WHEN RULES ARE HIDDEN: A PAIRED EVALUATION OF ABDUCTIVE REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) are increasingly used for scientific discovery, where they must not only reason from known rules but also infer hypotheses and latent mechanisms from observations. Abductive reasoning is therefore a central capability, yet its evaluation faces two challenges. First, abduction is confounded by deduction: an incorrect prediction may arise from inferring an inadequate hypothesis or from reasoning incorrectly from it. Second, finite observations can admit multiple plausible hypotheses, making evaluation against a single ground-truth explanation problematic. We introduce RuleCast, a paired benchmark that addresses both challenges. Each problem is posed under Complete-Rule and Hidden-Rule conditions that differ only in whether the latent rules required for prediction are provided or must be inferred from evidence, so their gap characterizes the additional difficulty of inferring missing premises. Problems are built on controlled transition-system worlds in three scientific domains, where multiple evidence-consistent hypotheses may exist but agree on every evaluated forecast, removing the need for a unique ground-truth hypothesis. Across open-weight and proprietary reasoning models, forecast accuracy consistently drops when rules must be inferred, even for models that execute the given rules nearly perfectly. An error analysis over the enumerable hypothesis space locates the bottleneck in verifying candidate hypotheses against the evidence rather than in executing a committed hypothesis.

1 INTRODUCTION

Computation begins with rules. Scientific discovery often begins without them. From early models of computation to modern artificial intelligence, a central ambition has been to solve increasingly complex problems with less human specification. Large language models (LLMs) have carried this ambition into science, moving from applying established knowledge toward participating in discovery: they are used to generate hypotheses, design experiments, interpret observations, and discover latent mechanisms or laws from data (Lu et al., 2024; Jansen et al., 2024; Woodruff et al., 2026; Wiemann et al., 2026). These tasks demand different forms of reasoning: some require deriving consequences from known rules, whereas others require identifying regularities in observations or inferring the rules that could explain them (He et al., 2025b; Sheng et al., 2025).

Among these capabilities, abductive reasoning is particularly central to scientific discovery (King et al., 2004): scientists are rarely given all of the rules governing a phenomenon and must instead infer plausible hypotheses that connect and explain the available evidence (Kakas et al., 1992; Eiter & Gottlob, 1995). Yet evaluating this capability presents two fundamental challenges. First, **deductive confounding** arises because an abductively inferred hypothesis is typically assessed through its downstream consequences (Marx, 1967). Given evidence E, a model must first infer a hypothesis H and then reason forward from H to obtain a prediction Y:

$$E \xrightarrow{\text{Abduction}} H \xrightarrow{\text{Deduction}} Y. \tag{1}$$

Consequently, an incorrect prediction does not by itself reveal whether the model failed to infer an adequate hypothesis or instead failed to reason correctly from it; analogous conflations have been documented in inductive rule learning (Qiu et al., 2024; Cheng et al., 2024; Li et al., 2025). Second, **abductive underdetermination** arises because multiple hypotheses may be consistent with the same set of observations (Eiter & Gottlob, 1995; He et al., 2025a; Sun & Saparov, 2025). Requiring a

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.

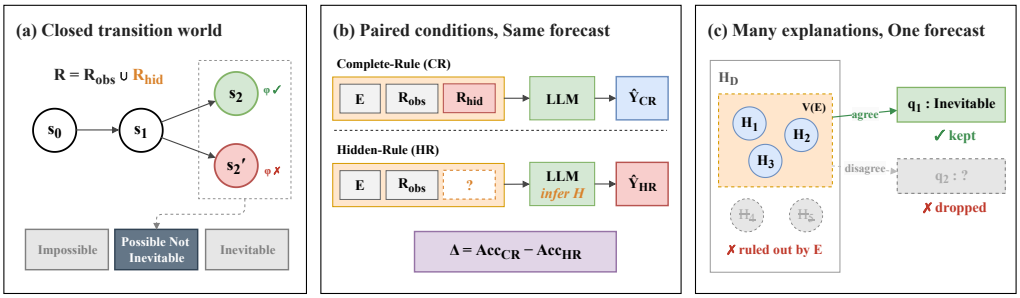

Figure 1: **Overview of the paired evaluation.** (a) Each query asks whether φ holds on none, some, or every reachable outcome of a closed transition world. (b) CR and HR share the same world, evidence, and query and differ only in whether R_{hid} is given. (c) We keep only queries on which every hypothesis consistent with E agrees, so each item has one correct forecast even when its explanation is not unique.

model to recover a single ground-truth hypothesis may therefore penalize alternative explanations that are equally compatible with the available evidence.

We address these challenges with two corresponding design choices. First, we introduce a **paired evaluation** that constructs two matched conditions for each problem, differing only in whether the latent rules required to explain the evidence are provided. In the Complete-Rule (CR) condition, the model receives the evidence E, the observed rules R_{obs}, and the hidden rules R_{hid}, and forecasts directly by deduction. In the Hidden-Rule (HR) condition, it receives the same evidence and observed rules but is told only that additional hidden rules exist, so it must first infer an explanatory hypothesis before forecasting (Figure 1 (b)). Because the two conditions share the same world, evidence, and forecast queries, CR controls for the model's ability to reason forward once the rules are available, and the CR–HR gap characterizes the additional difficulty of inferring the premises required for deduction. Second, to address abductive underdetermination, we construct **controlled transition-system worlds** that are closed yet combinatorially rich, in which each query asks whether a proposition holds in none, some but not all, or all reachable outcomes of an experiment (Figure 1 (a)). Because observed rules, hidden rules, evidence, and forecast queries can be precisely manipulated, the full hypothesis space can be enumerated. We construct each problem so that multiple hypotheses may remain consistent with the evidence, but all of them agree on every retained forecast query (Figure 1 (c)). The model therefore need not identify the particular hidden rule used to generate the world; it only needs to infer an explanation sufficient to determine the forecast. This allows us to evaluate abductive reasoning without assuming that the true explanation is uniquely identifiable.

Across open-weight and proprietary reasoning models, every model forecasts less accurately under HR, even when it executes the given rules almost perfectly under CR. The gap widens with the depth of the observed experiments and narrows as more of the hypothesis space remains consistent with the evidence. An error analysis against the full hypothesis space locates the bottleneck in verifying candidates against the evidence: models both adopt hypotheses the evidence has already ruled out and reject correct ones after calculation errors.

Our main contributions are as follows:

- **Paired evaluation of abductive reasoning.** We introduce a paired evaluation that compares matched forecasting problems with and without the rules required for prediction, separating the difficulty of inferring missing premises from that of reasoning once they are available.
- **A controlled benchmark for abduction under underdetermination.** We construct controlled transition-system worlds in three scientific domains, in which multiple hypotheses may remain consistent with the evidence but all of them agree on every retained forecast query, so no uniquely identifiable ground-truth explanation is required.
- **Empirical characterization of the abductive gap.** Across a range of reasoning models, inferring the rules consistently lowers forecast accuracy, even for models that execute the given rules nearly perfectly; an error analysis locates the bottleneck in verifying candidate hypotheses against the evidence.

Table 1: **Comparison with related benchmarks.** *Verifiable Outputs*: correctness is checked formally, not by human or LLM judgment. *Alternatives Accepted*: consistent non-generating hypotheses are credited (△: only via extra per-hypothesis evaluation). *Experimental Evidence*: evidence comes from experiment outcomes. *Held-out Prediction*: hypotheses are tested on unseen instances, not only matched to a reference. *Matched Control*: paired with a rule-given condition.

Benchmark	Verifiable Outputs	Alternatives Accepted	Experimental Evidence	Held-out Prediction	Matched Control
ART-αNLG (Bhagavatula et al., 2020)	✗	△	✗	✗	✗
AbductionRules (Young et al., 2022)	✓	✗	✗	✗	✗
RULEARN (He et al., 2025b)	✓	✗	✓	✗	✗
GEAR (He et al., 2025a)	✓	△	✗	✗	✗
INABHYD (Sun & Saparov, 2025)	✓	△	✗	✗	✗
Elenchos (Steiglechner et al., 2026)	✓	✗	✓	✗	✗
DiscoveryWorld (Jansen et al., 2024)	✓	✗	✓	✓	✗
DiscoveryPhysics (Wiemann et al., 2026)	✓	△	✓	✓	✗
HypoArena (Zhong et al., 2026)	✗	△	✗	✗	✗
RuleCast (Ours)	✓	✓	✓	✓	✓

2 Related Work

Abductive Reasoning Benchmarks and Evaluation. Existing abductive reasoning benchmarks differ in the source of their evidence and how they judge a hypothesis. Evidence is given as natural-language narratives, as in ART (Bhagavatula et al., 2020) and SemEval-2026 AER (Cao et al., 2026); as facts and rules in a synthetic formal theory, as in AbductionRules (Young et al., 2022), INABHYD (Sun & Saparov, 2025), and DeFAb (Cooper & Velasquez, 2026); or as outcomes of queries to an executable system, as in RULEARN (He et al., 2025b) and Elenchos (Steiglechner et al., 2026). Correctness is judged either against a single reference, which rejects valid alternatives, or through additional per-hypothesis evaluation: human judgment in ART, proof or verifier checks in INABHYD and DeFAb, and execution against the observations in GEAR (He et al., 2025a). None isolates hypothesis formation from prediction. Our benchmark (Table 1) draws evidence from experiment outcomes, pairs each problem with a rule-given control, and scores only forecasts shared by all evidence-consistent hypotheses, crediting any consistent explanation at no additional cost.

Scientific Discovery Benchmarks. Recent benchmarks evaluate LLMs on AI-driven scientific discovery, asking models to infer hypotheses from literature (Yang et al., 2025), observational data (Majumder et al., 2025), or conclusion-free contexts (Zhong et al., 2026). Simulated environments further let agents design experiments and test inferred regularities or physical laws on unseen instances (Jansen et al., 2024; Wiemann et al., 2026). Generated hypotheses are typically assessed by comparison with reference hypotheses or explanations, often using LLM judges, or by relative preference among models (Zhong et al., 2026). In contrast, RuleCast pairs each problem with a rule-given control and evaluates only forecasts shared by all evidence-consistent hypotheses, separating abductive inference from deduction without assuming a unique explanation (Table 1).

Formal abduction. Logic-based abduction seeks hypotheses from a specified set of admissible hypotheses that, together with a background theory, explain an observation (Kakas et al., 1992; Eiter & Gottlob, 1995). Because multiple explanations may exist, prior work has studied preference criteria such as subset and cardinality minimality, together with the complexity of the associated reasoning problems (Bylander et al., 1991; Eiter & Gottlob, 1995). We likewise fix an explicit hypothesis space, but instead of selecting a single explanation, we evaluate forecasts on which all evidence-consistent hypotheses agree, so the explanation need not be uniquely identifiable.

Transition System A transition system is a general formalism for representing the dynamics of a system through a set of states and a transition relation specifying how the system may evolve from one state to another (Baier & Katoen, 2008). Because of this domain-agnostic abstraction, transition systems have been widely used to model diverse systems, including computer programs (Baier & Katoen, 2008), concurrent and reactive systems (Aceto et al., 2007), communication protocols (Ter Beek et al., 2007), and discrete scientific processes (Bugenhagen & Beard, 2012). More recently, transition systems and related finite-state models have also been used to construct synthetic reasoning bench-

marks for LLMs. TempoBench (Holzer et al., 2025) uses finite-state automata to evaluate temporal execution and causal reasoning, while LLM-FSM (Wu et al., 2026) procedurally generates finite-state machines with configurable transition structures to synthesize verifiable reasoning problems. In our benchmark, we use finite transition systems as a common abstraction for constructing controlled scientific worlds, enabling exact enumeration of admissible hypotheses and their resulting forecasts.

3 Evaluation Framework

3.1 Task Formulation

We formalize abductive forecasting as the problem of predicting the consequences of partially unknown governing rules from observed experiments. To this end, we consider a public design D specifying a closed-world scientific domain and its admissible hypotheses. This design determines a hypothesis space $\mathcal{H}_D$, and the model observes experimental evidence E before predicting outcomes on new experimental queries. We describe the concrete domains and world construction in Section 4. Let $\mathcal{Q}$ denote the family of admissible experimental queries. A query $q = (x, \varphi)$ consists of an experiment x and a proposition φ on final states. For an admissible hypothesis H, we write $f_q(H) = F_\varphi(H, x)$ for its exact forecast outcome. Let $Q \subseteq \mathcal{Q}$ be the finite, nonempty set of queries used for evaluation. Our evaluation focuses on the predictive consequences of the inferred knowledge on these queries. We adopt forecasting as the primary target for two reasons. First, it captures whether knowledge inferred from experimental evidence is sufficient to predict previously unseen outcomes. Second, a common forecast target enables a matched comparison between settings in which the underlying rules are provided and those in which they must be inferred, as introduced in Section 3.3.

3.2 Forecasting under Hypothesis Underdetermination

A central challenge in evaluating abductive reasoning is hypothesis underdetermination: the observed evidence may be consistent with multiple admissible hypotheses. If such hypotheses imply different outcomes for a novel experiment, the evidence alone does not uniquely determine which forecast should be considered correct. Consequently, evaluating a single intended latent theory as the unique answer may penalize alternative hypotheses that are equally consistent with the available evidence.

To formalize this issue, we fix the admissible hypothesis space $\mathcal{H}_D$ independently of the observed evidence and model responses. Given evidence $E = ((q_i, y_i))_{i=1}^n$, we define its version space (Mitchell, 1977) as

$$\mathcal{V}_D(E) = \{H \in \mathcal{H}_D : f_{q_i}(H) = y_i,\ \forall i\}\,. \tag{2}$$

As observations accumulate, the version space can only shrink. For a finite hypothesis space, a finite set of observations identifies a unique hypothesis whenever the available queries distinguish it from every alternative (Proposition A.2). We assume $\mathcal{V}_D(E) \neq \varnothing$, but do not require it to contain a unique hypothesis. Fix any $H^* \in \mathcal{V}_D(E)$. As shown in Theorem A.1, the version space is exactly the observation-equivalence class of H^* on the evidence. Thus, multiple admissible theories may remain observationally consistent with the same evidence.

We require *forecast invariance*: all hypotheses in $\mathcal{V}_D(E)$ must agree on every evaluated query.

Theorem 3.1 (Reference-independent forecast scoring)**.** *Let $\mathcal{V}_D(E) \neq \varnothing$ and let $Q \subseteq \mathcal{Q}$ be finite and nonempty. For a submitted forecast vector $\widehat{y} = (\widehat{y}_q)_{q\in Q}$ assigning one modal label to each query, define*

$$\mathrm{Acc}_H(\widehat{y}) = \frac{1}{|Q|}\sum_{q\in Q} \mathbf{1}\{\widehat{y}_q = f_q(H)\}.$$

Forecast invariance on Q holds if and only if, for every submitted vector $\widehat{y}$, $\mathrm{Acc}_H(\widehat{y})$ is independent of the reference hypothesis $H \in \mathcal{V}_D(E)$.

The full characterization and proof are given in Theorem A.3. Section 4.4 describes how this condition is enforced.

3.3 Paired Evaluation under Complete and Hidden Rules

To compare these two settings, we evaluate each instance under two matched conditions. In the Complete-Rule (CR) condition, the model is given both the observed rules R_{obs} and the hidden rules R_{hid}:

$$\widehat{y}_{\text{CR}} = M(E, R_{\text{obs}}, R_{\text{hid}}, Q). \tag{3}$$

In the Hidden-Rule (HR) condition, R_{hid} is withheld:

$$\widehat{y}_{\text{HR}} = M(E, R_{\text{obs}}, Q). \tag{4}$$

requiring the model to infer sufficient information about the missing rules from the experimental evidence before forecasting the outcome.

The two conditions share the same world, evidence, forecast queries, and target outcomes, differing only in whether the hidden rules are provided. By construction, forecast-invariant instances have the same exact target vector in both conditions (Corollary 4.1). CR therefore provides a matched rule-given control for HR, and Section 3.4 defines the corresponding performance comparison.

3.4 Paired Evaluation Metrics

We measure performance primarily by forecast accuracy, defined as the fraction of forecast queries answered correctly. Let Acc_{CR} and Acc_{HR} denote accuracy under the Complete-Rule and Hidden-Rule conditions, respectively. We define the CR–HR gap as $\Delta_{\text{CR}-\text{HR}} = \text{Acc}_{\text{CR}} - \text{Acc}_{\text{HR}}$.

Because CR and HR differ only in whether the hidden rules are provided, this gap characterizes the additional difficulty introduced when the rules required for forecasting must first be inferred from experimental evidence. We do not interpret $\Delta_{\text{CR}-\text{HR}}$ as a stagewise decomposition of abductive error: HR requires both inference of the missing rules and subsequent reasoning over them, and errors in these stages may interact (see Appendix A.3).

4 Benchmark Construction

4.1 Synthetic Scientific Domains

We instantiate our framework in three synthetic scientific domains: physics, chemistry, and biology (Figure 2). All three share a common abstraction: a world state is a finite collection of discrete variables whose evolution is governed by a finite set of condition–effect rules. Each domain simplifies characteristic processes in its field while keeping the state and rule spaces finite enough for exhaustive analysis. The **physics** domain, following finite symbolic abstractions of continuous and hybrid systems (Girard et al., 2008; Kloetzer & Belta, 2008), represents a system with bounded integer quantities and binary modes, and rules conditionally change quantities or switch modes. The **chemistry** domain, inspired by multi-compartment vesicles (Elani et al., 2014), represents discrete amounts of species across compartments, and rules describe reactions within a compartment, optionally requiring a catalyst or blocked by an inhibitor, and transport between compartments. The **biology** domain, following qualitative logical models of gene regulatory networks (Abou-Jaoudé et al., 2016; Karlebach & Shamir, 2008), represents components with discrete activity levels and regulatory scores, and rules let the level of one component shift the score of the next. To prevent reliance on prior domain knowledge, all entities have invented or semantically neutral names (e.g., species "quorene"), so forecasts must be derived from the stated rules and evidence alone.

4.2 Rules and World Dynamics

Each domain defines a finite grammar that specifies which condition–effect rules are admissible. A world is governed by a set of such rules, $R = R_{\text{obs}} \cup R_{\text{hid}}$, where the observed rules R_{obs} are given in both evaluation conditions and the hidden rules R_{hid} are withheld in the Hidden-Rule condition. Because both the state space and the grammar are finite, we can enumerate all admissible rules and compute exactly the transition dynamics that any rule set induces. This lets us discard worlds containing ineffective rules: we retain only rule sets in which every observed rule changes the state

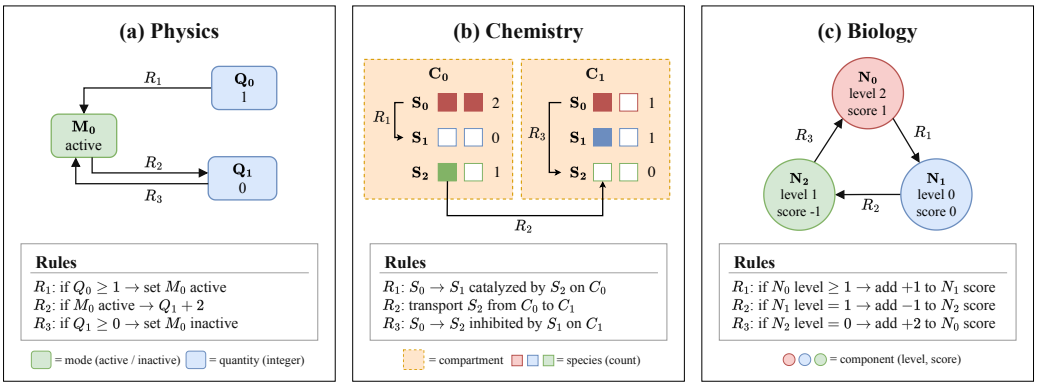

Figure 2: **Three synthetic scientific domains.** Each panel shows an example state and its condition–effect rules. (a) Physics: integer quantities and binary modes. (b) Chemistry: species amounts within compartments. (c) Biology: components with an activity level and a regulatory score, where the sign of a score sets the component's target level. Worked transitions are given in Appendix D.

for at least one input and cannot be removed without changing the induced dynamics. We further impose domain-specific structural constraints to avoid degenerate worlds.

The domains differ in how applicable rules are composed into state transitions: physics applies them simultaneously, chemistry sequentially, and biology asynchronously. The resulting dynamics may be nondeterministic, as a rule set may map a state to multiple successor states, so an experiment may induce multiple trajectories and a set of reachable final states rather than a single outcome. Different rule sets may induce identical transition dynamics. We preserve these as distinct rule-level hypotheses while accounting for their behavioral equivalence when computing predictions and validating benchmark instances. Full rule grammars and transition procedures are in Appendix D.

4.3 Experiments and Forecasts

An experiment is defined by an initial state and a sequence of interventions applied over a finite number of transition steps. At each step, the world first evolves according to the transition dynamics induced by the governing rule set, after which the intervention overwrites the value of a designated state variable. The resulting state becomes the input to the next transition. We consider experiments of one to three steps and, because the state and intervention spaces are finite, enumerate the possible experiments used during benchmark construction.

Given an experiment x and a rule set R, we compute the set of all final states reachable under the induced dynamics, denoted by $\mathrm{Reach}_R(x)$. This set is nonempty and may contain multiple states when the transition dynamics branch. A forecast query $q = (x, \varphi)$ consists of an experiment x and a proposition φ about its final states, such as whether a variable equals, exceeds, or falls below a given value. We assign a modal forecast label according to the truth of φ over all reachable final states:

$$f_q(R) = F_\varphi(R, x) = \begin{cases} \texttt{INEVITABLE}, & \text{if } \varphi(s) \text{ holds for all } s \in \mathrm{Reach}_R(x), \\ \texttt{IMPOSSIBLE}, & \text{if } \varphi(s) \text{ holds for no } s \in \mathrm{Reach}_R(x), \\ \texttt{POSSIBLE_NOT_INEVITABLE}, & \text{otherwise.} \end{cases}$$

Each modal label summarizes the reachable outcomes under a fixed rule set. For forecast-invariant queries, all evidence-consistent completions agree on the label, which is also obtained by pooling their reachable final states (Lemma A.5). Since the transition systems are finite, reachable states and forecast labels can be computed exactly.

4.4 Evidence, Version Spaces, and Instance Validation

For each observed rule set R_{obs}, we define $\mathcal{H}(R_{\mathrm{obs}})$ as the set of admissible completions obtained by adding the prescribed number of hidden rules, including the ground-truth completion $R^* = R_{\mathrm{obs}} \cup R_{\mathrm{hid}}$. Different completions may induce identical transition dynamics but remain distinct rule-

Algorithm 1 Benchmark Instance Construction

Require: Observed rules R_{obs}, ground-truth completion R^*, finite candidate query set $\mathcal{C} \subseteq \mathcal{Q}$
1: Enumerate admissible completions $\mathcal{H}(R_{\text{obs}})$
2: Generate distinguishing evidence $E = ((q_i, y_i))_{i=1}^n$ under R^*
3: $\mathcal{V}(E) \leftarrow \{R \in \mathcal{H}(R_{\text{obs}}) : f_{q_i}(R) = y_i,\ \forall i\}$
4: $\mathcal{C}_{\text{valid}}(E) \leftarrow \{q \in \mathcal{C} : |\{f_q(R) : R \in \mathcal{V}(E)\}| = 1\}$
5: **if** $|\mathcal{C}_{\text{valid}}(E)|$ is insufficient **then**
6: **return** REJECT
7: **end if**
8: Sample forecasts $Q \subseteq \mathcal{C}_{\text{valid}}(E)$
9: **return** $(R_{\text{obs}}, R^*, E, Q)$

level hypotheses. For this fixed public design D, which includes the observed rules and prescribed hidden-rule count, we write $\mathcal{H}(R_{\text{obs}}) = \mathcal{H}_D$ and $\mathcal{V}(E) = \mathcal{V}_D(E)$.

We construct evidence from distinguishing experiments whose forecasts differ between the observed and complete rule sets. Given evidence $E = ((q_i, y_i))_{i=1}^n$ generated under R^*, where $q_i = (x_i, \varphi_i)$, the version space contains all admissible completions consistent with the evidence:

$$\mathcal{V}(E) = \{R \in \mathcal{H}(R_{\text{obs}}) : f_{q_i}(R) = y_i,\ \forall i\}\,.$$

We retain a forecast query $q = (x, \varphi)$ whenever all hypotheses in the version space agree:

$$|\{f_q(R) : R \in \mathcal{V}(E)\}| = 1.$$

The singleton test provides an exact validation criterion for the paired benchmark.

Corollary 4.1 (Exact validation and shared targets)**.** *Suppose the admissible completion space and candidate query set are finite and explicitly enumerable, all required forecast labels are computed exactly, and $\mathcal{V}(E) \neq \varnothing$. Every query accepted by the singleton test has a label independent of the evidence-consistent reference completion; every query rejected by this test admits two consistent completions with different labels. For any nonempty set of accepted queries, the generating completion $R^* \in \mathcal{V}(E)$ supplied in CR and the HR task share the same target vector, and each task admits an exact procedure returning it.*

The corollary follows from Corollaries A.4 and A.6; their proofs are given in the appendix A. Algorithm 1 summarizes the construction procedure.

4.5 DATASET COMPOSITION

Each domain has four structural configurations, $|R_{\text{obs}}| \in \{2, 3\}$ and $|R_{\text{hid}}| \in \{1, 2\}$, with 100 valid instances per configuration, for 400 instances per domain and 1,200 in total. Each instance contains five evidence observations and ten forecast queries. To support analysis of evidence depth, 20% of instances contain only one-step evidence, 20% only two-step, 20% only three-step, and the remaining 40% a mixture. To avoid trivial biases in the target distribution, each evidence set contains exactly one `POSSIBLE_NOT_INEVITABLE` outcome, and forecast labels are balanced across `INEVITABLE`, `POSSIBLE_NOT_INEVITABLE`, and `IMPOSSIBLE`. At least eight of the ten forecast queries are distinguishing, and instances that cannot satisfy these constraints are discarded.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUPS

We evaluate GPT-5 (low and high reasoning effort), GPT-5-mini, GPT-5-nano, GPT-OSS-120B, and Qwen3-32B, Qwen3-30B-A3B, and Qwen3-8B in thinking mode on all 1,200 instances under both CR and HR, each with its maximum supported generation budget. Both conditions use the same prompt template and ask only for a JSON object mapping each forecast query to a modal label, without requesting reasoning or rule inference. Queries without a valid label are scored as incorrect, and we report coverage, the fraction of instances with a complete, valid prediction, to separate incomplete generation from incorrect prediction. Prompts and settings are in Appendix E.2.

Table 2: Main results on the paired CR and HR evaluation. $\Delta = \mathrm{Acc}_{\mathrm{CR}} - \mathrm{Acc}_{\mathrm{HR}}$.

	Physics			Chemistry			Biology			Overall		
Model	$\mathrm{Acc}_{\mathrm{CR}}$	$\mathrm{Acc}_{\mathrm{HR}}$	Δ	$\mathrm{Acc}_{\mathrm{CR}}$	$\mathrm{Acc}_{\mathrm{HR}}$	Δ	$\mathrm{Acc}_{\mathrm{CR}}$	$\mathrm{Acc}_{\mathrm{HR}}$	Δ	$\mathrm{Acc}_{\mathrm{CR}}$	$\mathrm{Acc}_{\mathrm{HR}}$	Δ
Qwen3-8B	0.766	0.515	+0.251	0.618	0.389	+0.229	0.618	0.397	+0.221	0.667	0.433	+0.234
Qwen3-30B-A3B	0.816	0.539	+0.277	0.660	0.399	+0.261	0.641	0.449	+0.192	0.706	0.462	+0.243
Qwen3-32B	0.803	0.623	+0.180	0.723	0.481	+0.242	0.709	0.476	+0.233	0.745	0.527	+0.218
GPT-OSS-120B	0.996	0.931	+0.064	0.986	0.882	+0.104	0.981	0.877	+0.104	0.988	0.897	+0.091
GPT-5-nano	0.831	0.477	+0.353	0.627	0.371	+0.256	0.694	0.389	+0.305	0.717	0.413	+0.305
GPT-5-mini	0.997	0.953	+0.044	0.988	0.860	+0.128	0.986	0.886	+0.100	0.990	0.900	+0.091
GPT-5 (low)	0.996	0.932	+0.064	0.989	0.835	+0.153	0.991	0.842	+0.150	0.992	0.870	+0.122
GPT-5 (high)	1.000	0.988	+0.011	0.999	0.984	+0.015	0.999	0.976	+0.023	0.999	0.983	+0.016

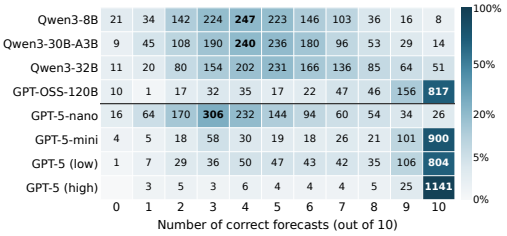

Figure 3: Distribution of HR scores.

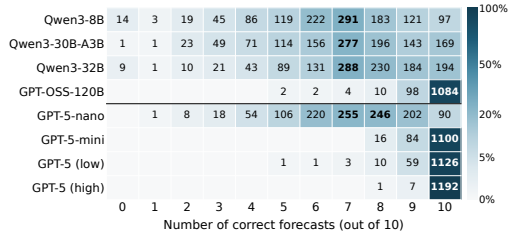

Figure 4: Distribution of CR scores.

5.2 Consistent CR–HR Gap Across Models

Table 2 reports a positive CR–HR gap for every evaluated model in each domain. This pattern remains evident for model configurations approaching the CR ceiling. For GPT-5 (high), 1,192 of the 1,200 CR instances are answered correctly on all ten forecast queries, indicating that rule-given forward prediction is nearly saturated within the difficulty range considered in this evaluation. The instance-level score distributions show the same contrast: HR scores are more dispersed across instances, whereas CR scores for near-ceiling configurations are concentrated near the maximum (Figures 3 and 4). This leaves limited room for forward-prediction error under explicitly provided rules to account for the corresponding HR deficit. Among configurations with near-ceiling CR accuracy, HR also incurs greater generation cost, with median token usage 1.62–$2.08\times$ that of CR, as reported in Table 3.

For GPT-5, the magnitude of the CR–HR gap is also sensitive to inference-time reasoning. Increasing reasoning effort from `low` to `high` raises HR accuracy from 0.870 to 0.983, whereas CR accuracy changes only from 0.992 to 0.999. Consequently, the CR–HR gap decreases from 0.122 to 0.016. Thus, while the gap is positive across all evaluated models, the GPT-5 comparison shows that it can be substantially reduced with greater reasoning effort.

5.3 Where and Why the Gap Arises

We analyze what widens the CR–HR gap, examining evidence depth and version-space fraction, and where HR forecasting fails, locating the errors of GPT-OSS-120B within the enumerable hypothesis space. Full results for all models are in Appendix C.

Evidence depth. Table 3 reports the gap by the number of steps in the observed experiments. To enable this analysis, we constructed 20% of the problems each with only one-step, only two-step, and only three-step evidence, yielding 240 problems per condition; the remaining 40%, which mix evidence lengths, are excluded. For every model except Qwen3-8B, the gap grows with the number of steps, often substantially: GPT-5 (low) rises from 0.019 to 0.169, and GPT-5-mini from 0.029 to 0.125. CR sees the same evidence as HR but, given the rules, has no need to use it. The increase therefore reflects not the difficulty of multi-step forward execution but the burden of inferring rules backward from evidence. With one-step evidence, the effect of the hidden rules appears directly in the observed outcome. As the number of steps grows, branches split further, and the effect of the hidden rules becomes entangled with the public rules and interventions, making it harder to trace back.

Table 3: (a) Coverage, including the maximum supported tokens (Max Tok.); (b) median generated tokens in thousands; (c) CR–HR gap Δ by evidence depth.

	(a) Coverage			(c) Tokens		(c) Evidence Depth		
Model	Max Tok.	CR	HR	CR	HR	1 step	2 steps	3 steps
Qwen3-8B	40K	0.988	0.986	12.2K	10.8K	0.249	0.227	0.223
Qwen3-30B-A3B	40K	0.999	0.998	11.0K	9.5K	0.221	0.245	0.272
Qwen3-32B	40K	0.993	0.993	10.1K	13.9K	0.189	0.231	0.253
GPT-OSS-120B	128K	1.000	0.992	28.4K	48.2K	0.063	0.077	0.105
GPT-5-nano	400K	0.999	0.999	52.5K	33.7K	0.267	0.316	0.317
GPT-5-mini	400K	0.999	0.999	15.2K	31.6K	0.029	0.080	0.125
GPT-5 (low)	400K	1.000	1.000	6.8K	11.3K	0.019	0.116	0.169
GPT-5 (high)	400K	1.000	1.000	16.6K	26.8K	0.000	0.009	0.023

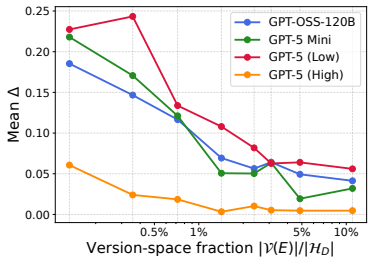

Figure 5: CR–HR gap Δ as a function of the version-space fraction $|\mathcal{V}(E)|/|\mathcal{H}_D|$.

Version-space fraction. Figure 5 shows the gap as a function of the version-space fraction $|\mathcal{V}(E)|/|\mathcal{H}_D|$, the proportion of hidden rule sets consistent with the evidence in each problem. This fraction equals the probability that a hidden rule set chosen uniformly at random from all admissible ones lies in the version space. For all four models whose CR accuracy is near ceiling, the gap is larger when the fraction is smaller: it is 0.19–0.23 in the smallest bin and falls to 0.03–0.06 in the largest. % replace with exact values This follows from how the benchmark is constructed. Since every hypothesis in the version space yields the same forecasts, a model need not find the true rules; it only needs to reach some hypothesis in the version space. When the fraction is large, a roughly correct hypothesis suffices; when it is small, the model must use the evidence to rule out most of the hypothesis space. Note that the fraction is not a controlled variable: problems with small fractions mostly have two hidden rules, so the result reflects both factors (Appendix E.1).

Failure localization. Because the hypothesis space is enumerable, we can check exactly whether a hypothesis stated by a model lies in the version space. We therefore run GPT-OSS-120B in a separate condition that additionally asks it to state the hidden rules it infers; we keep this separate from the main HR evaluation because committing to explicit rules itself changes model behavior. The model states an evidence-consistent rule set in 80.6% of problems (91.5% with one hidden rule, 69.7% with two); the remainder are either ruled out by the evidence or fall outside the domain grammar. With a consistent rule, its forecast accuracy (0.985) matches its CR accuracy (0.988). With an inconsistent rule, accuracy drops to 0.631, yet 84.5% of its forecasts still match the labels implied by its own stated rule, indicating that it executes the committed hypothesis faithfully but commits to one the evidence has already ruled out. The main HR traces reveal the complementary failure: traces that miss three or more forecasts frequently end by declaring the evidence contradictory (66.3% vs. 6.7% of perfect traces), and in each of three such traces we read in full, the model had generated the correct rule but rejected it after a single-step calculation error. Verification thus fails in both directions, suggesting that the bottleneck lies in checking candidates against the evidence rather than in executing a committed hypothesis (Appendix C.2).

6 Conclusion

We introduced RuleCast, a benchmark that evaluates abductive reasoning by pairing each forecasting problem with a matched rule-given control. The Complete-Rule and Hidden-Rule conditions share the same world, evidence, and forecast queries and differ only in whether the hidden rules are provided, so their gap isolates the additional difficulty of inferring the premises required for prediction. Built on controlled transition-system worlds, the benchmark allows multiple hypotheses to remain consistent with the evidence while requiring them to agree on every evaluated forecast. Across the evaluated models, forecast accuracy consistently drops when the rules must be inferred, even for models that execute the given rules nearly perfectly. The gap widens with evidence depth and narrows as more of the hypothesis space remains consistent with the evidence, and an error analysis over the enumerable hypothesis space locates the bottleneck in verifying candidate hypotheses against the evidence rather than in executing a committed hypothesis. The gap also narrows with greater reasoning effort, suggesting that harder configurations, such as more hidden rules or longer evidence, are a natural next step for tracking progress in abductive reasoning.

AI USE STATEMENT

We used generative AI tools to assist with writing, to improve the clarity, grammar, and readability of the manuscript, to help identify related work, and as coding assistants for implementing the benchmark generation and evaluation code. The authors reviewed and edited all AI-assisted text, and reviewed and tested all AI-assisted code, and take full responsibility for the final content of the paper.

ETHICS STATEMENT

This work develops a benchmark for evaluating abductive reasoning in language models. All instances are generated programmatically from synthetic worlds, without human participants or personal data, and the evaluated models are used under their respective licenses and terms. Because the domains are simplified abstractions, benchmark performance should not be taken as evidence that a model can reliably infer hidden mechanisms in real scientific settings, which require independent experimental validation.

REPRODUCIBILITY STATEMENT

Benchmark construction is described in Section 4 and Algorithm 1, with full domain specifications in Appendix D. Proofs of the theoretical results in Sections 3.2 and 4.4 are in Appendix A, and prompts, models, and generation settings are in Appendices E.2 and B. We will release the benchmark instances, generation code, and evaluation scripts; an anonymized version is included in the supplementary material.

REFERENCES

Wassim Abou-Jaoudé, Pauline Traynard, Pedro T Monteiro, Julio Saez-Rodriguez, Tomáš Helikar, Denis Thieffry, and Claudine Chaouiya. Logical modeling and dynamical analysis of cellular networks. *Frontiers in genetics*, 7:94, 2016.

Luca Aceto, Anna Ingólfsdóttir, Kim Guldstrand Larsen, and Jiri Srba. *Reactive systems: modelling, specification and verification*. cambridge university press, 2007.

Rajeev Alur, Costas Courcoubetis, Thomas A. Henzinger, and Pei Hsin Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In Robert L. Grossman, Anil Nerode, Anders P. Ravn, and Hans Rischel (eds.), *Hybrid Systems*, pp. 209–229, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg. ISBN 978-3-540-48060-0.

Christel Baier and Joost-Pieter Katoen. *Principles of model checking*, volume 735. 2008.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. Abductive commonsense reasoning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=Byg1v1HKDB.

Scott M Bugenhagen and Daniel A Beard. Specification, construction, and exact reduction of state transition system models of biochemical processes. *The Journal of Chemical Physics*, 137(15), 2012.

Tom Bylander, Dean Allemang, Michael C Tanner, and John R Josephson. The computational complexity of abduction. *Artificial intelligence*, 49(1-3):25–60, 1991.

Pengfei Cao, Mingxuan Yang, Yubo Chen, Chenlong Zhang, Mingxuan Liu, Kang Liu, and Jun Zhao. Semeval-2026 task 12: Abductive event reasoning: Towards real-world event causal inference for large language models. In *Proceedings of the 20th International Workshop on Semantic Evaluation (2026)*, pp. 3659–3672, 2026.

Kewei Cheng, Jingfeng Yang, Haoming Jiang, Zhengyang Wang, Binxuan Huang, Ruirui Li, Shiyang Li, Zheng Li, Yifan Gao, Xian Li, et al. Inductive or deductive? rethinking the fundamental reasoning abilities of llms. *arXiv preprint arXiv:2408.00114*, 2024.

Patrick Cooper and Alvaro Velasquez. Defab: A verifiable benchmark for defeasible abduction in foundation models. *arXiv preprint arXiv:2606.18557*, 2026.

Thomas Eiter and Georg Gottlob. The complexity of logic-based abduction. *Journal of the ACM (JACM)*, 42(1):3–42, 1995.

Yuval Elani, Robert V Law, and Oscar Ces. Vesicle-based artificial cells as chemical microreactors with spatially segregated reaction pathways. *Nature communications*, 5(1):5305, 2014.

Antoine Girard, A. Agung Julius, and George J. Pappas. Approximate Simulation Relations for Hybrid Systems. *Discrete Event Dynamic Systems*, 18(2):163–179, June 2008. ISSN 1573-7594. doi: 10.1007/s10626-007-0029-9. URL https://doi.org/10.1007/s10626-007-0029-9.

Kaiyu He, Peilin Wu, Mian Zhang, Kun Wan, Wentian Zhao, Xinya Du, and Zhiyu Chen. Gear: A general evaluation framework for abductive reasoning. *arXiv preprint arXiv:2509.24096*, 2025a.

Kaiyu He, Mian Zhang, Shuo Yan, Peilin Wu, and Zhiyu Zoey Chen. Idea: Enhancing the rule learning ability of large language model agent through induction, deduction, and abduction. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 13563–13597, 2025b.

Nikolaus Holzer, William Fishell, Baishakhi Ray, and Mark Santolucito. Tempobench: Evaluating temporal causal reasoning in large language models. *arXiv preprint arXiv:2510.27544*, 2025.

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.

Antonis C Kakas, Robert A. Kowalski, and Francesca Toni. Abductive logic programming. *Journal of logic and computation*, 2(6):719–770, 1992.

Guy Karlebach and Ron Shamir. Modelling and analysis of gene regulatory networks. *Nature reviews Molecular cell biology*, 9(10):770–780, 2008.

Ross D King, Kenneth E Whelan, Ffion M Jones, Philip GK Reiser, Christopher H Bryant, Stephen H Muggleton, Douglas B Kell, and Stephen G Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427(6971):247–252, 2004.

Marius Kloetzer and Calin Belta. Dealing with nondeterminism in symbolic control. In Magnus Egerstedt and Bud Mishra (eds.), *Hybrid Systems: Computation and Control*, pp. 287–300, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-78929-1.

Jiachun Li, Pengfei Cao, Zhuoran Jin, Yubo Chen, Kang Liu, and Jun Zhao. Mirage: Evaluating and explaining inductive reasoning process in language models. In *International Conference on Learning Representations*, volume 2025, pp. 73944–73969, 2025.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

Nancy Lynch, Roberto Segala, and Frits Vaandrager. Hybrid i/o automata. *Information and Computation*, 185(1):105–157, 2003. ISSN 0890-5401. doi: https://doi.org/10.1016/S0890-5401(03)00067-1. URL https://www.sciencedirect.com/science/article/pii/S0890540103000671.

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*, volume 2025, pp. 4556–4579, 2025.

Melvin H Marx. Philosophy of natural science. *PsycCRITIQUES*, 12(2):95, 1967.

Tom M Mitchell. Version spaces: A candidate elimination approach to rule learning. In *Proceedings of the 5th international joint conference on Artificial intelligence-Volume 1*, pp. 305–310, 1977.

Giordano Pola, Antoine Girard, and Paulo Tabuada. Approximately bisimilar symbolic models for nonlinear control systems. *Automatica*, 44(10):2508–2516, 2008. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2008.02.021. URL https://www.sciencedirect.com/science/article/pii/S0005109808001933.

Linlu Qiu, Liwei Jiang, Ximing Lu, Melanie Sclar, Valentina Pyatkin, Chandra Bhagavatula, Bailin Wang, Yoon Kim, Yejin Choi, Nouha Dziri, et al. Phenomenal yet puzzling: Testing inductive reasoning capabilities of language models with hypothesis refinement. In *International Conference on Learning Representations*, volume 2024, pp. 1452–1481, 2024.

Joseph M Replogle, Reuben A Saunders, Angela N Pogson, Jeffrey A Hussmann, Alexander Lenail, Alina Guna, Lauren Mascibroda, Eric J Wagner, Karen Adelman, Gila Lithwick-Yanai, et al. Mapping information-rich genotype-phenotype landscapes with genome-scale perturb-seq. *Cell*, 185(14):2559–2575, 2022.

Yu Sheng, Wanting Wen, Linjing Li, and Daniel Zeng. Evaluating generalization capability of language models across abductive, deductive and inductive logical reasoning. In *Proceedings of the 31st international conference on computational linguistics*, pp. 4945–4957, 2025.

Julius Steiglechner, Lucas Mahler, and Gabriele Lohmann. Llms can see the smoke but not the fire: Evaluating abductive reasoning with elenchos. *arXiv preprint arXiv:2607.12733*, 2026.

Yunxin Sun and Abulhair Saparov. Do language models follow occam's razor? an evaluation of parsimony in inductive and abductive reasoning. *arXiv preprint arXiv:2509.03345*, 2025.

Paulo Tabuada. Approximate simulation relations and finite abstractions of quantized control systems. In Alberto Bemporad, Antonio Bicchi, and Giorgio Buttazzo (eds.), *Hybrid Systems: Computation and Control*, pp. 529–542, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-71493-4.

Maurice H Ter Beek, Alessandro Fantechi, Stefania Gnesi, and Franco Mazzanti. An action/state-based model-checking approach for the analysis of communication protocols for service-oriented applications. In *International Workshop on Formal Methods for Industrial Critical Systems*, pp. 133–148. Springer, 2007.

René Thomas. Regulatory networks seen as asynchronous automata: a logical description. *Journal of theoretical biology*, 153(1):1–23, 1991.

Matt L Wiemann, Lindsay M Smith, Peter Melchior, Siddharth Mishra-Sharma, Andrew Gordon Wilson, Pavel Izmailov, and Carolina Cuesta-Lázaro. Discoverphysics: Benchmarking llms for out-of-the-box scientific thinking. *arXiv preprint arXiv:2605.26087*, 2026.

David P Woodruff, Vincent Cohen-Addad, Lalit Jain, Jieming Mao, Song Zuo, MohammadHossein Bateni, Simina Branzei, Michael P Brenner, Lin Chen, Ying Feng, et al. Accelerating scientific research with gemini: Case studies and common techniques. *arXiv preprint arXiv:2602.03837*, 2026.

Yuheng Wu, Berk Gokmen, Zhouhua Xie, Peijing Li, Caroline Trippel, Priyanka Raina, and Thierry Tambe. Llm-fsm: Scaling large language models for finite-state reasoning in rtl code generation. *arXiv preprint arXiv:2602.07032*, 2026.

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*, volume 2025, pp. 33251–33277, 2025.

Nathan Young, Qiming Bao, Joshua Bensemann, and Michael J Witbrock. Abductionrules: Training transformers to explain unexpected inputs. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 218–227, 2022.

Tianyun Zhong, Wangyi Jiang, Wei Wang, Xuanang Chen, Yaojie Lu, Shiwei Ye, Yuzhen Shi, Boyu Yang, Jinghang Wang, Han Li, et al. Before the action: Benchmarking llms on prospective hypothesis discovery. *arXiv preprint arXiv:2607.15766*, 2026.

A Formal Foundations

This appendix formalizes the properties needed for the benchmark construction and paired evaluation. We first characterize the hypotheses that remain consistent with finite evidence and how this set changes as observations accumulate. We then characterize when the evidence determines a reference-independent forecast target, and finally specialize these results to the modal semantics of our transition-system benchmark.

A.1 Evidence and Version Spaces

Let D denote the public design of a benchmark instance, and let $\mathcal{H}_D \neq \varnothing$ be the corresponding set of admissible hypotheses. Let $\mathcal{Q}$ denote the family of admissible experimental queries. Throughout this appendix, we abbreviate the three modal forecast labels as

$$\mathcal{Y} = \{\mathrm{IMP}, \mathrm{PNI}, \mathrm{INEV}\},$$

corresponding to *impossible*, *possible not inevitable*, and *inevitable*, respectively. For each $q \in \mathcal{Q}$, let

$$f_q : \mathcal{H}_D \to \mathcal{Y}$$

denote the exact forecast label under a hypothesis.

For $J \subseteq \mathcal{Q}$, we say that two hypotheses are *observation-equivalent on* J, written

$$H \sim_J H',$$

if

$$f_q(H) = f_q(H') \qquad \text{for every } q \in J.$$

We denote the corresponding equivalence class of H by $[H]_J$.

Let

$$E = ((q_i, y_i))_{i=1}^n$$

be a finite sequence of observed query–label pairs, where $q_i \in \mathcal{Q}$ and $y_i \in \mathcal{Y}$, and let

$$I_E = \{q_i : 1 \leq i \leq n\}$$

be the set of distinct observed queries. The *version space* induced by E is

$$\mathcal{V}_D(E) = \{H \in \mathcal{H}_D : f_{q_i}(H) = y_i \text{ for every } i\}.$$

We assume throughout that $\mathcal{V}_D(E) \neq \varnothing$.

Theorem A.1 (Version Space as an Evidence-Equivalence Class)**.** *For any $H^* \in \mathcal{V}_D(E)$, we have*

$$\mathcal{V}_D(E) = [H^*]_{I_E}.$$

Thus, the evidence identifies exactly the hypotheses that agree with H^* on all observed queries, but need not identify H^* uniquely.

proof. Let $H^* \in \mathcal{V}_D(E)$. By definition,

$$y_i = f_{q_i}(H^*) \qquad (1 \leq i \leq n).$$

Hence, for any $H \in \mathcal{H}_D$,

$$\begin{aligned} H \in \mathcal{V}_D(E) &\iff f_{q_i}(H) = y_i \quad \text{for every } i \\ &\iff f_{q_i}(H) = f_{q_i}(H^*) \quad \text{for every } i \\ &\iff H \sim_{I_E} H^*. \end{aligned}$$

□

The version space becomes more restrictive as additional observations are introduced.

Proposition A.2 (Evidence Accumulation and Identification)**.** *Fix a finite hypothesis space $\mathcal{H}_D$, a reference hypothesis $H^* \in \mathcal{H}_D$, and a family $\mathcal{A} \subseteq \mathcal{Q}$ of queries available for observation. For $q_i \in \mathcal{A}$, define*

$$E_n = \big((q_i, f_{q_i}(H^*))\big)_{i=1}^n, \qquad V_n = \mathcal{V}_D(E_n).$$

Then

$$[H^*]_{\mathcal{A}} \subseteq V_{n+1} = V_n \cap f_{q_{n+1}}^{-1}\big(\{f_{q_{n+1}}(H^*)\}\big) \subseteq V_n.$$

The last inclusion is strict exactly when some $H \in V_n$ disagrees with H^* on q_{n+1}.

Moreover, there exists a finite set $K \subseteq \mathcal{A}$ such that, for

$$E_K = \{(q, f_q(H^*)) : q \in K\},$$

we have

$$\mathcal{V}_D(E_K) = [H^*]_{\mathcal{A}}.$$

Consequently, a finite collection of available observations uniquely identifies H^* if and only if

$$[H^*]_{\mathcal{A}} = \{H^*\}.$$

proof. Adding observation q_{n+1} retains exactly those hypotheses in V_n that predict the observed outcome $f_{q_{n+1}}(H^*)$, which gives

$$V_{n+1} = V_n \cap f_{q_{n+1}}^{-1}\big(\{f_{q_{n+1}}(H^*)\}\big) \subseteq V_n.$$

The inclusion is strict precisely when at least one hypothesis in V_n predicts a different outcome on q_{n+1}. Every hypothesis in $[H^*]_{\mathcal{A}}$ agrees with H^* on every available query and therefore remains in every V_n.

For each $H \notin [H^*]_{\mathcal{A}}$, choose a query $q_H \in \mathcal{A}$ such that

$$f_{q_H}(H) \neq f_{q_H}(H^*),$$

which exists by definition of observation equivalence. Since $\mathcal{H}_D$ is finite, the collection

$$K = \{q_H : H \notin [H^*]_{\mathcal{A}}\}$$

is finite. Every hypothesis outside $[H^*]_{\mathcal{A}}$ is excluded by at least one observation in E_K, while every hypothesis in $[H^*]_{\mathcal{A}}$ remains consistent. Hence

$$\mathcal{V}_D(E_K) = [H^*]_{\mathcal{A}}.$$

The final statement follows immediately. □

A.2 Forecast-Invariant Evaluation

Global identification of the underlying hypothesis is stronger than the benchmark requires. For evaluation, it is sufficient that all hypotheses consistent with the observed evidence agree on the forecasts to be scored.

For $q \in \mathcal{Q}$, define the set of forecast labels compatible with the evidence by

$$\Gamma_E(q) = \{f_q(H) : H \in \mathcal{V}_D(E)\}.$$

A finite, nonempty set $Q \subseteq \mathcal{Q}$ is *forecast-invariant under E* if

$$|\Gamma_E(q)| = 1 \qquad \text{for every } q \in Q.$$

For a submitted forecast vector

$$\widehat{y} = (\widehat{y}_q)_{q \in Q} \in \mathcal{Y}^Q$$

and a reference hypothesis $H \in \mathcal{V}_D(E)$, define

$$\mathrm{Acc}_H(\widehat{y}) = \frac{1}{|Q|} \sum_{q \in Q} \mathbf{1}\{\widehat{y}_q = f_q(H)\}.$$

Theorem A.3 (Forecast Invariance and Reference-Independent Scoring)**.** *Let $\mathcal{V}_D(E) \neq \varnothing$, fix $H^* \in \mathcal{V}_D(E)$, and let $Q \subseteq \mathcal{Q}$ be finite and nonempty. The following conditions are equivalent:*

1. $|\Gamma_E(q)| = 1$ *for every* $q \in Q$.

2. $$\mathcal{V}_D(E) \subseteq [H^*]_Q.$$

3. *For every submitted forecast vector* $\widehat{y}$*, the value*

$$\mathrm{Acc}_H(\widehat{y})$$

is independent of the choice of $H \in \mathcal{V}_D(E)$.

When these conditions hold, there is a unique common target vector

$$y_{E,Q} = (f_q(H^*))_{q\in Q} = (f_q(H))_{q\in Q} \qquad \textit{for every } H \in \mathcal{V}_D(E).$$

In particular, neither unique identification of H^* *nor agreement of the consistent hypotheses outside* Q *is required.*

proof. Conditions 1 and 2 are equivalent by definition: every $\Gamma_E(q)$ is a singleton exactly when every hypothesis in $\mathcal{V}_D(E)$ agrees with H^* on every query in Q.

If condition 2 holds, every consistent reference hypothesis has the same target vector on Q, so the accuracy of any fixed submission is independent of the chosen reference. Thus condition 2 implies condition 3.

Conversely, suppose condition 1 fails. Then there exist $H_1, H_2 \in \mathcal{V}_D(E)$ whose predictions differ on at least one query in Q. Consider the submitted vector

$$\widehat{y} = (f_q(H_1))_{q\in Q}.$$

Then

$$\mathrm{Acc}_{H_1}(\widehat{y}) = 1,$$

whereas

$$\mathrm{Acc}_{H_2}(\widehat{y}) < 1.$$

Therefore condition 3 fails. The common-target statement follows immediately from condition 1. □

This characterization gives an exact validation rule whenever the admissible hypothesis space can be enumerated.

Corollary A.4 (Exact Forecast Validation)**.** *Suppose* $\mathcal{H}_D$ *and a candidate query set* $\mathcal{C}$ *are finite and explicitly enumerable, all required forecast labels* $f_q(H)$ *can be computed exactly, and* $\mathcal{V}_D(E) \neq \varnothing$*. Define*

$$\mathcal{C}_{\mathrm{valid}}(E) = \{q \in \mathcal{C} : |\Gamma_E(q)| = 1\}.$$

Then the singleton test accepts exactly those candidate queries whose target is independent of the evidence-consistent reference hypothesis. Every rejected query admits two hypotheses in $\mathcal{V}_D(E)$ *with different forecast labels. Consequently, any nonempty subset of* $\mathcal{C}_{\mathrm{valid}}(E)$ *admits reference-independent forecast-accuracy scoring.*

proof. For any candidate query q, the singleton condition

$$|\Gamma_E(q)| = 1$$

holds exactly when all hypotheses in $\mathcal{V}_D(E)$ give the same answer. If it fails, $\Gamma_E(q)$ contains at least two realized outcomes, hence two evidence-consistent hypotheses disagree on the target. The final statement follows from Theorem A.3. □

A.3 Modal Semantics and Paired Forecasting

We now specialize the preceding results to the transition-system benchmark. A forecast query

$$q = (x, \varphi)$$

consists of an experiment x and a proposition φ on final states. For an admissible hypothesis H, let $\mathrm{Reach}_H(x)$ denote the nonempty set of final states reachable under H.

For any nonempty set of states S, define

$$\mathrm{Mod}_\varphi(S) = \begin{cases} \mathrm{INEV}, & \varphi(s) = 1 \text{ for every } s \in S, \\ \mathrm{IMP}, & \varphi(s) = 0 \text{ for every } s \in S, \\ \mathrm{PNI}, & \text{otherwise.} \end{cases}$$

The exact forecast label under H is

$$f_q(H) = \mathrm{Mod}_\varphi\big(\mathrm{Reach}_H(x)\big).$$

Lemma A.5 (Consensus Preserves Modal Semantics)**.** *Suppose $\mathcal{V}_D(E) \neq \varnothing$ and*

$$\Gamma_E(q) = \{y\}$$

for $q = (x, \varphi)$. Define the pooled reachable set

$$\mathrm{Reach}_E(x) = \bigcup_{H \in \mathcal{V}_D(E)} \mathrm{Reach}_H(x).$$

Then

$$\mathrm{Mod}_\varphi\big(\mathrm{Reach}_E(x)\big) = y = f_q(H) \qquad \textit{for every } H \in \mathcal{V}_D(E).$$

proof. If $y = \mathrm{INEV}$, every state reachable under every consistent hypothesis satisfies φ, so every state in their union also satisfies φ. The argument for $y = \mathrm{IMP}$ is identical.

If $y = \mathrm{PNI}$, choose any $H \in \mathcal{V}_D(E)$. Since $f_q(H) = \mathrm{PNI}$, $\mathrm{Reach}_H(x)$ contains both a state satisfying φ and a state violating it. Both belong to $\mathrm{Reach}_E(x)$, so the pooled reachable set also has modal label PNI. □

Trajectory uncertainty versus hypothesis disagreement. The modal label PNI within a fixed hypothesis represents uncertainty among trajectories generated by that hypothesis. This differs from disagreement between hypotheses. For example, if one evidence-consistent hypothesis predicts INEV and another predicts IMP, pooling their reachable states may produce a PNI label, but the completion-specific forecasts disagree and the query fails the singleton test. Lemma A.5 shows that the pooled and completion-specific interpretations coincide on the forecast-invariant queries retained by the benchmark.

Corollary A.6 (Shared Targets and Exact-Solver Attainability)**.** *Suppose $\mathcal{H}_D$ is finite and explicitly enumerable, all semantic outcomes can be computed exactly, and Q is finite, nonempty, and forecast-invariant under evidence E. Let the complete rule set supplied in CR be any*

$$H^* \in \mathcal{V}_D(E).$$

Then CR and HR have the same target vector $y_{E,Q}$. Moreover, both tasks admit an exact procedure achieving forecast accuracy one:

$$\mathrm{Acc}_{\mathrm{CR}}^{\mathrm{exact}} = \mathrm{Acc}_{\mathrm{HR}}^{\mathrm{exact}} = 1.$$

proof. In CR, the exact procedure directly computes $f_q(H^*)$ for every $q \in Q$. In HR, an exact procedure enumerates the version space $\mathcal{V}_D(E)$ and computes $\Gamma_E(q)$ for every $q \in Q$. Forecast invariance makes every such set a singleton, and Theorem A.3 implies that both procedures return the same vector $y_{E,Q}$. □

Scope of the paired comparison. The preceding results ensure that unresolved ambiguity about the reference completion cannot explain errors on retained forecast queries. They do not imply that the CR–HR accuracy gap is an additive decomposition of pure hypothesis-inference error.

To make this distinction explicit, consider a hypothetical two-stage predictor that first outputs an admissible hypothesis $\widehat{H} \in \mathcal{H}_D$ and then produces a forecast $\widehat{y}_q$. For a retained query with common target y_q^*,

$$\Pr[\widehat{y}_q \neq y_q^*] \leq \Pr[\widehat{H} \notin \mathcal{V}_D(E)] + \Pr[\widehat{y}_q \neq f_q(\widehat{H})].$$

Indeed, if $\widehat{H} \in \mathcal{V}_D(E)$ and $\widehat{y}_q = f_q(\widehat{H})$, forecast invariance implies

$$\widehat{y}_q = y_q^*.$$

The second term concerns execution of the inferred hypothesis $\widehat{H}$, whereas CR measures execution when the supplied complete rule set H^* is given directly. The benchmark also does not require the model to expose an intermediate hypothesis. Accordingly, the CR–HR gap characterizes the additional difficulty of the hidden-rule condition but does not separately identify its internal inference and execution components.

B Experimental Details

Evaluation protocol. Each model is evaluated on all 1,200 benchmark instances under both the Complete-Rule (CR) and Hidden-Rule (HR) conditions, yielding 2,400 prompts per model. The two conditions share the same system prompt, user-prompt template, world description, evidence, and forecast queries; the only difference is whether the hidden rules are listed in the user prompt. In the HR condition, the prompt states that additional hidden rules exist and gives their number, but not their content. The system prompt instructs the model to reason only from the fictional system described in the user message and prohibits web search, retrieval, code execution, calculators, and external solvers. The full prompts are shown in Appendix E.2.

Models and generation settings. Table 4 lists the evaluated models and their generation settings. Open-weight models are served with vLLM 0.28.0 on NVIDIA RTX PRO 6000 Blackwell Max-Q GPUs, and GPT-5-family models are queried through the OpenAI API using the 2025-08-07 snapshots of GPT-5, GPT-5-mini, and GPT-5-nano. Qwen3 models are run with thinking enabled, and models with configurable reasoning effort use the effort level shown in the table. Open-weight models are decoded greedily (temperature 0, top-p 1.0, seed 0). The GPT-5-family API does not accept sampling parameters, so these requests specify only the messages, token budget, and reasoning effort, and decoding follows the API's default behavior. We generate a single response per instance and condition, with no retries or resampling. Each model is given the largest generation budget supported by its serving configuration, so that failures reflect the reasoning process rather than an artificially small budget.

Table 4: Evaluated models and generation settings.

Model	Backend	Reasoning setting	Temp / top-p	Max tokens
Qwen3-8B	vLLM	enable thinking = true	0.0 / 1.0	40K
Qwen3-30B-A3B	vLLM	enable thinking = true	0.0 / 1.0	40K
Qwen3-32B	vLLM	enable thinking = true	0.0 / 1.0	40K
GPT-OSS-120B	vLLM	effort = high	0.0 / 1.0	128K
GPT-5-nano	OpenAI API	effort = high	API Default	128K
GPT-5-mini	OpenAI API	effort = high	API Default	128K
GPT-5 (low)	OpenAI API	effort = low	API Default	128K
GPT-5 (high)	OpenAI API	effort = high	API Default	128K

Output parsing and validity. Models are instructed to return only a JSON object that maps each forecast identifier (`f00`–`f09`) to one of the three labels `impossible`, `possible_not_inevitable`, and `inevitable`. We extract the final JSON object from the response, excluding any separate reasoning channel. A forecast query receives a *valid label* if its

identifier appears exactly once with one of the three labels as its value. A response is *incomplete* if it exhausts the generation budget or cannot be parsed, in which case none of its queries receives a valid label. Under the rule-stating variant used for GPT-OSS-120B in Section C.2, the object carries an additional `additional_rules` field: a list of sentences written in the same form as the observed rules. A stated rule set is *valid* if it lists exactly the number of rules the instance declares, every sentence maps to a rule in the domain grammar, and the rules together form a legal rule set. Instances with an invalid rule set are excluded from the faithfulness analysis but retained in forecast accuracy.

Scoring and coverage. Instance accuracy is the fraction of the ten forecast queries an instance labels correctly. Incomplete responses are not resampled: a query that yields no valid label is counted as incorrect, so an instance that yields none receives zero accuracy. Reported accuracies average over all 1,200 instances, including incomplete ones. To separate incomplete generation from incorrect prediction, we report coverage for each model and condition (Table 3): the fraction of instances for which all ten queries carry a valid label. The lowest coverage in any model–condition cell is 0.986 (Qwen3-8B under HR), so incomplete generation accounts for at most 1.4% of instances anywhere in the evaluation.

Token usage. We report the mean number of generated tokens per response, including reasoning tokens. For API models, this is the output-token count returned by the API, which includes both reasoning and visible output tokens. For open-weight models served with vLLM, it is the number of sampled tokens, covering the full generation including the Qwen3 `<think>` block and the GPT-OSS analysis channel. Responses that reach the generation budget are counted at that budget.

C Additional Analysis

C.1 Gap by Version-Space Fraction for All Models

Figure 6 extends Figure 5 to all eight model configurations, using the same bins of the version-space fraction $|\mathcal{V}(E)|/|\mathcal{H}_D|$. Two regimes emerge. For models whose CR accuracy is near ceiling (GPT-OSS-120B, GPT-5-mini, and GPT-5 (low)), the gap decreases almost monotonically. GPT-5 (high) follows the same trend at a lower level (0.06 to below 0.01). For the Qwen3 models and GPT-5-nano, the gap stays large across all bins and declines only mildly. These models are far from the CR ceiling (0.67–0.75), and their HR accuracy (0.41–0.53) is low. The gap therefore mainly reflects the overall difference between the two conditions and varies little with how much the evidence must narrow the hypothesis space. The fraction is not a controlled variable: small fractions arise mostly from instances with two hidden rules.

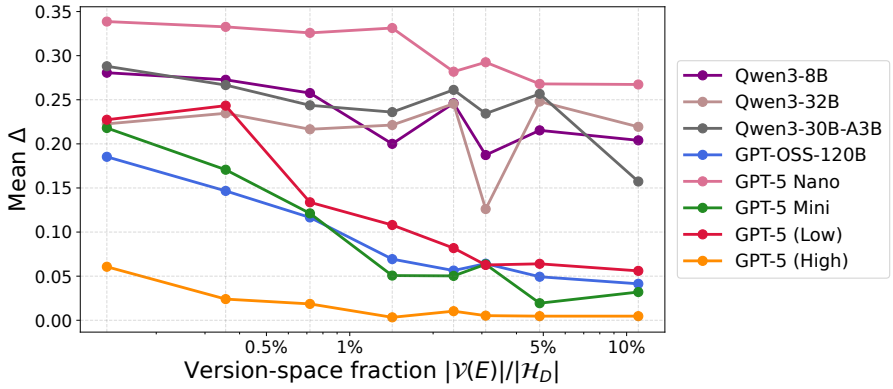

Figure 6: CR–HR gap Δ as a function of the version-space fraction $|\mathcal{V}(E)|/|\mathcal{H}_D|$ for all evaluated models. Figure 5 shows the subset of models with near-ceiling CR accuracy.

Table 5: Trace statistics of GPT-OSS-120B in the HR condition, grouped by the number of incorrect forecasts (out of 10). Contradiction declarations are statements that no rule set fits the evidence; all-impossible responses label every forecast as impossible.

	Incorrect forecasts		
	0	1–2	≥3
# traces	817	202	181
Median generated tokens	43.5K	54.9K	78.7K
Contradiction declarations	6.7%	17.3%	66.3%
All-impossible responses	0	0	53

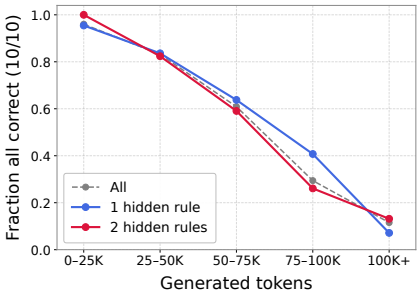

Figure 7: Fraction of GPT-OSS-120B HR instances with all ten forecasts correct, by the number of generated tokens.

Table 6: Rule recovery and forecast faithfulness for GPT-OSS-120B. A stated rule is *consistent* when it lies in the version space, *inconsistent* when it is a legal rule the evidence rules out, and *invalid* when it could not be mapped to the domain grammar. FAITH is the share of the ten forecasts that match the labels implied by the model's own stated rule, so it is undefined for invalid rules.

	1 hidden rule			2 hidden rules			Overall		
Stated rule	n	ACC	FAITH	n	ACC	FAITH	n	ACC	FAITH
Consistent	549	0.986	0.986	418	0.983	0.983	967	0.985	0.985
Inconsistent	38	0.647	0.847	119	0.625	0.844	157	0.631	0.845
Invalid	13	0.315	—	63	0.619	—	76	0.567	—
All	600	0.950	0.977	600	0.874	0.952	1,200	0.912	0.965

C.2 WHERE HIDDEN-RULE FORECASTING FAILS

Trace analysis. Because open-weight models expose their full reasoning traces, we analyzed all 1,200 HR traces of GPT-OSS-120B to locate where errors arise. The traces follow a common skeleton: the model first works out the semantics of the world model, then proposes candidate rules and checks them against the observed experiments, rejecting a candidate when it fails and proposing another. Only after committing to a set it judges consistent with every observation does it simulate the forecast experiments. Accuracy declines as reasoning grows longer. Traces that miss three or more forecasts use about 1.8× as many tokens as perfect traces (Table 5), and the share of perfect answers falls monotonically from 0.960 below 25K tokens to 0.115 above 100K. This decline persists within each hidden-rule stratum (Figure 7), so it is not explained by the number of hidden rules alone. Long failing traces often end by declaring the evidence contradictory. Statements that no rule set fits the observations appear in 6.7% of perfect traces but in 66.3% of traces that miss three or more forecasts. In 53 of these, the model concludes that no admissible evolution exists and labels all ten forecasts impossible. We read three of these collapsed traces in full. In all three, the model had in fact generated the correct rule, at 68%, 43%, and 78% of the way through the trace, but rejected it after a single-step calculation error while checking it against the observations. In one case the model substituted the wrong base score. In another, it correctly restated the intervention order given in the prompt and then applied it in reverse. In the third, it scored the intermediate state of one experiment against the statement of another. In all three cases, the model spent its remaining budget re-reading the world rules and never re-checked its own calculation.

Rule recovery. Our primary goal is paired evaluation rather than rule recovery. However, because the hypothesis space in our benchmark is fully enumerable, we can determine exactly whether a stated rule set lies in the version space and which forecast labels it entails. We exploit this by changing only the output format for GPT-OSS-120B, asking it to state the hidden rules it infers alongside its forecasts. The main HR prompt does not request the rules so that its output format matches that of CR. The format change also alters behavior: because the model must commit to a hypothesis, contradiction declarations drop to 6.5% and all-impossible responses to 5. We therefore treat this analysis as a separate condition from the main HR evaluation. As shown in Table 6, the model states

Table 7: Summary of the three synthetic domains.

	Physics	**Chemistry**	**Biology**
State	quantities, modes	species in compartments	levels, scores
Rules	change quantity or mode	reaction, transport	shift next score
Composition	simultaneous	sequential	asynchronous
Nondeterminism	net numerical effect	reaction amount	updated component

an evidence-consistent rule set for 80.6% of problems, but this rate falls from 91.5% with one hidden rule to 69.7% with two. The remaining cases are either *inconsistent*, stating a rule the evidence rules out, or *invalid*, stating a rule outside the domain grammar. Both increase sharply with two hidden rules (6.3% to 19.8% and 2.2% to 10.5%, respectively). As an example of an invalid rule, in biology a rule's effect applies only to the next component in the cycle, yet the model sometimes ignores this order and states a rule targeting an arbitrary component. The agreement between the model's forecasts and the labels entailed by its own stated rule (faithfulness) exposes errors in executing a committed hypothesis. With a consistent rule, faithfulness and accuracy are both 0.985, nearly identical to the model's CR accuracy of 0.988. With an inconsistent rule, accuracy drops to 0.631, yet faithfulness remains at 0.845. Executing a committed hypothesis is thus nearly as reliable as executing the supplied rules; errors arise from committing to a hypothesis the evidence has already ruled out.

Summary. Together, the two analyses locate the bottleneck of HR forecasting neither in generating hypotheses nor in executing a committed one, but in verifying candidates against the evidence. This verification fails in both directions: the model rejects a correct hypothesis after a calculation error (the collapsed traces), or accepts a hypothesis the evidence refutes (the inconsistent rules). Since the same simulation is accurate at CR level when producing the final forecasts, the difficulty lies not in simulation per se but in using it as a test inside the search loop.

D Domain Specification

In this section, we provide the detailed design and specification of each synthetic scientific domain. Table 7 summarizes the three domains. For each domain, we first describe the scientific motivation behind its abstraction, simplifications, and dynamical choices. We then provide the exact domain specification presented to models in the benchmark prompts.

D.1 Physics

Domain Design Physical systems are commonly represented as hybrid systems combining continuous state variables with discrete operating modes, guards, and resets (Alur et al., 1993; Lynch et al., 2003). Such continuous and hybrid systems can in turn be abstracted into finite symbolic transition systems, where discrete symbols represent aggregates or quantizations of continuous states (Tabuada, 2007; Pola et al., 2008). Finite abstractions may naturally be nondeterministic, retaining multiple admissible successors when the underlying behavior is unresolved at the abstraction resolution (Girard et al., 2008; Kloetzer & Belta, 2008). Figure 8 provides a concrete example of these dynamics, illustrating how the domain rules and an external intervention jointly determine the successor state.

Exact Prompt Specification The following box reproduces the exact domain specification used in the benchmark. This text is inserted into the WORLD MODEL section of the full user prompt provided in Appendix E.2.

World Model for Physics

Entities

- "probe coordinate" (Q0): a quantity, an integer from -2 to 2.
- "constraint coordinate" (Q1): a quantity, an integer from -2 to 2.

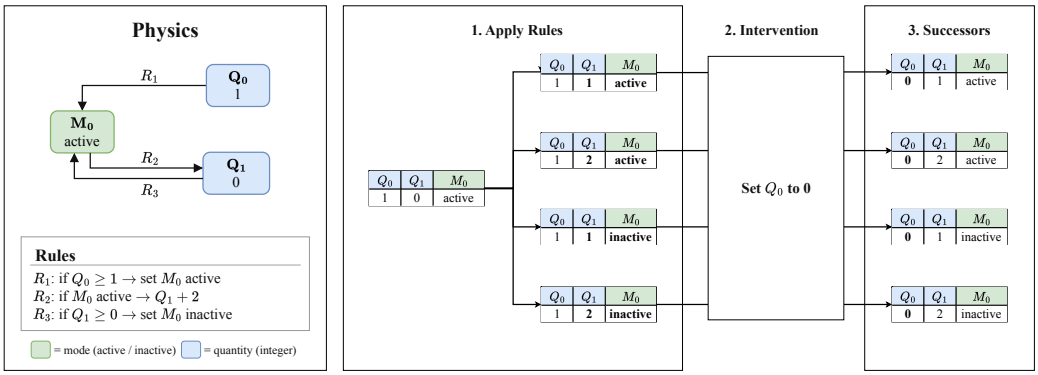

Figure 8: **Illustrative example of the Physics domain.** The example shows how simultaneous rule application, nondeterministic branching, and an external intervention determine the successor states. Starting from $(Q_0, Q_1, M_0) = (1, 0, \text{active})$, conflicting updates from R_1 and R_3 yield two admissible values of M_0, while the $+2$ response from R_2 allows Q_1 to either move one level toward its target or reach it directly. These independent choices produce four admissible endogenous successor states. The intervention is applied last, setting Q_0 to 0 in every branch.

- "boundary regime" (M0): a mode, either active or inactive.

Dynamics
Every rule has the following form:

- Exactly one condition and exactly one effect.
- Condition: a quantity is compared with a constant from -2 to 2 (equal, at least, or at most), or a mode is checked to be active or inactive.
- Effect: one of -2, -1, +1, or +2 is added to a quantity, or a mode is set to active or inactive.
- No two rules are identical.

State transitions occur in this exact order within each step:

1. Every rule is evaluated against the state at the start of the step, and every rule whose condition holds is applied simultaneously.
2. Quantities: if several rules add to the same quantity, their amounts are summed. If the sum leaves the range -2 to 2, it is adjusted to the nearest allowed value; the result is the quantity's target. If the target differs from the current value by 1, the quantity takes the target. If it differs by 2 or more, the quantity either reaches the target in this step or moves only 1 toward it, and each outcome yields one admissible successor state.
3. Modes: if one rule activates a mode and another deactivates it in the same step, each value yields one admissible successor state. Choices for different quantities and modes are independent and combine.
4. If there is an external intervention in this step, it is applied last (see Interventions).

Interventions
An intervention sets one quantity or one mode to a specified value. It is applied at the end of the step, after all rules have been applied and adjusted, and overwrites whatever value the rules produced. A step with no external action applies the rules only.

D.2 CHEMISTRY

Domain Design The chemistry domain is inspired by multi-compartment vesicles in which reaction products move between compartments and initiate subsequent reactions (Elani et al., 2014). State-transition models have also been used to describe the qualitative behavior of biochemical reaction systems (Bugenhagen & Beard, 2012). Each state records the count of each species in each compartment. Reaction rules consume units of one species and produce the same number of units of

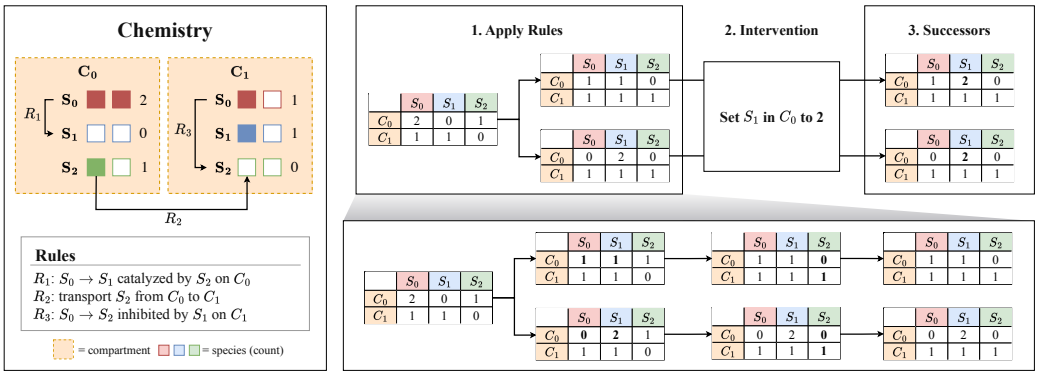

Figure 9: **Illustrative example of the Chemistry domain.** The example shows how reaction and transport rules are applied sequentially within a transition step. In C_0, catalyst S_2 enables R_1 to convert either one or two units of S_0 into S_1, producing two admissible branches. Rule R_2 then transports one unit of S_2 from C_0 to C_1, where R_3 is blocked by the presence of its inhibitor S_1. The intervention is applied last, setting the amount of S_1 in C_0 to two units in both branches. The lower panel shows the sequential rule checks in detail.

another within a compartment, while transport rules move units of a species between compartments. A reaction may require a catalyst to be present or an inhibitor to be absent in the same compartment. Figure 9 provides a concrete example of these dynamics, illustrating how the domain rules and intervention produce a state transition.

Exact Prompt Specification The following box reproduces the exact domain specification used in the benchmark. This text is inserted into the WORLD MODEL section of the full user prompt provided in Appendix E.2.

World Model for Chemistry

Entities

- "central chamber" (C0): a compartment.
- "outer chamber" (C1): a compartment.
- "quorene" (S0): a species.
- "azurene" (S1): a species.
- "ionex" (S2): a species.
- Each compartment holds 0 to 2 units of each species.

Dynamics
Every rule has the following form:

- Reaction: in one compartment, one unit of a species is consumed and one unit of another species is produced. A reaction may involve at most one catalyst or one inhibitor, which is a third species, different from the two in the reaction, in the same compartment: with a catalyst, the reaction occurs only if that species is present there; with an inhibitor, the reaction occurs only if that species is absent there.
- Transport: one unit of a species moves from one compartment to another.
- A reaction or transport that would take the amount of a species below 0 or above 2 does not occur.
- No two rules are identical.

State transitions occur in this exact order within each step:

1. The rules are checked once each, in the order they are listed: all public rules first, then the additional rules. Each rule is checked against the current state as changed by the rules before it, and if its condition holds it is applied immediately.

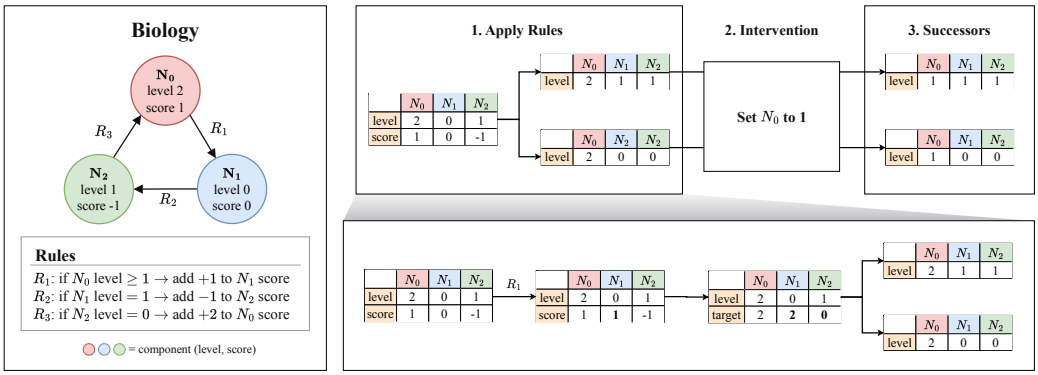

Figure 10: **Illustrative example of the Biology domain.** The example shows how score-based targets, asynchronous updating, and a post-update intervention determine the successor states. Starting from $(N_0, N_1, N_2) = (2, 0, 1)$ with base scores $(1, 0, -1)$, only R_1 holds, so the score of N_1 rises from 0 to 1. The target levels are therefore $(2, 2, 0)$. Both N_1 and N_2 differ from their targets, but only one component moves per step, so there are two endogenous successors, $(2, 1, 1)$ and $(2, 0, 0)$. The external intervention then sets N_0 to 1 in both branches.

2. When a rule applies, one unit reacts or moves. If two units could (two units to consume and room for two units on the other side), then either one or two units react or move, and each case yields one admissible successor state.
3. If there is an external intervention in this step, it is applied last (see Interventions).

Interventions
An intervention sets the amount of one species in one compartment to a specified value. It is applied at the end of the step, after all rules have been applied, and overwrites whatever amount the rules produced. A step with no external action applies the rules only.

D.3 BIOLOGY

Domain Design Gene regulatory networks are commonly modeled as qualitative logical networks (Abou-Jaoudé et al., 2016). Genes take discrete expression levels and regulate one another through thresholded activating or repressing interactions. Because the response times of different genes are unknown, updates are typically asynchronous (Abou-Jaoudé et al., 2016). Each step moves one gene one level toward its target (Thomas, 1991). Which gene responds first yields multiple admissible successors. Interventions correspond to expression perturbations such as knockdown or overexpression (Replogle et al., 2022). They set a gene's level while leaving the regulatory rules intact. Figure 10 provides a concrete example of these dynamics, illustrating how the domain rules and intervention produce a state transition.

Exact Prompt Specification The following box reproduces the exact domain specification used in the benchmark. This text is inserted into the WORLD MODEL section of the full user prompt provided in Appendix E.2.

World Model for Biology

Entities

- "regulator alpha" (N0): a component with a level, which is 0, 1, or 2, and a base score, which is fixed for the whole experiment and given with the initial state.
- "switch delta" (N1): a component with a level, which is 0, 1, or 2, and a base score, which is fixed for the whole experiment and given with the initial state.
- "response gamma" (N2): a component with a level, which is 0, 1, or 2, and a base score, which is fixed for the whole experiment and given with the initial state.

- The components form a cycle in the order listed: each component regulates the next, and the last regulates the first.

Dynamics
Every rule has the following form:

- Exactly one condition and exactly one effect.
- Condition: the level of one component is at least, at most, or exactly one of 0, 1, or 2.
- Effect: add -2, -1, +1, or +2 to the regulatory score of the component that follows the condition's component in the cycle.
- No two rules are identical.

State transitions occur in this exact order within each step:

1. Each component's regulatory score is reset to its base score; then every rule whose condition holds at the start of the step is applied simultaneously. If multiple rules affect the same component, their contributions are added together.
2. The target level of a component is 0 if its regulatory score is negative, 1 if the score is zero, and 2 if the score is positive. A component whose current level differs from its target level may move one level toward the target. If several components can move, exactly one of them is chosen and moves; each choice yields one admissible successor state. If no component can move, the state is unchanged.
3. If there is an external intervention in this step, it is applied last (see Interventions).

Interventions
An intervention sets the level of one component to a specified value. It is applied at the end of the step, after all rule applications and level changes, and overwrites whatever level the step produced. Base scores are never changed by interventions. A step with no external action applies the rules only.

E Benchmark Construction Details

E.1 Domain Complexity

Table 8 summarises the size of the hypothesis space behind each of the twelve panels. The first four columns are fixed by the domain. *States* is the number of joint configurations of the world's variables: physics has two integer quantities in $[-2, 2]$ and one binary mode ($5 \times 5 \times 2 = 50$); chemistry tracks the amount (0–2) of three species in two compartments ($3^6 = 729$); biology tracks a level and a base score for each of three components ($3^6 = 729$). *Rules* is the number of distinct, canonical rules the domain grammar admits after removing rules that are dead (never fire or never change the state) or non-canonical; it is 320 for physics, whose condition/effect grammar is a free product over quantities and constants, 108 for biology, and only 42 for chemistry, whose reactions and transports are constrained to one consumed and one produced unit with at most one catalyst or inhibitor. *Experiments* counts the intervention sequences over the three steps of an experiment, and *Queries* the propositions about the final state that an experiment can ask.

The remaining columns describe the abduction problem. Each panel fixes R_{obs} public rules and hides R_{hid} additional ones drawn from the same grammar. *Hidden Dynamics* is $|\mathcal{H}|$, the number of behaviourally distinct completions of a public rule set: hidden-rule combinations whose full theories induce identical transition functions are merged into one hypothesis, and we report the median over the 150 public rule sets of a panel. Because different rules often induce the same behaviour once the public rules are fixed, $|\mathcal{H}|$ is smaller than the raw number of combinations, by a factor that depends on the domain: roughly 70% of physics combinations and almost all chemistry combinations remain distinct with one hidden rule, but only 25–35% of biology combinations and 45% of physics combinations do with two. *Dynamics* is the total number of distinct transition functions across all public rule sets of the panel, i.e. the size of the world space from which problems are sampled. A second hidden rule enlarges the hypothesis space by two orders of magnitude in every domain, from tens or hundreds to thousands (biology, chemistry) or 22,000 (physics). $|\mathcal{V}(E)|$ *Med.* is the median, over the 100 problems of a panel, of the version space: the number of hypotheses in $\mathcal{H}$ consistent with the five observed experiments. $|\mathcal{V}(E)|/|\mathcal{H}_D|$ *Med.* divides it by $|\mathcal{H}|$ and measures how much the

Table 8: Dataset statistics across panels.

Domain	States	Rules	Experiments	Queries	$\mathbf{R_{obs}}$	$\mathbf{R_{hid}}$	Hidden Dynamics	Dynamics	$\lvert\mathcal{V}(E)\rvert$ Med.	$\lvert\mathcal{V}(E)\rvert/\lvert\mathcal{H}_D\rvert$ Med.
Physics	50	320	2379	20	2	1	219	32.8k	4	1.8%
					2	2	22,506	3.36M	73	0.34%
					3	1	221	32.8k	4	1.8%
					3	2	22,673	3.35M	118	0.49%
Chemistry	729	42	2379	30	2	1	41	5,981	1	2.6%
					2	2	1,138	166k	4	0.37%
					3	1	38	5,742	1	2.8%
					3	2	1,018	153k	5	0.46%
Biology	729	108	1110	15	2	1	25	3,750	2	8.3%
					2	2	1,035	155k	9.5	1.0%
					3	1	60	8,904	2	3.6%
					3	2	1,381	206k	10	0.7%

evidence must narrow the hypothesis space; it is the difficulty axis we use in Section 5.3. Chemistry problems are identified almost uniquely by five experiments (median version space of one with a single hidden rule), biology leaves a handful of alternatives, and physics with two hidden rules leaves around a hundred, yet those hundred are a smaller fraction of a much larger space (0.34–0.49%). By construction every hypothesis in the version space assigns the same label to every forecast, so a large version space does not make the forecasts ambiguous; it only means that the abductive target is a class of behaviourally equivalent theories rather than a single rule set.

E.2 Prompt Construction

System Prompt

Reason only from the fictional scientific system in the user message. Do not use web search, retrieval, code execution, calculators, or external solvers. Give a self-contained answer in your own words.

User Prompt (HR)

TASK
Abductive forecasting. Use the observed experiments to reason about the unknown additional rule(s), then classify the statement of every forecast experiment.
Entity names are arbitrary identifiers and carry no semantic meaning. This problem is self-contained: use only the rules stated here.

MODAL INTERPRETATION
Use the following meanings exactly:

- **Impossible**: no admissible evolution makes the statement true.
- **Possible but not inevitable**: some, but not all, admissible evolutions make the statement true.
- **Inevitable**: every admissible evolution makes the statement true.

WORLD MODEL
[Domain-specific world model: state variables, rule grammar, and transition procedure. The exact text for each domain is given in Appendix D.]

PUBLIC RULES
The following rules are known to be true.

- **B00**: if "probe coordinate" is at least 0, set "boundary regime" to active.
- **B01**: if "constraint coordinate" is at most 1, add +2 to "constraint coordinate".

In addition to the public rules above, exactly one **additional rule** of the form described in WORLD MODEL exists. It is not directly given and must be inferred from the observed experiments.

OBSERVED EXPERIMENTS

d00
Initial state

- Initial quantities: "probe coordinate"=-2, "constraint coordinate"=-2.
- Initial modes: "boundary regime"=active.

Interventions

- Step 0: set "constraint coordinate" to -2
- Step 1: set "probe coordinate" to -1
- Step 2: set "probe coordinate" to 1

Execution
Run exactly 3 steps.
Observed result
Impossible: at the final step, "constraint coordinate" equals 1.

FORECAST EXPERIMENTS

f00
Initial state

- Initial quantities: "probe coordinate"=-1, "constraint coordinate"=0.
- Initial modes: "boundary regime"=inactive.

Interventions

- Step 0: set "boundary regime" to active
- Step 1: set "boundary regime" to inactive
- Step 2: set "probe coordinate" to 0

Execution
Run exactly 3 steps.
Statement
At the final step, "constraint coordinate" is at most 0.
f09
Initial state

- Initial quantities: "probe coordinate"=-1, "constraint coordinate"=1.
- Initial modes: "boundary regime"=inactive.

Interventions

- Step 0: set "constraint coordinate" to -1
- Step 1: set "probe coordinate" to 2
- Step 2: set "boundary regime" to inactive

Execution
Run exactly 3 steps.
Statement
At the final step, "constraint coordinate" equals 0.

REQUIRED DECISION
Using the public rules and the observed experiments, classify the Statement of every forecast experiment under the admissible rules consistent with the evidence, as exactly one of:

- `impossible`
- `possible_not_inevitable`
- `inevitable`

OUTPUT FORMAT
Return only:

```
{
  "forecasts": {
    "f00": "<classification>",
    ...
    "f09": "<classification>"
  }
}
```

Constraints:

- Include every forecast ID exactly once (f00 through f09).
- Do not include explanations.
- Output valid JSON only.

User Prompt (CR)

TASK
Forecasting with supplied rules. Use the public rules together with the supplied additional rule(s) to classify the statement of every forecast experiment.
Entity names are arbitrary identifiers and carry no semantic meaning. This problem is self-contained: use only the rules stated here.

MODAL INTERPRETATION
Use the following meanings exactly:

- **Impossible**: no admissible evolution makes the statement true.
- **Possible but not inevitable**: some, but not all, admissible evolutions make the statement true.
- **Inevitable**: every admissible evolution makes the statement true.

WORLD MODEL
[Domain-specific world model: state variables, rule grammar, and transition procedure. The exact text for each domain is given in Appendix D.]

PUBLIC RULES
The following rules are known to be true.

- **B00**: if "probe coordinate" is at least 0, set "boundary regime" to active.
- **B01**: if "constraint coordinate" is at most 1, add +2 to "constraint coordinate".

In addition to the public rules above, exactly one **additional rule** of the form described in WORLD MODEL exists, and it is supplied below.

SUPPLIED ADDITIONAL RULES
The following additional rule(s) hold together with the public rules.

- **A00**: if "probe coordinate" equals -1, add -2 to "constraint coordinate".

OBSERVED EXPERIMENTS
d00
Initial state

- Initial quantities: "probe coordinate"=-2, "constraint coordinate"=-2.
- Initial modes: "boundary regime"=active.

Interventions

- Step 0: set "constraint coordinate" to -2
- Step 1: set "probe coordinate" to -1
- Step 2: set "probe coordinate" to 1

Execution
Run exactly 3 steps.
Observed result
Impossible: at the final step, "constraint coordinate" equals 1.

FORECAST EXPERIMENTS
f00
Initial state

- Initial quantities: "probe coordinate"=-1, "constraint coordinate"=0.
- Initial modes: "boundary regime"=inactive.

Interventions

- Step 0: set "boundary regime" to active
- Step 1: set "boundary regime" to inactive
- Step 2: set "probe coordinate" to 0

Execution
Run exactly 3 steps.
Statement
At the final step, "constraint coordinate" is at most 0.

f09
Initial state

- Initial quantities: "probe coordinate"=-1, "constraint coordinate"=1.
- Initial modes: "boundary regime"=inactive.

Interventions

- Step 0: set "constraint coordinate" to -1
- Step 1: set "probe coordinate" to 2
- Step 2: set "boundary regime" to inactive

Execution
Run exactly 3 steps.
Statement
At the final step, "constraint coordinate" equals 0.

REQUIRED DECISION
Using the public rules and the supplied additional rule(s), classify the Statement of every forecast experiment as exactly one of:

- `impossible`
- `possible_not_inevitable`
- `inevitable`

OUTPUT FORMAT
Return only:

```
{
  "forecasts": {
    "f00": "<classification>",
    ...
    "f09": "<classification>"
  }
}
```

Constraints:

- Include every forecast ID exactly once (f00 through f09).
- Do not include explanations.
- Output valid JSON only.