widetilde{Y}_k)}$, where $\mathrm{bin}(T)$ denotes quantile bins for continuous variables and observed categories for categorical variables. HypAgent uses a tunable threshold ϵ, requiring $s(T) \geq \epsilon$ for a candidate to be retained for further analysis. Among candidates that clear this threshold, the highest-scoring candidate is selected. Triage therefore allows HypAgent to cheaply filter a broad set of directions before committing to analyze a single candidate.

Refinement. Triage identifies a promising starting point, but this candidate hypothesis is specific to the narrow slice of data HypAgent queried previously and influenced by the previous hypotheses. Committing the initial candidate directly would therefore favor simple explanations of what may be more structured relationships. Once a candidate $T^\star$ is selected, HypAgent shifts from broad search to exploitation where it queries the data to refine $T^\star$ towards a hypothesis that most accurately explains the remaining outcome. These queries inspect the candidate across different regions of its distribution, subpopulations, and potential conditioning variables, progressively refining the initial direction into a more specific hypothesis. Refinement seeks to answer: *Given a direction, what more specific formulation explains more of that variation in the outcome residual?*

Once HypAgent narrows down to a specific hypothesis, two sources of ambiguity still remain. First, the relationship associated with $T^\star$ may change depending on another variable. To test this, HypAgent leverages the world knowledge and reasoning capabilities of LLMs to identify candidate moderators M and fits $\widetilde{Y}_k = \beta_0 + \beta_T T^\star + \beta_M M + \beta_{TM}(T^\star \times M)$. The interaction coefficient β_{TM} measures whether the relationship associated with $T^\star$ changes with M. This resolves the ambiguity of opposing effects across different contexts that can appear weak or even disappear when viewed only in aggregate. A second ambiguity is confounders, where a candidate can explain residual variation because it is associated with another variable that is responsible for the signal. HypAgent therefore identifies potential confounders Z among covariates strongly associated with $T^\star$ and recomputes the effect after adjustment. If the relationship disappears after adjustment, the initial evidence is not sufficient to support the proposed claim. Otherwise, the evidence collected during refinement is used to specify the variables, conditions, direction, and magnitude of the final hypothesis before it is added to the confirmed set. Refer to Appendix D.1.3 for implementation details.

6 Experiments

We evaluate four aspects of Open-ended Hypothesis Discovery: **RQ1:** How accurately and efficiently do methods discover hypotheses? **RQ2:** How does hypothesis structure affect recovery? **RQ3:** How do data size and task complexity affect discovery? **RQ4:** What does each component of HypAgent contribute?

Empirical Setup. We compare HypAgent with eight baselines spanning statistical and causal methods (**Correlation Mining**, **Subgroup Analysis**, and **PC** (Spirtes et al., 1993)), LLM-based methods (**XGBoost+LLM** and **LLM (no code)**), and agents (**ReAct** (Yao et al., 2022), **Claude Code**, and **AutoDiscovery** (Agarwal et al., 2025)). All LLM-based methods use Claude Sonnet 4.6, and all outputs are scored using the same OpenHypBench evaluator with 60,000 post-intervention samples and a 10% effect-magnitude tolerance. We compute confidence intervals using 2,000 cluster-bootstrap resamples over tasks. Appendix D.3 details method inputs, configurations, and output adapters.

6.1 RQ1: Overall Performance on Open-ended Hypothesis Discovery

(a) Overall performance and runtime

Method	Precision	Recall	F_1	Time (s/task)
Correlation Mining	0.00	0.01	0.00	53
Subgroup Analysis	0.06	0.05	0.05	158
PC	0.00	0.00	0.00	504
XGBoost+LLM	0.20	0.40	0.30	778
LLM (no code)	0.18	0.33	0.24	711
AutoDiscovery	0.21	0.38	0.27	2,458
ReAct	0.09	0.16	0.11	1,986
Claude Code	0.31	0.39	0.35	3,308
HypAgent	**0.47**	**0.51**	**0.49**	2,681

(b) Recall across proposal budgets

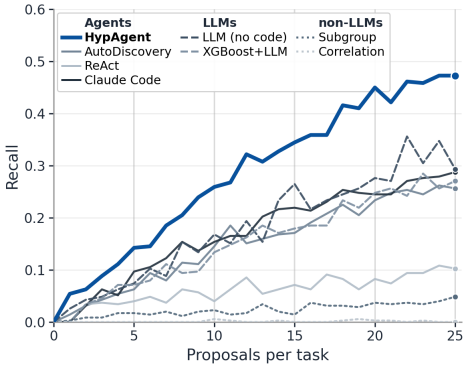


Figure 3: Overall discovery performance on OpenHypBench. (a) Point estimates at each method's reported operating point; time is mean end-to-end discovery time per task and excludes the shared evaluator. Confidence intervals are reported in Figure 6 in the appendix. (b) Recall after uniformly subsampling each method to a common budget of up to 25 proposals per task.

HypAgent achieves the strongest overall discovery performance. Figure 3 compares methods on precision, recall, and F_1. HypAgent achieves a precision of 0.47, recall of 0.51, and F_1 of 0.49. Claude Code has the strongest baseline precision at 0.31 and the strongest baseline F_1 at 0.35, while XGBoost+LLM has the strongest baseline recall at 0.40. HypAgent therefore improves both the correctness and coverage of the discovered hypotheses.

Failure modes vary across method families. Statistical and causal methods achieve near-zero performance. These tools are effective when given a prespecified treatment or goal, but lack a mechanism to identify the conditions and effect magnitudes required by OpenHypBench. LLM-based methods generate a wider range of hypotheses, but their higher recall is accompanied by lower precision. In particular, we observe that the vanilla LLM baseline can propose plausible hypotheses but cannot test them against the data, while XGBoost+LLM grounds its proposals in feature importance without fully validating their causal structure. These results show that Open-ended Hypothesis Discovery requires both expressive hypothesis generation and data-grounded validation. Agentic baselines show contrasting performance, with ReAct performing the worst while Claude Code is the strongest baseline agent by F_1. AutoDiscovery's surprisal objective may help explain its similar performance to generic LLM methods because an unexpected observational association can receive more search priority than a valid but unsurprising causal mechanism. This leads to a relatively high recall (0.38) but modest precision (0.21) on OpenHypBench.

HypAgent remains strongest under matched proposal budgets. Figure 3b further compares recall under matched proposal budgets. HypAgent achieves the highest recall throughout the range from one to 25 proposals per task, with its advantage generally increasing as the budget grows. This result indicates that HypAgent provides a better precision-recall tradeoff rather than obtaining greater coverage through a larger output set. Other LLM-based methods broadly follow similar trajectories. This suggests that the increase in recall of LLMs using XGBoost as a feature selector stems from their greater quantity of hypothesis proposals. Finally, **Open-ended Hypothesis Discovery remains far from solved.** No baseline exceeds an F_1 of 0.35, and even HypAgent recovers only 51% of the planted hypotheses.

6.2 RQ2: Effect of Hypothesis Structure

Hypothesis structure strongly determines discoverability. Figure 4 shows that discovery performance varies substantially across hypothesis structures. HypAgent achieves the highest recall on five of the seven archetypes, whereas XGBoost+LLM performs best on U-shaped effects and LLM (no code) performs best on monotone effects. These structures require more than selecting the correct variable: a threshold claim must identify where the relationship changes, a subpopulation claim must specify where it applies, and a Simpson's-reversal claim must distinguish the marginal direction from the direction within strata.

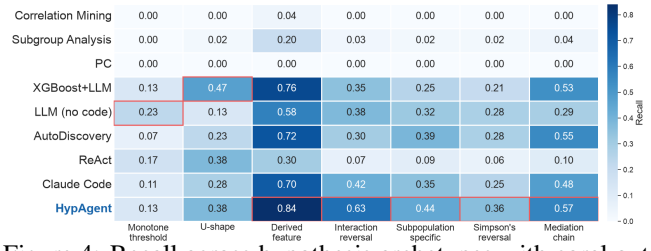


Figure 4: Recall across hypothesis archetypes with coral outlines marking the best-performing method for each archetype.

Compositional mechanisms are easier to recover than thresholds and aggregation reversals. We observe that hypotheses formed as a derivation of other features are consistently easier to discover, with performance between 0.58 and 0.84 across strong baselines. Surprisingly, monotone threshold hypotheses are difficult to recover across all methods, as are Simpson's reversals and hypotheses specific to a subpopulation. U-shaped and interaction-reversal hypotheses exhibit higher variance in performance, while hypotheses mediated or confounded by related features are modestly recovered by agentic baselines. Overall, methods perform better on hypotheses involving semantically meaningful feature compositions or causal chains than on hypotheses whose structure depends on subtle thresholds, subgroup boundaries, or reversals under aggregation.

The main source of failure is characterizing the effects of relevant variables. We systematically analyze HypAgent's failed recoveries by categorizing the 172 missed planted hypotheses. Wrong-form errors account for 87 misses (50.6%), missed variables account for 53 (30.8%), and wrong effect magnitudes account for 32 (18.6%). Under these criteria, characterization errors account for 119 of the 172 missed mechanisms (69.2%), after a relevant variable has already been identified. Refer to Figure 7 and Appendix D.4 for the complete breakdown.

6.3 RQ3: Factors Affecting Open-ended Hypothesis Discovery

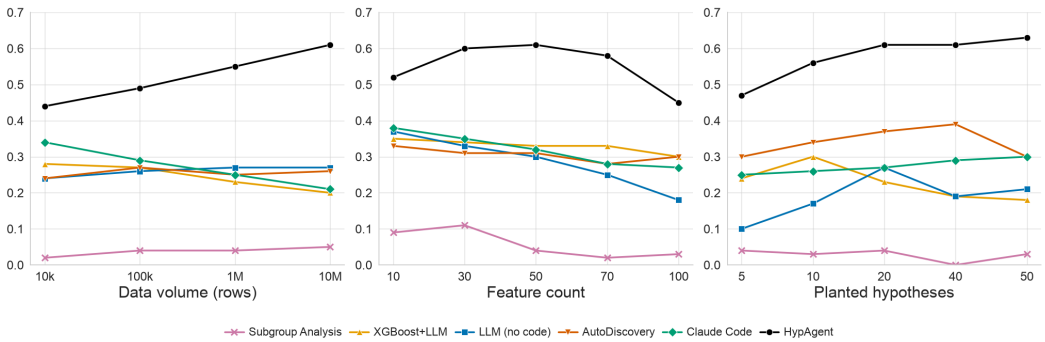


Figure 5: F_1 as the number of rows, features, and planted hypotheses varies.

To study the factors affecting discovery, we vary one benchmark dimension at a time across all domains while holding the others at a common default of 10^6 rows, 50 features, and 20 planted hypotheses, and average the results across tasks. We observe that additional data benefit HypAgent, whose F_1 increases from 0.44 at 10^4 rows to 0.61 at 10^7, while most baselines remain flat or decline. Increasing the number of features generally hurts performance as HypAgent reaches $F_1 = 0.61$ at 50 features but falls to 0.45 at 100. By contrast, increasing the number of planted hypotheses does not consistently increase difficulty as HypAgent improves from $F_1 = 0.47$ with five hypotheses to approximately 0.63 with 50, while the baselines exhibit weaker or inconsistent trends. These findings indicate that more data strengthens the evidence available for testing a candidate, and more planted mechanisms can provide additional discoverable signal. Hence, **the search space matters more than the amount of available evidence**, since Open-ended Hypothesis Discovery provides no prespecified target and its main difficulty lies in deciding which explanations to test rather than gathering enough observations to test them.

6.4 RQ4: Contribution of HypAgent Components

We ablate each component of HypAgent across all 15 tasks of OpenHypBench. Our findings reveal that residualization prevents dominant explanations from repeatedly occupying the search, allowing later iterations to focus on unexplained outcome variation and helping prevent duplicate hypotheses. Triage directs the limited refinement budget toward candidate directions with stronger residual evidence. Refinement has the largest contribution because it transforms a broad direction into a specific hypothesis by determining its conditions, effect, and relevant variables, while filtering observational patterns that are not robust. Together, the components support complementary stages of discovery: residualization promotes diversity, triage improves search allocation, and refinement improves the specificity and reliability of the resulting hypotheses.

Table 3: Ablation of HypAgent components across all 15 tasks. Ablated rows report changes relative to the full method.

Configuration	P	R	F_1	Impact
HypAgent	0.47	0.51	0.49	–
− Residualization	−0.15	−0.12	−0.13	68.8% increase in duplicates
− Triage	−0.08	−0.13	−0.11	Suboptimal direction in 25.5% of cases
− Refinement	−0.31	−0.36	−0.34	58.5% increase in unsupported hypotheses

7 Conclusion

Open-ended Hypothesis Discovery shifts the focus from explaining a prespecified relationship to discovering multiple data-supported hypotheses from a dataset and an outcome. We introduced OpenHypBench to evaluate this capability in controlled causal worlds, separating observational discovery from interventional validation, and HypAgent to address the problem. HypAgent outperforms all baselines, yet most of its missed recoveries occur after a relevant variable has already been surfaced. Progress in this setting therefore depends less on generating more plausible proposals and more on deciding where to search and precisely characterizing how a discovered relationship behaves. We hope OpenHypBench and HypAgent serve as a foundation for studying Open-ended Hypothesis Discovery and enable broader discovery systems. Appendix A discusses the scope and limitations of this study.

AI USE STATEMENT

This work uses LLMs in HypAgent, several evaluated baselines, benchmark-mechanism construction, and mechanism-equivalence matching for recall. As described in Section 4.2, intervention planning and interventional support decisions are computed deterministically rather than delegated to an LLM. Generative AI tools were also used for code assistance, drafting, and language editing during implementation and manuscript preparation. The authors reviewed the AI-assisted material and take full responsibility for the final implementation, experimental results, claims, and manuscript.

ETHICS STATEMENT

OpenHypBench uses synthetic observations generated from structural causal models grounded in the schemas and empirical marginal distributions of publicly available datasets. This study does not recruit human participants, use private datasets, or perform interventions on real-world populations. The synthetic mechanisms should not be interpreted as established causal relationships in the source populations, and we do not claim a formal privacy guarantee from synthetic generation alone. Source datasets and their listed licenses are documented in Appendix C, Table 5. The principal risks are overstating autonomous discovery capabilities and treating simulated causal support as sufficient for real-world decisions. Accordingly, HypAgent produces candidate hypotheses for further investigation, not actionable scientific or policy recommendations; real-world use requires independent, domain-appropriate validation.

REPRODUCIBILITY STATEMENT

Benchmark construction is described in Section 4 and Appendix C; interventional evaluation and recall matching are described in Section 4.2 and Appendix D. Agent configurations and baseline implementations are provided in Appendices D.2 and D.3, respectively. We plan to release an anonymized package containing SCM specifications, planted mechanism sets, generation scripts and seeds, distribution parameters permitted for redistribution, the evaluator, agent and baseline implementations, prompt templates, run configurations and seeds, and the saved method outputs used for the reported results. These artifacts are intended to support regeneration of the 1M–10M-row tasks without distributing every generated table. For future evaluations, fresh discovery targets require re-instantiating mechanisms and their parameters rather than merely resampling observations from a published SCM. The anonymized repository link will be provided on the first page.

REFERENCES

Dhruv Agarwal, Bodhisattwa Prasad Majumder, Reece Adamson, Megha Chakravorty, Satvika Reddy Gavireddy, Aditya Parashar, Harshit Surana, Bhavana Dalvi Mishra, Andrew McCallum, Ashish Sabharwal, and Peter Clark. Autodiscovery: Open-ended scientific discovery via bayesian surprise. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=kJqTkj2HhF&noteId=wIKK6jEkkz.

Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pp. 207–216. Association for Computing Machinery, 1993. doi: 10.1145/170035.170072.

Ibrahim Aljarah. Students' academic performance dataset, 2016. URL https://www.kaggle.com/datasets/aljarah/xAPI-Edu-Data.

Barry Becker and Ronny Kohavi. Adult, 1996. URL https://archive.ics.uci.edu/dataset/2/adult.

Centers for Disease Control and Prevention. Behavioral risk factor surveillance system survey data, 2026. URL https://www.cdc.gov/brfss/annual_data/annual_data.htm.

Daqing Chen. Online retail ii, 2012. URL https://archive.ics.uci.edu/dataset/502/online+retail+ii.

Tingting Chen, Beibei Lin, Zifeng Yuan, Qiran Zou, Hongyu He, Anirudh Goyal, Yew-Soon Ong, and Dianbo Liu. Hypospace: A diagnostic benchmark for set-valued hypothesis generation under underdetermination and sublinear coverage bounds. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=QpjtK65JHO.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, Vishal Dey, Mingyi Xue, Frazier N. Baker, Benjamin Burns, Daniel Adu-Ampratwum, Xuhui Huang, Xia Ning, Song Gao, Yu Su, and Huan Sun. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=6z4YKr0GK6.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68, 2018. doi: 10.1111/ectj.12097.

David Maxwell Chickering. Optimal structure identification with greedy search. *J. Mach. Learn. Res.*, 3:507–554, 2002. URL https://api.semanticscholar.org/CorpusID:1191614.

John Clore, Krzysztof Cios, Jon DeShazo, and Beata Strack. Diabetes 130-us hospitals for years 1999–2008, 2014. URL https://archive.ics.uci.edu/dataset/296/diabetes+130-us+hospitals+for+years+1999-2008.

Paulo Cortez. Student performance, 2008. URL https://archive.ics.uci.edu/dataset/320/student+performance.

Credit Fusion and Will Cukierski. Give me some credit, 2011. URL https://www.kaggle.com/competitions/GiveMeSomeCredit.

Carlos García Ling, ElizabethHMGroup, FridaRim, inversion, Jaime Ferrando, Maggie, neuraloverflow, and xlsrln. H&m personalized fashion recommendations, 2022. URL https://www.kaggle.com/competitions/h-and-m-personalized-fashion-recommendations.

Nathan George. All lending club loan data, 2019. URL https://www.kaggle.com/datasets/wordsforthewise/lending-club.

Ken Gu, Ruoxi Shang, Ruien Jiang, Keying Kuang, Richard-John Lin, Donghe Lyu, Yue Mao, Youran Pan, Teng Wu, Jiaqian Yu, Yikun Zhang, Tianmai M. Zhang, Lanyi Zhu, Mike A Merrill, Jeffrey Heer, and Tim Althoff. BLADE: Benchmarking language model agents for data-driven science. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (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/.

Siyuan Guo, Cheng Deng, Ying Wen, Hechang Chen, Yi Chang, and Jun Wang. DS-agent: Automated data science by empowering large language models with case-based reasoning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16813–16848. PMLR, 2024.

Sirui Hong, Yizhang Lin, Bang Liu, Bangbang Liu, Binhao Wu, Ceyao Zhang, Danyang Li, Jiaqi Chen, Jiayi Zhang, Jinlin Wang, Li Zhang, Lingyao Zhang, Min Yang, Mingchen Zhuge, Taicheng Guo, Tuo Zhou, Wei Tao, Robert Tang, Xiangtao Lu, Xiawu Zheng, Xinbing Liang, Yaying Fei, Yuheng Cheng, Yongxin Ni, Zhibin Gou, Zongze Xu, Yuyu Luo, and Chenglin Wu. Data interpreter: An LLM agent for data science. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19796–19821, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1016. URL https://aclanthology.org/2025.findings-acl.1016/.

Peter Jansen, Marc-Alexandre Côté, Tushar Khot, Erin Bransom, Bhavana Dalvi Mishra, Bodhisattwa Prasad Majumder, Oyvind Tafjord, and Peter Clark. Discoveryworld: A virtual environment for developing and evaluating automated scientific discovery agents. *Advances in neural information processing systems*, 37:10088–10116, 2024.

Liqiang Jing, Zhehui Huang, Xiaoyang Wang, Wenlin Yao, Wenhao Yu, Kaixin Ma, Hongming Zhang, Xinya Du, and Dong Yu. DSBench: How far are data science agents from becoming data science experts? In *International Conference on Learning Representations*, 2025.

Ross D. King, Jem Rowland, Stephen G. Oliver, Michael Young, Wayne Aubrey, Emma Byrne, Maria Liakata, Magdalena Markham, Pinar Pir, Larisa N. Soldatova, Andrew Sparkes, Kenneth E. Whelan, and Amanda Clare. The automation of science. *Science*, 324(5923):85–89, 2009. doi: 10.1126/science.1165620. URL https://www.science.org/doi/abs/10.1126/science.1165620.

Jakub Kuzilek, Martin Hlosta, and Zdenek Zdrahal. Open university learning analytics dataset, 2015. URL https://archive.ics.uci.edu/dataset/349/open+university+learning+analytics+dataset.

Haokun Liu, Sicong Huang, Jingyu Hu, Yangqiaoyu Zhou, and Chenhao Tan. Hypobench: Towards systematic and principled benchmarking for hypothesis generation. *arXiv preprint arXiv:2504.11524*, 2025a.

Haokun Liu, Yangqiaoyu Zhou, Mingxuan Li, Chenfei Yuan, and Chenhao Tan. Literature meets data: A synergistic approach to hypothesis generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 245–281, Vienna, Austria, July 2025b. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.12. URL https://aclanthology.org/2025.acl-long.12/.

Yujie Liu, Zonglin Yang, Tong Xie, Jinjie Ni, Ben Gao, Yuqiang Li, Shixiang Tang, Wanli Ouyang, Erik Cambria, and Dongzhan Zhou. ResearchBench: Benchmarking LLMs in scientific discovery via inspiration-based task decomposition. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 13187–13207, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.644. URL https://aclanthology.org/2026.findings-acl.644/.

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. Discoverybench: Towards data-driven discovery with large language models. In *International Conference on Learning Representations*, 2025.

Stéphane G. Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993. doi: 10.1109/78.258082.

National Center for Health Statistics. National health and nutrition examination survey: Questionnaires, datasets, and related documentation, 2026. URL https://wwwn.cdc.gov/nchs/nhanes/.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2 edition, 2009. doi: 10.1017/CBO9780511803161.

Judea Pearl. The seven tools of causal inference, with reflections on machine learning. *Communications of the ACM*, 62(3):54–60, 2019.

Biqing Qi, Kaiyan Zhang, Kai Tian, Haoxiang Li, Zhang-Ren Chen, Sihang Zeng, Ermo Hua, Jinfang Hu, and Bowen Zhou. Large language models as biomedical hypothesis generators: A comprehensive evaluation. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=q36rpGlG9X.

Meg Risdal and Will Cukierski. Market basket analysis, 2017. URL https://www.kaggle.com/competitions/instacart-market-basket-analysis.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A. Lauffenburger, and Garry P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721): 523–529, 2005. doi: 10.1126/science.1105809. URL https://doi.org/10.1126/science.1105809.

Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009. doi: 10.1126/science.1165893. URL https://www.science.org/doi/abs/10.1126/science.1165893.

Marco Scutari. Learning bayesian networks with the bnlearn r package. *Journal of Statistical Software*, 35(3):1–22, 2010. doi: 10.18637/jss.v035.i03. URL https://doi.org/10.18637/jss.v035.i03.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT Press, Cambridge, MA, 1993.

U.S. Census Bureau. American community survey public use microdata sample (pums), 2026a. URL https://www.census.gov/programs-surveys/acs/microdata.html.

U.S. Census Bureau. Current population survey: Annual social and economic supplements, 2026b. URL https://www.census.gov/data/datasets/time-series/demo/cps/cps-asec.html.

Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242, 2018. doi: 10.1080/01621459.2017.1319839.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1601.

Zonglin Yang, Wanhao Liu, Ben Gao, Tong Xie, Yuqiang Li, Wanli Ouyang, Soujanya Poria, Erik Cambria, and Dongzhan Zhou. Moose-chem: Large language models for rediscovering unseen chemistry scientific hypotheses. In *International Conference on Learning Representations*, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

I-Cheng Yeh. Default of credit card clients, 2009. URL https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients.

Yangqiaoyu Zhou, Haokun Liu, Tejes Srivastava, Hongyuan Mei, and Chenhao Tan. Hypothesis generation with large language models. In *Proceedings of the 1st Workshop on NLP for Science (NLP4Science)*, pp. 117–139, Miami, FL, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.nlp4science-1.10. URL https://aclanthology.org/2024.nlp4science-1.10/.

A Limitations

This work focuses on controlled evaluation of open-ended discovery in large tabular datasets. OpenHypBench uses synthetic causal mechanisms grounded in real-world schemas, enabling precise interventional evaluation without treating the resulting mechanisms as verified relationships in the source domains. We study binary outcomes and seven mechanism families, while extensions to other outcomes, modalities, and forms of scientific explanation remain outside the present scope. To isolate differences between discovery procedures, all LLM-based methods use a common backbone. Cross-family evaluation is left to future work. Similarly, HypAgent performs observational discovery and is evaluated through interventions within the benchmark SCM, rather than claiming causal identification in real-world applications. Our evaluation measures interventional support and mechanism recovery under a finite search budget. Assessing downstream scientific utility with domain experts and constructing continually refreshed hidden mechanisms are important directions for future evaluation.

B Functional Form Definitions

Let T denote a treatment variable, M a moderator or subgroup variable, and Y the binary outcome. Thresholds and subgroup boundaries are fixed from the empirical quantiles of the corresponding task variables at generation time. Table 4 summarizes the seven structural forms represented in OpenHypBench.

Table 4: Functional forms in OpenHypBench. Counts are aggregated across all 15 tasks.

Form	Definition and interventional signature	Count
Monotone threshold	T changes Y after crossing a threshold, e.g., $\phi(T) = w\mathbb{I}[T > q]$.	30
Non-monotone	The effect is strongest at both tails or within an interior band; the resulting curve contains at least one direction change.	47
Derived feature	The treatment is a function of multiple variables, $U = g(T_1, T_2)$, such as a ratio, product, or difference.	50
Interaction reversal	The direction of the effect of T changes across values of a moderator variable M.	71
Subpopulation-specific	The effect is active only under a condition $c(M)$.	57
Simpson's reversal	The marginal association between T and Y has the opposite direction from the association within strata of M.	47
Mediation	The treatment affects the outcome through an intermediate variable, $T \to M \to Y$.	49
Total		**351**

C Benchmark Details

OpenHypBench contains 15 tasks spanning census, e-commerce, education, finance, and health. Each task borrows its variable vocabulary, types, semantics, and empirical marginal distributions from a public dataset. The causal graph, structural equations, planted mechanisms, and generated rows are constructed specifically for OpenHypBench. Table 5 reports the complete task inventory. Every outcome is binary; the outcome column describes the event represented by $Y = 1$.

Overall, the benchmark contains 53–65 visible features and 1M–10M rows per task. The three tasks in each domain use distinct source schemas rather than different samples from the same schema.

C.1 Hypothesis and SCM Construction

Candidate mechanisms were proposed using Claude Opus 4.8 through the Anthropic API. For each task, the model received the domain, the current schema, the designated outcome, and four few-shot examples. It did not receive observations from the generated task or the final SCM. Each curated mechanism was represented as $h = (X_h, \psi, d, c, m_h)$ and translated into an individual causal DAG. Here, X_h identifies the variables involved, ψ defines the structural form, d gives the direction, c specifies an optional population condition, and m_h is the absolute interventional change in outcome probability. If a given hypothesis can be cleanly represented as a tuple, we keep it. We treat two hypotheses as distinct when they involve different participating-variable sets X_h, different structural forms, or non-equivalent conditions. Differences in wording or effect-magnitude estimates alone do not constitute distinct mechanisms. These hypotheses are then converted to DAGs and composed into one task-level graph, which defines the SCM.

Variables may serve as background causes, confounders, moderators, mediators, causes, decoys, or null variables. Background variables are upstream causes without outcome terms. Confounders cause both a treatment and the outcome, moderators alter another mechanism through an interaction or gate, and mediators lie on a directed treatment–outcome path. Causes correspond to mechanisms in the answer key. Decoys correlate with causes through shared parents but have no outcome term, while null variables are independent.

Table 5: Task list of OpenHypBench.

Task ID	Domain	Source dataset	URL	License	Rows	Outcome ($Y = 1$)	Features	Hyp.
census_acs	Census	ACS PUMS (U.S. Census Bureau, 2026a)	link	U.S. public domain	1M	Housing cost burden	63	23
census_adult	Census	UCI Adult (Becker & Kohavi, 1996)	link	CC BY 4.0	5M	High income	63	25
census_cps	Census	CPS ASEC (U.S. Census Bureau, 2026b)	link	U.S. public domain	10M	Long commute	64	25
ecom_hm	E-commerce	H&M (García Ling et al., 2022)	link	Competition rules	5M	Customer churn	61	23
ecom_instacart	E-commerce	Instacart (Risdal & Cukierski, 2017)	link	Competition rules	1M	Product reordered	63	21
ecom_onlineretail	E-commerce	Online Retail II (Chen, 2012)	link	CC BY 4.0	10M	High lifetime value	62	24
edu_oulad	Education	OULAD (Kuzilek et al., 2015)	link	CC BY 4.0	1M	Course completion	58	22
edu_studentperf	Education	Student Performance (Cortez, 2008)	link	CC BY 4.0	5M	High exam performance	64	24
edu_xapi	Education	xAPI-Edu-Data (Aljarah, 2016)	link	CC BY-SA 4.0	10M	High engagement class	65	24
fin_givemecredit	Finance	Give Me Some Credit (Credit Fusion & Cukierski, 2011)	link	Competition rules	5M	Financial distress	58	25
fin_lendingclub	Finance	Lending Club (George, 2019)	link	CC0 listing	1M	High interest-rate tier	65	24
fin_taiwan	Finance	Taiwan Default (Yeh, 2009)	link	CC BY 4.0	10M	Payment default	59	21
health_brfss	Health	BRFSS (Centers for Disease Control and Prevention, 2026)	link	U.S. public domain	5M	Frequent poor mental health	65	21
health_diabetes130	Health	Diabetes 130-Hospitals (Clore et al., 2014)	link	CC BY 4.0	10M	Readmission within 30 days	53	24
health_nhanes	Health	NHANES (National Center for Health Statistics, 2026)	link	NCHS public-use data agreement	1M	Metabolic syndrome	63	25
						Total planted hypotheses		**351**

Before materialization, we reject specifications containing undefined parents, directed cycles, missing moderators, or numerical primitives applied to categorical variables. This guarantees a valid topological generation order for every task.

D Evaluation Protocol

Each proposal contains its natural-language claim, involved variables, claimed direction and magnitude, and an optional condition. The seven archetypes describe the mechanisms planted in OpenHypBench and hypothesis generators are not required to name these categories. LLM-based methods return the required fields as structured outputs, while fixed adapters convert statistical and causal baseline outputs into the same representation. We first parse and validate these fields deterministically. If parsing fails, Claude Opus 4.8 performs a fallback parse using fixed few-shot examples, temperature zero, and JSON output. Proposals that remain invalid are counted as unsupported. In our evaluations, none of the proposed hypotheses had any parsing issues.

Before evaluation, proposals are deduplicated using the mechanism-identity rule in Section 3, retaining the first proposal encountered for each identity. Precision is the fraction of these distinct proposals supported by intervention. For recall, supported proposals are first shortlisted against planted mechanisms using variable overlap, interventional response shape, condition, and magnitude. Claude Opus 4.8 then makes the final equivalence decision at temperature zero with JSON output. Multiple proposals matched to the same mechanism count once. A supported proposal that matches no planted mechanism contributes to precision but not recall, allowing valid hypotheses outside the planted set to receive credit.

D.1 HypAgent Implementation

D.1.1 Residualization

After every accepted hypothesis, residualization is updated using all variables named by the accepted hypotheses. Each continuous variable is expanded into indicators for its five empirical-quantile bins. Each categorical variable is one-hot encoded. Let $\Phi(S_k)$ denote the combined representation of the hypotheses accepted through round k. We fit an ordinary least-squares model to the original binary outcome:

$$\hat{f}_k = \arg\min_f \|Y - f(\Phi(S_k))\|_2^2, \qquad \widetilde{Y}_k = Y - \hat{f}_k(\Phi(S_k)). \tag{1}$$

The fit uses the full dataset and is jointly recomputed after every acceptance. Overlapping hypotheses are therefore not subtracted independently. Their shared explanatory contribution is resolved by the joint fit. Rejected hypotheses remain in memory but do not enter $\Phi(S_k)$.

D.1.2 Proposal and Triage

At each round, the proposer generates six broad candidate directions for explaining the current residual based on a randomly queried slice of data. Its context contains the dataset schema, outcome name, current residual, accepted and rejected hypothesis memories, and results returned by previous analysis calls. The context is assembled from these elements at each round rather than retaining an unrestricted conversation history. The LLM accesses data by writing analysis code and receives the

resulting statistics or summaries rather than the complete table in its context. Triage evaluates each candidate through a candidate-specific query of 1,000 observations. The triage score is

$$s(T) = \frac{\mathrm{Var}\Big(\mathbb{E}[\widetilde{Y}_k \mid \mathrm{bin}(T)]\Big)}{\mathrm{Var}(\widetilde{Y}_k)}. \tag{2}$$

This score measures the fraction of residual variation associated with the candidate. Candidates with $s(T) < 0.02$ are discarded. The highest-scoring remaining direction is passed to refinement.

If none of the six candidates clears the threshold, the search terminates. Residual fitting and refinement use the full dataset.

D.1.3 Refinement

Refinement receives the selected direction, dataset schema, outcome, full observational dataset, and memories of previously accepted and rejected hypotheses. It may execute arbitrary analyses for up to 10 code turns. The LLM is not given a fixed menu of functional forms. It determines how to formulate the relationship from its analysis and may select thresholds, ranges, moderators, treatment values, conditions, and derived features $U = g(T_1, T_2)$.

Derived features are introduced during refinement after triage has selected an initial direction. The LLM chooses both the constituent variables and the function g from its data analysis. Finally, HypAgent examines the 15 variables most correlated with the selected candidate. Continuous moderators retain their numerical values and categorical moderators are one-hot encoded. A moderator is retained when its interaction coefficient satisfies $p < 0.01$ and its standardized magnitude exceeds the fixed hyperparameter 0.005.

The final natural-language hypothesis explicitly states its involved variables, direction d, magnitude m_h, and any condition c. It does not return a separate benchmark functional-form label. The relationship described in the hypothesis text is mapped to the benchmark representation only during evaluation.

Direction and magnitude are estimated from the original binary outcome rather than from the residual used during triage. The LLM selects the treatment values according to its analysis. The claimed magnitude is

$$m_h = \left|\widehat{\Pr}(Y = 1 \mid T = t_{\mathrm{hi}}, c) - \widehat{\Pr}(Y = 1 \mid T = t_{\mathrm{lo}}, c)\right|. \tag{3}$$

When c is absent, the contrast is calculated over the full population.

Before acceptance, the claimed direction is recomputed from the data and must agree with the direction stated in the hypothesis:

$$\mathrm{sign}(\hat{\tau}_{\mathrm{data}}) = \mathrm{sign}(\hat{\tau}_{\mathrm{claim}}). \tag{4}$$

For $d = \mathrm{mixed}$, this agreement is checked separately within each region or condition stated by the hypothesis rather than through one aggregate sign. The effect must also survive covariate adjustment. We select the 12 covariates most correlated with the proposed treatment and include them jointly in an ordinary least-squares regression of the current residual $\widetilde{Y}_k$ on the treatment and selected covariates. Let $\hat{\beta}_T^{\mathrm{adj}}$ denote the adjusted treatment coefficient. We convert it to an interdecile effect using

$$\Delta_T = q_{0.9}(T) - q_{0.1}(T), \qquad \hat{\tau}_{\mathrm{adj}} = \hat{\beta}_T^{\mathrm{adj}} \Delta_T. \tag{5}$$

For a conditional hypothesis, this calculation is performed on the subset satisfying its stated condition. A hypothesis is accepted when its direction agrees with the recomputed direction and

$$|\hat{\tau}_{\mathrm{adj}}| > 0.02. \tag{6}$$

Because Y is binary, this threshold corresponds to an adjusted two-percentage-point outcome difference over the treatment's interdecile range.

Every refinement attempt consumes one unit of the budget $B = 50$, regardless of whether the resulting hypothesis is accepted. A rejected complete claim is stored in memory and is not retried. Exclusion applies to the complete claim rather than its variables. This allows subsequent hypotheses to reuse a variable under a different relationship or condition.

Algorithm 1 HypAgent: Open-ended Hypothesis Discovery through iterative residual search.

Require: Dataset D, outcome Y, refinement budget B, triage threshold ϵ
Ensure: Accepted hypothesis set S

$S \leftarrow \emptyset$ ▷ accepted hypotheses
$S_{\text{rej}} \leftarrow \emptyset$ ▷ rejected complete claims
while budget B remains **do**
  $\widetilde{Y} \leftarrow \textsc{Residualize}(D, Y, S)$ ▷ remove variation represented by accepted hypotheses
  $\mathcal{C} \leftarrow \textsc{Propose}(\text{schema}(D), \widetilde{Y}, S, S_{\text{rej}})$ ▷ propose six candidate directions
  $T^\star \leftarrow \textsc{Triage}(D, \widetilde{Y}, \mathcal{C}, \epsilon)$ ▷ select using residual signal
  if $T^\star = \varnothing$ **then**
    break ▷ no candidate clears triage
  end if
  $\hat{h} \leftarrow \textsc{Refine}(D, \widetilde{Y}, T^\star)$ ▷ formulate and validate the claim
  $\textsc{ConsumeBudget}(B)$ ▷ accepted and rejected attempts count
  if $\hat{h}$ passes the acceptance checks **then**
    $S \leftarrow S \cup \{\hat{h}\}$ ▷ update the residual in the next round
  else
    $S_{\text{rej}} \leftarrow S_{\text{rej}} \cup \{\hat{h}\}$ ▷ do not retry the same complete claim
  end if
end while
return S

Although the budget permits up to 50 refinement attempts, it did not bind in the reported runs. Runs terminated after 20–36 refinement attempts when no candidate in the current proposal batch cleared the triage threshold.

D.2 HypAgent Configuration

All evaluated LLM components use Claude Sonnet 4.6. HypAgent proposes six candidate directions per round at temperature 0.7, scans moderators at temperature 0.6, and validates at temperature 0.5. It uses five quantile bins for continuous variables and observed categories for categorical variables. The triage threshold is $\epsilon = 0.02$. For confounding checks in refinement, HypAgent examines the 12 observed covariates most strongly associated with the candidate treatment. A proposal is rejected when its absolute interdecile-scaled adjusted effect is at most 0.02. Runs terminate when no candidate clears the triage threshold. In the reported runs, HypAgent used 20–36 iterations, with a mean of 26.3.

D.3 Baseline Implementations

We use the same method order throughout the paper. **Correlation Mining** ranks features by point-biserial correlation with the outcome and returns the leading features as monotone hypotheses in the observed direction. **Subgroup Analysis** searches single-feature subgroups and ranks them by outcome lift. **PC** applies conditional-independence tests and returns edges adjacent to the outcome as candidate causal relationships. **XGBoost+LLM** fits a gradient-boosted classifier and supplies its ranked feature importances, variable-distribution summaries, and 1,000 sampled rows to the LLM, which formulates complete hypotheses. Following the example-driven setup of Zhou et al. (2024), **LLM (no code)** receives the domain, schema, outcome, variable-distribution summaries, and 1,000 sampled rows, but cannot execute code or request additional data. **ReAct** interleaves reasoning with code execution. **Claude Code** may write and execute analysis code over the complete table. **AutoDiscovery** proposes and tests discoveries using its original iterative procedure. All returned hypotheses are processed by the shared parser and scored by the same interventional evaluator. All LLM-based methods use Claude Sonnet 4.6 as the backbone.

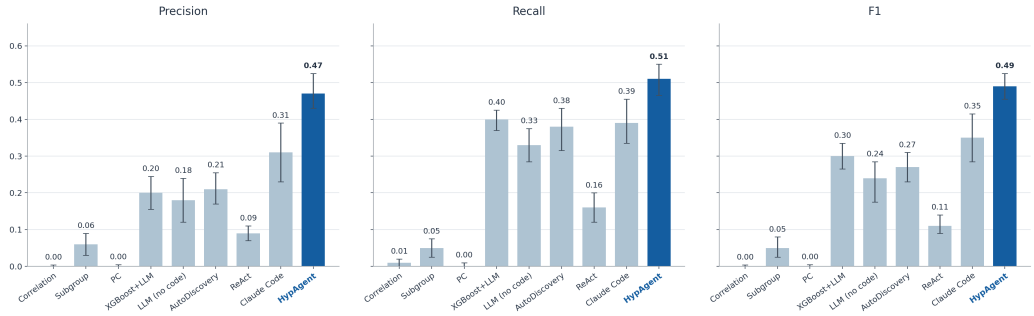


Figure 6: Precision, recall, and F_1 at each method's reported operating point. Whiskers show 95% cluster-bootstrap confidence intervals over tasks.

D.4 Failure Analysis

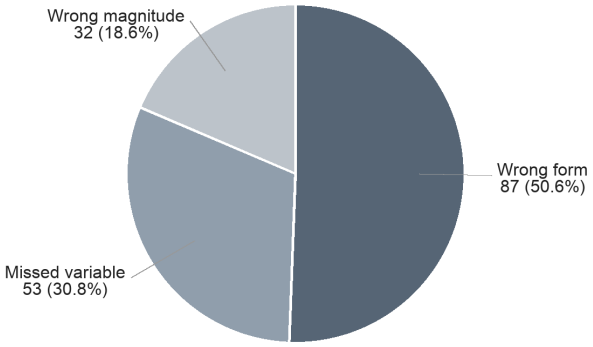


Figure 7: Failure modes among the 172 planted hypotheses missed by HypAgent.

E Prompt Templates

We report the prompts used to construct candidate benchmark mechanisms and for the two LLM-driven stages of HypAgent. Triage, confounder adjustment, and final validity checks are implemented programmatically and are not delegated to the LLM.

E.1 Benchmark Candidate Construction

The following template is instantiated separately for each target functional form. We request at least three candidates per form and retain only mechanisms that pass the structural checks described in Appendix C.1.

```
SYSTEM:
You are an expert in causal modeling and synthetic benchmark
construction. Given a domain, dataset schema, outcome, and target
functional form, propose plausible causal hypotheses that can be
instantiated in a structural causal model. Return JSON only.

USER:
Domain:
{domain}
```

```
Dataset schema:
{schema}

Designated outcome:
{outcome}

Target functional form:
{functional_form}

Definition of the target functional form:
{functional_form_definition}

Examples:
{four_few_shot_examples}

Generate {k} distinct causal hypotheses that instantiate the specified
functional form. Every generated hypothesis must use that form.

Represent each hypothesis as h = (X_h, psi, d, c, m_h), where X_h
contains the variables involved, psi is the specified functional form,
d is the effect direction, c is an optional condition or subpopulation,
and m_h is the expected effect magnitude under the relevant
intervention contrast.

Each hypothesis must:
- use only variables in the supplied schema;
- state an interventional causal claim rather than an association;
- identify the treatment and any moderator, mediator, stratum, or
  component variables required by the functional form;
- be precise enough to compile into a causal DAG and structural
  equation;
- be plausible in the supplied domain and substantively distinct from
  the other candidates;
- not use the outcome as an input variable; and
- not claim to have observed data patterns or statistics, because no
  dataset rows are provided.

Return a JSON list:
[
  {
    "X_h": ["<treatment>", "<optional additional variables>"],
    "psi": "{functional_form}",
    "relationship": "<precise causal relationship>",
    "d": "positive | negative | mixed",
    "c": "<condition or null>",
    "m_h": "<expected effect magnitude>",
    "causal_roles": {
      "<variable>": "treatment | moderator | mediator | stratum |
                     component"
    },
    "hypothesis": "<concise natural-language causal claim>"
  }
]
```

E.2 HypAgent Prompts

Candidate Proposal

```
SYSTEM:
You are a data scientist proposing directions to investigate.
Return JSON only.

USER:
Dataset schema:
{schema}

Outcome:
{outcome}

Random sample of 1,000 rows:
{sample_rows}

Summary of the current residual:
{residual_summary}

Hypotheses already confirmed:
{confirmed_hypotheses}

Hypotheses previously rejected:
```

```
{rejected_hypotheses}

Results from previous analysis calls:
{previous_analysis_results}

Propose {k} distinct and promising new directions that could explain
variation in the outcome.

Each proposal should identify a primary variable to investigate and,
when relevant, other variables that may participate in the
relationship. Prioritize directions that are not already explained by
the confirmed hypotheses. Do not restate or superficially reformulate
a confirmed hypothesis.

At this stage, propose broad directions rather than complete
hypotheses. Do not assume a predefined functional form. A later
refinement step will determine the specific relationship, conditions,
direction, and effect magnitude.

Return a JSON list:
[
  {
    "variable": "<primary column>",
    "partners": ["<optional related column>", "..."],
    "rationale": "<brief reason this direction may explain the outcome>"
  }
]
```

Candidate Refinement

```
SYSTEM:
You are a data scientist refining a promising direction into a
precise, testable hypothesis. Explore the data by writing Python.
Use <code>...</code> to run code and return the final result inside
<hypothesis>...</hypothesis> as JSON. Do not propose unrelated
directions.

USER:
Dataset schema:
{schema}

Outcome:
{outcome}

HYPOTHESES ALREADY CONFIRMED:
{confirmed_hypotheses}

HYPOTHESES PREVIOUSLY REJECTED:
{rejected_hypotheses}

DIRECTION SELECTED BY TRIAGE:
{candidate}

In your Python namespace you have:
- `df`: the full {n_rows:,} rows, containing all columns and the raw
  outcome.
- `resid`: a NumPy array aligned with `df`. It contains the outcome
  after the effects of all confirmed hypotheses have been removed
  and therefore represents the remaining unexplained signal.

Your job is to refine the selected direction into ONE precise
hypothesis. Focus on the selected primary variable and determine:
- which variables are involved;
- how they are related to the remaining outcome;
- whether the relationship depends on a range, condition, partner
  variable, or subpopulation;
- the direction of the relationship under the identified
  conditions; and
- its estimated magnitude.

Use code to inspect the selected variable across its distribution or
categories. Investigate partner variables or conditions only when
supported by the data. Compare the relevant groups or regions using
the full dataset before producing the final estimate.

Do not assume a predefined functional form. Let the observed
relationship determine the hypothesis. Do not switch to an unrelated
primary variable. Confounder adjustment and final validity checks are
performed separately after this step.
```

```
If the selected direction does not support a sufficiently specific
hypothesis, return:
<hypothesis>{"supported": false}</hypothesis>

Otherwise, return:
<hypothesis>
{
  "supported": true,
  "X_h": ["<primary variable>", "<other involved variables>"],
  "psi": "<concise description of the observed relationship>",
  "d": "<positive, negative, or mixed>",
  "c": "<condition, range, or subpopulation; null if unconditional>",
  "m_h": <estimated magnitude>,
  "text": "<one-sentence testable hypothesis>"
}
</hypothesis>

You have {max_turns} code turns.
```

LLM (no code)

```
SYSTEM:
You are a data scientist generating testable hypotheses from
observed data. Return JSON only.

USER:
Domain: {domain}
Dataset description: {scenario}
Outcome: {outcome}

Schema:
{schema}

Variable distributions:
{distribution_summaries}

Random sample of 1,000 rows:
{sample_rows}

Using the observed examples and distribution summaries, generate
{k} distinct, testable hypotheses about variables that may affect
the outcome. You cannot execute code or request additional data.

Each hypothesis must specify a natural-language claim, all variables
involved, the claimed direction, the claimed absolute change in
outcome probability, and an optional condition or population.
Do not assign benchmark archetype names. Avoid duplicate or
paraphrased claims.

Return:
[
  {
    "text": "<one-sentence testable hypothesis>",
    "variables": ["<variable>", "..."],
    "direction": "<positive, negative, or mixed>",
    "magnitude": <absolute probability change between 0 and 1>,
    "condition": "<condition or population, or null>"
  }
]
```

XGBoost+LLM

```
SYSTEM:
You are a data scientist converting predictive evidence into
testable hypotheses. Return JSON only.

USER:
Domain: {domain}
Dataset description: {scenario}
Outcome: {outcome}

Schema:
{schema}

Variable distributions:
{distribution_summaries}

Random sample of 1,000 rows:
{sample_rows}
```

```
Features ranked by XGBoost gain importance:
{ranked_features}

Using the observed examples, distribution summaries, and feature
ranking, generate {k} distinct, testable hypotheses about variables
that may affect the outcome. Feature importance is predictive rather
than causal evidence. You cannot execute code or request additional
data.

Each hypothesis must specify a natural-language claim, all variables
involved, the claimed direction, the claimed absolute change in
outcome probability, and an optional condition or population.
Do not assign benchmark archetype names. Avoid duplicate or
paraphrased claims.

Return:
[
  {
    "text": "<one-sentence testable hypothesis>",
    "variables": ["<variable>", "..."],
    "direction": "<positive, negative, or mixed>",
    "magnitude": <absolute probability change between 0 and 1>,
    "condition": "<condition or population, or null>"
  }
]
```