arXiv:2602.18843v3 [cs.AI] 3 May 2026

ABD: Default–Exception Abduction in Finite First-Order Worlds

Serafim Batzoglou*

Abstract

Abduction in knowledge representation is often framed as "explaining away" inconsistencies between a background theory and observations by hypothesizing missing facts or exceptions. Despite decades of KR work on abduction, there are few modern benchmarks with exact, solver-checkable semantics for multi-world first-order exception synthesis. We introduce ABD, a solver-checkable benchmark for default–exception abduction over small finite relational worlds, and use it here for a diagnostic evaluation of frontier language models. Each instance provides (a) a set of finite structures with observed facts, and (b) a fixed default-like first-order theory that may be violated by those observations. A model must output a first-order *abnormality rule* $\alpha(x)$ whose substitution for Ab restores satisfiability while keeping exceptions sparse.

We formalize three observation regimes with distinct completion semantics. **ABD-Full** assumes closed-world observation. **ABD-Partial** allows unknown atoms under *existential* completion: α is valid if *some* completion makes the repaired theory satisfiable, with cost optimized in the best case. **ABD-Skeptical** uses *universal* completion: α is valid only if the repaired theory is satisfiable under *every* completion, with cost measured in the worst case.

Because domains are finite, validity and costs are computed via SMT (Z3), enabling exact verification and controlled difficulty. In this paper, we use ABD to evaluate eleven frontier LLMs on 600 instances spanning all three scenarios and seven default theories. Among the strongest high-validity models, prompt-set cost gaps of $\sim$1–1.5 extra exceptions per world remain, and valid-formula AST stays in the low teens, while GPT-5.4 is a low-gap, high-AST outlier with lower validity. Holdout evaluation reveals distinct generalization profiles: in ABD-Full and ABD-Partial, the dominant failure is *parsimony inflation*; in ABD-Skeptical, it is *validity brittleness*—rules that work on prompt worlds often break on holdouts—while survivors show smaller gap inflation.

1 Introduction

Default reasoning is a cornerstone of knowledge representation: we often model domains with rules that hold "normally," while allowing rare exceptions. In classical KR formalisms—default logic (Reiter 1980), circumscription (McCarthy 1980), and stable-model semantics (Gelfond and Lifschitz 1988)—exceptions are not bugs but a feature: they encode domain irregularities without abandoning general rules. From the perspective of *abduction* (Poole 1989; Eiter and Gottlob 1995), exceptions are precisely what we infer when observations conflict with what a default theory predicts.

Recent work shows that LLMs can emit first-order logical forms and support logic-theory induction under formal feedback (Pei, Du, and Jin 2025; Gandarela, Carvalho, and Freitas 2025). *Can such models propose correct, compact exception rules that repair default theories across multiple relational worlds?* Answering this requires an evaluation setting that removes natural-language ambiguity, supports quantifiers, and remains mechanically checkable.

We propose ABD, a solver-checkable benchmark for default–exception abduction on finite first-order worlds. The benchmark is general in the sense that any system that outputs a first-order abnormality rule can be evaluated on it; the empirical focus here is a diagnostic study of frontier language models. Each instance consists of several small relational structures ("worlds") and a fixed first-order background theory Θ that uses an abnormality predicate Ab to block defaults. Rather than providing labeled targets, we provide only the theory and the observed facts; the model must output a definition of abnormality—a first-order formula $\alpha(x)$ such that substituting α for Ab repairs the theory across all prompt worlds. Evaluation then combines validity with a parsimony objective: among valid hypotheses, we prefer rules that mark as few abnormal domain elements as possible.

The output $\alpha(x)$ is therefore best viewed as a shared repair policy for a default theory over multiple relational worlds. The task is to restore consistency with minimal exceptional structure, not to recover a unique planted target formula.

Three Observation Regimes. We study three scenarios that share the same interface but differ in how missing information is handled: **ABD-Full** (closed-world), **ABD-Partial** (existential completion: valid if *some* completion satisfies the repaired theory), and **ABD-Skeptical** (universal completion: valid only if *every* completion satisfies the repaired theory). Parsimony mirrors these quantifiers: best-case cost in ABD-Partial, worst-case in ABD-Skeptical. These regimes are formalized in Section 4.

Evaluation Beyond Validity. Binary validity quickly saturates for stronger models. However, validity is in principle trivial in this task: a model can simply label many (or all)

*Contact: serafim.batzoglou@gmail.com

elements as abnormal. We therefore emphasize *cost-based* metrics—how close a model comes to a solver-computed lower bound on abnormality count—and *size-conditioned* variants that group formulas by syntactic complexity (AST size) to discourage degenerate case-splitting.

Main Empirical Findings. We evaluate eleven frontier models across all three regimes. Holdout evaluation reveals two distinct failure modes: *parsimony inflation* in ABD-Full/Partial (gaps roughly double on fresh worlds) and *validity brittleness* in ABD-Skeptical (rules that work on prompt worlds often break on holdouts, though survivors show smaller gap inflation). The models also separate into distinct performance profiles: Opus-4.6, Gemini-3.1, DSR, and Grok-4.1-fast (Grok4.1f) form a high-validity cluster; GPT-5.4 attains the best gap metrics but with very large formulas and weak holdout survival; Kimi-K2-thinking (Kimi-K2t) is comparatively compact and conditionally robust but substantially less parsimonious. Details are in Section 6.

Contributions.

- We formalize default–exception abduction on finite first-order worlds with solver-checkable semantics under three observation regimes: ABD-Full (closed-world, fully observed), ABD-Partial (existential completion under missing atoms), and ABD-Skeptical (universal completion with worst-case, robust exception costs).
- We introduce cost-based scoring for parsimony, including gap-to-lower-bound, reference-relative gap, and size-conditioned analyses by formula complexity.
- We describe a difficulty-controlled generator that constructs multi-world instances and eliminates shortcut hypotheses via a lightweight, counterexample-guided (CEGIS-like) procedure.
- We evaluate eleven frontier models—Opus-4.6, Grok4, Grok4.1f, Gemini-3.1, Gemini-3, GPT-5.2, GPT-5.4, DeepSeek-Reasoner (DSR), Kimi-K2t, GPT-4o, and Hermes4—and analyze validity, parsimony gaps, formula complexity, and holdout generalization.

2 Related Work

Abduction, Defaults, and Non-Monotonic Reasoning. Abduction is often formalized as selecting hypotheses that, when added to background knowledge, entail observations while satisfying integrity constraints, typically with a preference for parsimonious explanations (Poole 1989; Kakas, Kowalski, and Toni 1992; Eiter and Gottlob 1995). This connects directly to non-monotonic KR frameworks for defeasible rules and exceptions, including default logic and circumscription (Reiter 1980; McCarthy 1980), and stable-model/answer-set semantics that support explicit minimization principles (including "minimize abnormalities" encodings) (Gelfond and Lifschitz 1988; Baral 2003; Brewka, Eiter, and Truszczyński 2011; Gebser et al. 2012). Recent benchmark work has also tested LLMs directly on ASP solving, showing that full answer-set computation remains substantially harder than ASP entailment or answer-set verification (Ren et al. 2025). Our benchmark adopts a solver-friendly abnormality-predicate encoding with finite grounding, but evaluates more than satisfiability: we quantify *parsimony gaps* between a model's proposed exception rule and a solver-computed minimum-exception lower bound, and we test whether the learned exception policy remains valid on holdout worlds.

Abductive Logic Programming and Diagnosis. Abductive logic programming (ALP) provides a general framework for explanation under integrity constraints and has been developed in both logic-programming and KR traditions (Kakas, Kowalski, and Toni 1992; Denecker and Kakas 2002). Model-based diagnosis similarly treats faults/abnormalities as latent causes chosen to restore consistency, often with minimality criteria and hitting-set structure (Reiter 1987; de Kleer and Williams 1987; Hamscher, Console, and de Kleer 1992). Our setting differs in interface and emphasis: the hypothesis is an explicit *first-order definition* of an abnormality predicate (not a set of abduced ground facts), and it must satisfy a *multi-world* specification, which makes it possible to separate repair-by-case from compact exception structure that generalizes.

Inductive Logic Programming and Relational Learning. ILP learns relational rules from examples and background knowledge, with classic systems and formalisms for inducing Horn clauses and relational definitions (Muggleton 1991; Muggleton and Raedt 1994; Quinlan 1990; Srinivasan 2001; Cropper and Muggleton 2020; Raedt 2008). Recent work has also studied LLMs as inductive theory learners under formal inference-engine feedback and graded target expressivity (Gandarela, Carvalho, and Freitas 2025), providing a complementary LLM-centered analysis of logic theory induction. While our hypotheses are symbolic and relational, our tasks are not supervised concept learning: each instance provides observations plus a default theory, and success depends on proposing an exception rule that restores satisfiability while minimizing abnormalities. This shifts the challenge from matching labels to balancing consistency and parsimony under an explicit non-monotonic objective, with exact verification. Synthesizing $\alpha(x)$ from multi-world specifications is also amenable to symbolic ILP and ASP-based learners such as Progol (Muggleton 1995) and ILASP (Law, Russo, and Broda 2014), which can search a hypothesis space with completeness guarantees relative to a language bias. The benchmark is meaningful beyond LLMs, but our empirical scope here is LLM diagnosis rather than an exhaustive comparison across symbolic, ILP-, ASP-, or neuro-symbolic systems.

Program Synthesis and Solver-Aided Search. Our tasks can be viewed as constrained synthesis: produce a symbolic artifact that satisfies a solver-checkable specification. This aligns with solver-aided languages and synthesis paradigms such as Sketch, CEGIS, and syntax-guided synthesis (Solar-Lezama et al. 2006; Solar-Lezama 2008; Alur et al. 2013; Torlak and Bodík 2014). A key difference is that our specifications are *multi-world* and explicitly *costed*: hypotheses must simultaneously repair multiple structures, and evaluation tracks both validity and the gap between a model's abnormality cost and a solver-computed lower bound.

Logic and Relational Reasoning Benchmarks. A broad range of benchmarks probe reasoning with rules, quantifiers,

and structured knowledge, often through natural-language interfaces (Tafjord, Dalvi, and Clark 2021; Han et al. 2024; Clark, Tafjord, and Richardson 2020). Such settings are valuable but can conflate logical competence with linguistic priors and make intermediate steps hard to verify. Recent work has also introduced InAbHyD, a synthetic benchmark for inductive and abductive reasoning with an Occam-style hypothesis metric over incomplete world models (Sun and Saparov 2025). Our benchmark instead presents extensional finite structures and uses solver-backed grounding to make correctness and costs exact, enabling diagnostics about exception sparsity, gap above a solver lower bound, and overfitting across holdout worlds.

Mathematical Reasoning and Formal Proof Benchmarks. Natural-language math benchmarks such as GSM8K and MATH test multi-step reasoning but remain mediated by natural-language problem statements and outputs (Cobbe et al. 2021; Hendrycks et al. 2021). Formal proof benchmarks (e.g., CoqGym, HOList, miniF2F, ProofNet, ProofGrid) evaluate mechanically verified outputs in proof assistants or formal proof languages (Yang and Deng 2019; Bansal et al. 2019; Zheng, Han, and Polu 2022; Azerbayev et al. 2023; Arkoudas and Batzoglou 2025), and large-scale theorem-proving efforts use learning to guide deductive search (Polu and Sutskever 2020). Our benchmark is complementary: rather than proving theorems, models must *synthesize compact exception definitions* that reconcile defaults with observations under a minimum-abnormality objective.

Neuro-Symbolic Learning and Large Language Models. Neuro-symbolic work explores combining learning with symbolic representations and constraint-based verification (d'Avila Garcez and Lamb 2023), including approaches that use abduction as a learning signal (Dai et al. 2019). Relatedly, FoVer translates natural-language reasoning into first-order logic and uses automated logical verification to check logical correctness, illustrating the value of verification-backed evaluation for LLM reasoning claims (Pei, Du, and Jin 2025). Recent LLM analyses highlight that models can produce syntactically plausible symbolic outputs while relying on shortcuts or overly long solutions when unconstrained (Wei et al. 2022; Dziri et al. 2023). Our evaluation emphasizes solver-verifiable semantics, parsimony-sensitive metrics (cost and gap), and holdout testing designed to expose hypotheses that repair observed worlds without capturing a stable general exception rule.

Abductive Reasoning in Natural Language. Abduction has also been studied in NLP as a model of commonsense explanation and narrative understanding, including classic logical accounts of interpretation-as-abduction (Hobbs et al. 1993) and benchmark-style tasks for causal/abductive inference (Roemmele, Bejan, and Gordon 2011; Mostafazadeh et al. 2016; Bhagavatula et al. 2020). These tasks are valuable for modeling everyday explanation but are mediated by language and typically lack a fully explicit world model with exact cost verification. By contrast, our benchmark isolates abductive structure in a formal finite-model setting where validity and costs are mechanically checkable and have a direct interpretation as exception cardinality.

3 Problem Setup

Finite Relational Worlds. We fix a relational signature $\Sigma = \{P, Q, R, S, =\}$, with unary predicates $P(\cdot), Q(\cdot)$, binary predicates $R(\cdot,\cdot), S(\cdot,\cdot)$, and equality. A *world* W has a finite domain $D_W = \{a_1, \ldots, a_n\}$ ($|D_W| \in \{9{-}12\}$; see Table 1) and (possibly partial) interpretations for the predicates in Σ. In the fully observed setting (ABD-Full), W specifies complete interpretations $P^W, Q^W \subseteq D_W$ and $R^W, S^W \subseteq D_W \times D_W$. In the partial-observation settings (ABD-Partial and ABD-Skeptical), W additionally designates a set of *unknown* ground atoms Ω_W whose truth values are not observed. In our benchmark, Ω_W ranges over ground atoms of R and S; P and Q are always fully observed. In ABD, a world does *not* include labels for a target concept. This fixed signature and small-domain setting is deliberate: it keeps all three semantics groundable and SMT-checkable, so our claims are about this finite-world benchmark family rather than arbitrary first-order settings.

Background Theories with Abnormality. Each instance specifies a fixed first-order theory Θ over an extended signature $\Sigma_{\mathsf{Ab}} = \Sigma \cup \{\mathsf{Ab}\}$, where $\mathsf{Ab}(\cdot)$ is a unary abnormality predicate. We use Θ to express default-like constraints that may be blocked by abnormality. A common pattern is an implication schema of the form

$$\forall x \, \Big(\mathrm{Ante}(x) \, \wedge \, \neg \mathsf{Ab}(x) \, \rightarrow \, \mathrm{Cons}(x) \Big), \tag{1}$$

where Ante and Cons are first-order formulas over Σ (often involving existentials over R, S). Intuitively, elements satisfying the antecedent are expected to satisfy the consequent *unless* they are declared abnormal.

Hypotheses as Definitions of Abnormality. A model outputs a single first-order formula $\alpha(x)$ (one free variable) over a restricted set of allowed predicates (specified per instance). We treat the prediction as a meta-level definition of abnormality:

$$\mathsf{Ab} := \alpha. \tag{2}$$

Equivalently, $\Theta[\mathsf{Ab} \mapsto \alpha]$ denotes the repaired theory obtained by substituting the instantiated formula $\alpha(t)$ for each atom $\mathsf{Ab}(t)$ in Θ. Example: under T1, suppose two prompt worlds contain violating objects a and c, and in each world the violating object also has an S-successor. If S is allowed in the hypothesis language, a shared rule such as $\alpha(x) := \exists z \, S(x, z)$ repairs both worlds jointly, illustrating that the task is to synthesize one shared rule, not one rule per world.

Symbolic Output Language. To support exact evaluation, $\alpha(x)$ is produced in a simple S-expression syntax. This syntax provides a canonical I/O format: prefix notation avoids precedence and associativity ambiguity and is robust inside single-line JSON outputs. The grammar admits boolean connectives (`and`, `or`, `not`, `implies`), first-order quantifiers (`forall`, `exists`), and atomic predicates from the allowed signature. We use AST size, or formula size, to mean the number of nodes in the parsed formula tree of α: each operator, predicate, variable, and constant counts as one node, and each quantifier contributes one node for the binder plus one for its bound variable.

Unknown Atoms and Completions. In partial worlds, each ground atom of R and S is either *known true*, *known false*, or *unknown*. Let Ω_W be the set of unknown atoms in W. A *completion* $\mathcal{C}$ assigns a truth value to every atom in Ω_W, yielding a fully specified world $W^{\mathcal{C}}$. We write $\mathrm{KnownFacts}(W)$ for the conjunction of all observed positive facts and observed negative facts in W (treating atoms not listed as true or unknown as observed false). In fully observed worlds, $\mathrm{KnownFacts}(W)$ coincides with $\mathrm{Facts}(W)$. For a completion $\mathcal{C}$, we take $\mathrm{Facts}(W^{\mathcal{C}})$ to be $\mathrm{KnownFacts}(W)$ together with the literals that fix each atom in Ω_W according to $\mathcal{C}$.

Finite Grounding and Solver Checks. All domains are finite, so satisfiability of Θ under a predicted α can be checked by grounding quantifiers over D_W and translating the result to SMT. In partial worlds, unknown atoms in Ω_W are represented as free Boolean variables in the SMT encoding, so we can quantify over completions by changing which constraints are asserted. Existential-completion validity (ABD-Partial) reduces to a satisfiability check with unknowns left free, whereas skeptical validity (ABD-Skeptical) reduces to the absence of a counterexample completion (checked as a single UNSAT query that asks whether there exists a completion consistent with $\mathrm{KnownFacts}(W)$ under which some grounded axiom fails in the repaired theory $\Theta[\mathsf{Ab} \mapsto \alpha]$). Because costs are cardinalities over a finite domain, we can compute parsimony objectives exactly (or with thresholded SAT checks) using Z3 (de Moura and Bjørner 2008). Throughout, Z3 is the exact verifier and cost evaluator for these finite grounded instances.

We formalize the existential and universal completion semantics as synthesis problems in Section 4.

4 Default–Exception Abduction Tasks

We now define the three ABD task variants as formal synthesis problems. Fix an instance $\mathcal{I}$ with background theory Θ and prompt worlds $\mathcal{W}_{\mathrm{p}} = \{W_1, \ldots, W_k\}$. The learner outputs $\alpha(x)$ and is evaluated by (i) validity and (ii) parsimony.

4.1 ABD-Full: Abduction Under Full Observation

In ABD-Full, each world W provides a complete interpretation of P, Q, R, S under a closed-world assumption for the given domain (any unlisted atom is false). We write $\mathrm{SAT}(\varphi)$ to mean that the finite grounded formula φ is satisfiable. A hypothesis α is *valid* on a world W if the repaired theory is satisfiable:

$$W \models_{\mathrm{ABD}} \alpha \iff \mathrm{SAT}\big(\Theta[\mathsf{Ab} \mapsto \alpha] \wedge \mathrm{Facts}(W)\big). \quad (3)$$

An α solves an instance iff it is valid on *all* prompt worlds.

Cost. For a valid hypothesis, we define the per-world abnormality count $\mathrm{cost}_W(\alpha) = |\{a \in D_W : W \models \alpha[a]\}|$. The instance-level cost is the sum across worlds:

$$\mathrm{Cost}(\alpha; \mathcal{W}_{\mathrm{p}}) = \sum_{W \in \mathcal{W}_{\mathrm{p}}} \mathrm{cost}_W(\alpha). \quad (4)$$

Lower-Bound Baseline and Gap. To measure parsimony, we compare a model's cost to a solver-computed lower bound obtained by allowing Ab to be assigned freely (not necessarily definable by a single α). Concretely, for each world we compute

$$\mathrm{OptCost}(W) = \min_{A \subseteq D_W} |A| \quad \text{s.t. } \mathrm{SAT}\big(\Theta[\mathsf{Ab} \mapsto A] \wedge \mathrm{Facts}(W)\big), \quad (5)$$

where $\Theta[\mathsf{Ab} \mapsto A]$ means that each grounded atom $\mathsf{Ab}(a)$ is set true iff $a \in A$, and define $\mathrm{OptCost}(\mathcal{W}_{\mathrm{p}}) = \sum_{W \in \mathcal{W}_{\mathrm{p}}} \mathrm{OptCost}(W)$. We report the *gap* $\mathrm{Gap}(\alpha) = \mathrm{Cost}(\alpha; \mathcal{W}_{\mathrm{p}}) - \mathrm{OptCost}(\mathcal{W}_{\mathrm{p}})$. Gap is defined only for valid hypotheses; invalid outputs contribute to validity/error rates but are excluded from gap averages. Because $\mathrm{OptCost}$ relaxes the single-formula constraint and allows Ab to vary freely by world, it lower-bounds the best *definable* cost; thus $\mathrm{Gap}(\alpha)$ conservatively overestimates suboptimality. On small domains the bound can often be matched by large extensional case-splits, motivating our AST- and holdout-based diagnostics.

This baseline is nevertheless solver-verifiable, inexpensive compared to full joint optimization, and empirically discriminative across models.

4.2 ABD-Partial: Existential Completion

ABD-Partial introduces missing information about the observed predicates (Section 3). Recall that Ω_W is the set of unknown atoms in W and $W^{\mathcal{C}}$ denotes the completed world under completion $\mathcal{C}$.

Validity Under Existential Completion. A hypothesis α is valid on W if there exists a completion that makes the repaired theory satisfiable:

$$W \models_{\exists\mathrm{comp}} \alpha \iff \exists \mathcal{C} : \mathrm{SAT}\big(\Theta[\mathsf{Ab} \mapsto \alpha] \wedge \mathrm{Facts}(W^{\mathcal{C}})\big). \quad (6)$$

Cost Optimized Over Completions. Because α may mention unknown relations, its extension can depend on the completion. We define cost as the minimum abnormality count over completions that satisfy the repaired theory:

$$\mathrm{cost}^{\exists}_W(\alpha) = \min_{\mathcal{C}} |\{a \in D_W : W^{\mathcal{C}} \models \alpha[a]\}| \quad \text{s.t. } \mathrm{SAT}\big(\Theta[\mathsf{Ab} \mapsto \alpha] \wedge \mathrm{Facts}(W^{\mathcal{C}})\big). \quad (7)$$

Instance cost sums these per-world values, and gaps are defined relative to a completion-aware lower bound that also leaves Ab unconstrained:

$$\mathrm{OptCost}^{\exists}(W) = \min_{\mathcal{C},\, A \subseteq D_W} |A| \quad \text{s.t. } \mathrm{SAT}\big(\Theta[\mathsf{Ab} \mapsto A] \wedge \mathrm{Facts}(W^{\mathcal{C}})\big). \quad (8)$$

As in ABD-Full, the free-Ab lower bound is conservative (Section 4.1).

4.3 ABD-Skeptical: Universal Completion

ABD-Skeptical retains partial observation but treats unknown atoms conservatively. Intuitively, a hypothesis should repair

the default theory not only for *some* completion of missing facts, but for *all* completions consistent with what is observed.

Validity Under Universal Completion. A hypothesis α is *skeptically valid* on W if the repaired theory is satisfiable for every completion:

$$W \models_{\forall\text{comp}} \alpha \iff \forall \mathcal{C} : \mathrm{SAT}\big(\Theta[\mathsf{Ab} \mapsto \alpha] \wedge \mathrm{Facts}(W^{\mathcal{C}})\big). \quad (9)$$

An α solves an instance iff it is valid on *all* prompt worlds under this universal-completion semantics.

Worst-Case Cost Over Completions. For a skeptically valid hypothesis, the per-world cost is the *worst-case* abnormality count over completions:

$$\mathrm{cost}^{\forall}_{W}(\alpha) = \max_{\mathcal{C}} \ |\{a \in D_W : W^{\mathcal{C}} \models \alpha[a]\}|. \quad (10)$$

The instance-level cost is the sum across prompt worlds:

$$\mathrm{Cost}^{\forall}(\alpha; \mathcal{W}_{\mathrm{p}}) = \sum_{W \in \mathcal{W}_{\mathrm{p}}} \mathrm{cost}^{\forall}_{W}(\alpha). \quad (11)$$

Lower-Bound Baseline and Gap. We lift the free-Ab lower bound (Section 4.1) to the skeptical setting by taking the worst case over completions:

$$\mathrm{OptCost}^{\forall}(W) = \max_{\mathcal{C}} \ \mathrm{OptCost}(W^{\mathcal{C}}), \quad (12)$$

and define $\mathrm{OptCost}^{\forall}(\mathcal{W}_{\mathrm{p}}) = \sum_{W \in \mathcal{W}_{\mathrm{p}}} \mathrm{OptCost}^{\forall}(W)$. We report the skeptical gap

$$\mathrm{Gap}^{\forall}(\alpha) = \mathrm{Cost}^{\forall}(\alpha; \mathcal{W}_{\mathrm{p}}) - \mathrm{OptCost}^{\forall}(\mathcal{W}_{\mathrm{p}}). \quad (13)$$

This baseline is again conservative in the sense of Section 4.1.

4.4 Planted Generator References and Holdout Evaluation

Planted Generator References. Many instances are generated by selecting a planted generator reference abnormality rule $\alpha^{\star}$ from a controlled template family and constructing worlds for which $\alpha^{\star}$ is valid and nontrivial. We treat $\alpha^{\star}$ as a *generator artifact*, not as the unique correct answer: models may find different valid rules with lower or higher cost.

Reference-Relative Cost and "Beating the Reference." To aid analysis and to detect shortcut hypotheses, we report the gap to the planted generator reference: $\mathrm{Gap}_{\mathrm{ref}}(\alpha) = \mathrm{Cost}(\alpha) - \mathrm{Cost}(\alpha^{\star})$. Negative values indicate that the model found a rule with fewer exceptions than the planted one. We treat such cases as diagnostics that the planted generator reference is an anchor rather than an optimum.

Size-Conditioned Metrics. A model can drive cost down by producing large disjunctive case splits tailored to the finite worlds. To separate such behavior from compact reasoning, we report size-conditioned metrics that group formulas by AST size into bins (Table 6).

Holdout Worlds. To probe generalization, each instance also has $k{=}5$ *holdout worlds* pre-generated from the same world sampler and theory parameters as the prompt worlds. Unlike prompt worlds, holdout worlds satisfy the same per-world acceptance checks but are *not* adversarially hardened against competitor formulas. Holdout is therefore an *out-of-filter distribution refresh*: matched in sampling distribution but not adversarially hardened. Holdout worlds are pre-generated and cached for reproducibility. We report holdout validity and cost gaps, as well as $\Delta\text{Gap} = \text{H-Gap} - \text{P-Gap}$ (the gap degradation from prompt worlds to holdout). For ABD-Skeptical, holdouts additionally require that the planted generator reference satisfies universal-completion validity. Prompt and holdout distributions remain closely matched in world counts and per-world reference costs.

5 Dataset Generation

This paper emphasizes *controllable difficulty*: we want instances where simple exception rules fail, multiple worlds jointly constrain the solution, and models cannot rely on a single brittle shortcut. We therefore generate instances using three ingredients:

(i) Controlled Hypothesis Families. The planted generator reference is sampled from a theory-compatible template library of S-expression formulas. The library is organized by quantifier-depth tier: Tier 0 propositional unary formulas (QD 0), Tier 1 single-quantifier formulas (QD 1), and curated Tiers 2–3 nested-quantifier formulas, including family-specific variants for richer allowed signatures. For each theory, we retain only templates whose predicates lie in the allowed hypothesis signature, so difficulty is controlled jointly by tier and theory-specific predicate scope. A diversity tracker also limits repeated reuse of the same template within a theory$\times$difficulty bucket.

(ii) Multi-World Filtering and Parsimony Floors. Candidate prompt-world sets are filtered both per world and jointly across worlds. Per world, the planted generator reference must be valid, require at least one abnormal object ($\mathrm{OptCost} \geq 1$), remain near-optimal (reference cost $\leq \mathrm{OptCost} + 1$), and keep abnormalities sparse ($\mathrm{OptCost}/|D_W| \leq 0.20$). Aggregate difficulty refers to the corresponding multi-world totals, especially the summed lower-bound cost and summed reference cost across the prompt worlds, so that we reject instances whose joint repair problem is still too easy even if each world alone is nontrivial.

(iii) CEGIS-Like Elimination of Shortcut Competitors. We then harden the prompt worlds against shortcut explanations. For each instance we build a theory-compatible competitor pool from simple Tier 0/1 formulas, mined shortcut formulas from prior model outputs, and a small set of mutants of the planted generator reference. A competitor is called a survivor if it remains valid on every current prompt world and stays within a small total-cost margin of the planted reference (typically 2). When survivors remain, we add adversarial worlds that invalidate at least one survivor or make it sufficiently more costly, and accept the instance only when no survivors remain; otherwise the instance is rejected at the world-budget limit.

Table 1 summarizes the released benchmark.

Table 1: **Abduction Benchmark Summary.** (a) Dataset statistics by scenario. (b) Default theories used in the benchmark; each theory defines when $\mathsf{Ab}(x)$ blocks a default conclusion.

(a) Dataset Statistics

Scenario	N	Prompt	Holdout	$\lvert D\rvert$	Theories
FULL	195	9.0	5.0	9–11	T1, T2, T3, T4, T5
PARTIAL	243	8.3	5.0	9–11	T1, T2, T3, T4, T5
SKEPTICAL	162	6.0	5.0	10–12	T1, T2, T3, T4, T5, T6, T7
Total	**600**				

(b) Default Theories: $\phi(x) \wedge \neg\mathsf{Ab}(x) \rightarrow \psi(x)$

ID	Antecedent $\rightarrow$ Consequent
T1	$\exists y(R(x,y) \wedge P(y)) \rightarrow Q(x)$
T2	$\exists y(R(x,y) \wedge P(y)) \rightarrow \exists z(S(x,z) \wedge Q(z))$
T3	$\exists y(S(x,y) \wedge P(y)) \rightarrow \exists z(R(x,z) \wedge Q(z))$
T4	$\exists y(R(x,y) \wedge P(y)) \rightarrow \exists z(S(x,z) \wedge \forall w(R(z,w) \rightarrow P(w)))$
T5	$\exists y(R(x,y) \wedge P(y)) \rightarrow \forall z(S(x,z) \rightarrow Q(z))$
T6	$P(x) \rightarrow \exists y(R(x,y))$
T7	$P(x) \rightarrow \forall y(R(x,y) \rightarrow Q(y))$

6 Evaluation and Experiments

6.1 Prompting and Inference Setup

We employ a **zero-shot** prompting strategy: no benchmark instances or planted generator references appear as in-context examples. Each prompt consists of four components: (1) a *system instruction* establishing the role ("expert in first-order logic and abductive reasoning") and requiring JSON output; (2) a *scenario-specific task description* that defines the observation semantics (closed-world, existential, or universal completion), the S-expression grammar for $\alpha(x)$, the validity/parsimony/simplicity objectives, and illustrative grammar examples; (3) the *instance specification*, comprising the allowed and forbidden predicates, the theory axioms, and the prompt worlds with their predicate interpretations (and, for ABD-Partial/Skeptical, unknown atoms); (4) an *output format block* requesting a single JSON line with the S-expression formula and a natural-language description. The prompt includes structured reasoning guidelines ("identify which objects must be abnormal; look for a pattern; verify against all worlds") but instructs models to reason internally and emit only the final JSON answer.

All models are queried at temperature $T = 0.1$ for near-deterministic outputs. For models offering a thinking mode, we used the provider-default reasoning configuration with 64K token budget; other models were queried in standard generation mode. We do not tune few-shot exemplars, model-specific chain-of-thought scaffolds, or bespoke reasoning prompts. Each of the 600 benchmark instances is presented independently (single-turn, no multi-round refinement). All results correspond to API snapshots from January–March 2026 (through March 7 for GPT-5.4); model behavior may differ in future versions.

Each model outputs a candidate $\hat{\alpha}(x)$ in S-expression syntax. We compute prompt validity and exception cost as defined in Section 4, with cost semantics varying by scenario. All gap values reported are *normalized* by the number of worlds to keep scores comparable across instances. Accordingly, prompt-set Gap averages only over prompt-valid predictions, holdout Gap averages only over holdout-valid predictions, and ΔGap is computed only on survivors valid on both prompt and holdout worlds. All validity percentages in the main tables are instance-level, not per-world. Each evaluation record corresponds to one model call on one benchmark instance, so a prediction counts as prompt-valid only if the same shared formula $\hat{\alpha}$ is valid on every prompt world in that instance; holdout-valid is defined analogously over that instance's holdout worlds. For overall model comparisons, we also estimate 95% percentile bootstrap confidence intervals over benchmark instances (2,000 resamples, stratified by scenario), using the same instance-level metrics and survivor-conditioned definitions as in the main tables. When the only parse failure is a suffix of missing right parentheses, we conservatively auto-close the formula at end of string, evaluate the repaired form, and report such cases separately from unrepaired parse errors. Syntax failures are separated from semantic invalidity in our reporting: across the deduplicated evaluation cache, 1.3% of outputs required conservative suffix auto-closing and 3.4% remained unrepaired parse failures, whereas 21.7% were well-formed but semantically invalid on the prompt worlds; among the strongest models, unrepaired syntax failures are near-zero, with GPT-5.4 standing out mainly in repair rate (6.7% auto-closed outputs).

6.2 Main Results

Table 2 summarizes prompt-set performance across models along three axes: (i) *validity* (does the repaired theory have any model?), (ii) *syntactic compactness* (how large are the prompt-valid formulas?), and (iii) *cost parsimony* (how close is the exception count to a solver lower bound?). The table reports repaired prompt validity (PV%), strict non-repaired prompt validity (PSV%), AST, Gap (vs. the relaxed free-Ab lower bound; see Section 4.1), and GRef (vs. the planted generator reference). Two patterns stand out.

Frontier models already separate into distinct performance profiles. Opus-4.6, Gemini-3.1, DSR, and Grok4.1f combine high prompt validity with relatively compact valid formulas, while GPT-5.4 pushes Gap and GRef much lower only by accepting substantially lower validity and much larger formulas. This contrast matters because formula size is not merely cosmetic: the largest-AST systems generalize substantially worse to holdout worlds, suggesting that some models reduce prompt cost through brittle case-splitting rather than portable repair rules. Bootstrap uncertainty does not materially change these conclusions. GPT-5.4 remains a clear low-gap/low-validity outlier under 95% instance-level bootstrap intervals, and Opus-4.6 remains the lowest-gap model within the $> 90\%$-validity cluster; by contrast, fine-grained ranking among DSR, Grok4.1f, Opus-4.6, and Gemini-3.1 on holdout validity or survivor-conditioned ΔGap should be treated as near-tied.

Cost Gap and Formula Size Separate Different Models. Among valid predictions, semantic cost and syntactic

Table 2: **Abduction Prompt-Set Performance.** Models are grouped by overall repaired prompt validity into high ($> 90\%$), intermediate, and low ($< 50\%$) bands; within each band, rows are ordered by overall Gap. Percentages are benchmark instances, not worlds. PV% = prompt-valid after conservative suffix repair; PSV% = strict prompt-validity without repair; AST = mean formula size among prompt-valid formulas; Gap = normalized extra exceptions above the solver lower bound; GRef = normalized extra exceptions vs. the planted generator reference. Gap and GRef average over prompt-valid formulas only. Lower is better for AST, Gap, and GRef. Best values bold.

	FULL					PARTIAL					SKEPTICAL					Overall				
Model	PV%	PSV%	AST	Gap	GRef	PV%	PSV%	AST	Gap	GRef	PV%	PSV%	AST	Gap	GRef	PV%	PSV%	AST	Gap	GRef
High Validity ($> 90\%$)																				
Opus-4.6	**100**	**100**	12.4	1.26	0.74	**98**	**98**	15.9	1.02	0.55	99	99	15.8	0.86	0.17	**99**	**99**	14.7	1.05	0.51
GPT-5.2	85	84	36.6	0.93	0.41	92	92	16.3	1.30	0.84	**100**	**100**	15.7	1.49	0.80	92	92	22.0	1.25	0.70
Gemini-3.1	99	99	12.5	1.27	0.76	97	97	11.4	1.40	0.93	98	98	13.5	1.18	0.50	98	98	12.3	1.30	0.76
Grok4.1f	96	94	12.8	1.32	0.80	95	95	9.0	1.69	1.23	96	96	11.8	1.46	0.82	95	95	11.0	1.51	0.98
DSR	96	94	11.9	1.42	0.90	96	96	9.4	1.54	1.08	96	93	11.7	1.69	1.01	96	95	10.8	1.55	1.00
Intermediate Validity																				
GPT-5.4	76	68	106.1	**0.13**	**-0.38**	86	81	61.5	**0.65**	**0.18**	94	92	32.6	**0.37**	**-0.28**	85	80	65.9	**0.42**	**-0.12**
Grok4	87	87	26.2	0.90	0.39	92	92	10.5	1.55	1.09	89	89	16.8	1.08	0.40	89	89	17.1	1.21	0.67
Gemini-3	77	77	18.1	1.23	0.72	65	65	16.5	1.22	0.79	93	93	15.3	1.25	0.59	77	77	16.6	1.23	0.70
Kimi-K2t	60	57	13.3	1.45	0.93	73	72	10.3	1.68	1.21	83	80	10.3	2.41	1.74	72	70	11.1	1.84	1.30
Low Validity ($< 50\%$)																				
Hermes4	3	3	**8.2**	2.85	2.31	1	1	**8.0**	1.78	1.44	4	4	6.5	3.11	2.56	2	2	7.4	2.80	2.29
GPT-4o	9	9	8.4	3.05	2.59	23	23	8.4	3.18	2.68	29	29	**4.7**	2.87	2.32	20	20	**6.9**	3.04	2.52

Table 3: **Abduction Prompt-Set Performance by Theory.** PV% = prompt-validity after conservative suffix repair; GRef = normalized extra exceptions vs. the planted generator reference, averaged over prompt-valid formulas only. T1–T5 aggregate across all scenarios; T6–T7 are ABD-Skeptical only. Lower is better for GRef. Best values bold.

	Opus-4.6		GPT-5.2		Gemini-3.1		Grok4.1f		DSR		GPT-5.4		Grok4		Gemini-3		Kimi-K2t		Hermes4		GPT-4o	
Theory	PV%	GRef	PV%	GRef	PV%	GRef	PV%	GRef	PV%	GRef	PV%	GRef	PV%	GRef	PV%	GRef	PV%	GRef	PV%	GRef	PV%	GRef
T1	**100**	0.81	97	1.10	97	1.03	97	1.46	95	1.46	95	**0.07**	92	0.95	97	0.99	76	1.97	3	4.00	16	5.29
T2	**99**	0.74	94	0.84	**99**	0.93	98	1.05	96	1.10	82	**0.05**	91	0.76	81	0.71	75	1.33	2	1.01	16	2.00
T3	**99**	0.71	88	0.67	98	0.71	97	1.02	97	0.97	82	**-0.14**	90	0.74	70	0.63	74	1.36	1	2.00	28	1.25
T4	**99**	0.46	88	0.78	97	0.89	91	1.12	95	1.13	86	**-0.18**	86	0.71	60	0.86	59	1.44	0	—	14	4.66
T5	98	0.16	96	0.61	98	0.56	92	0.80	**99**	0.86	85	**-0.21**	90	0.62	92	0.53	82	0.99	8	2.82	5	5.17
T6	**100**	-0.88	**100**	-0.88	**100**	-0.81	**100**	-0.61	93	-0.65	93	**-0.99**	86	-0.71	93	-0.79	64	-0.62	0	—	93	1.88
T7	96	0.03	**100**	0.24	**100**	0.24	96	0.64	92	0.60	96	**-0.05**	92	0.19	96	1.07	88	1.16	4	1.00	60	1.67

compactness are only partially aligned. Among the strongest high-validity models, normalized gaps cluster around roughly 1.0–1.6 additional abnormalities beyond the lower bound, while mean AST among valid formulas stays in the low teens (10.8–14.7 for Opus-4.6, Gemini-3.1, DSR, and Grok4.1f). GPT-5.4 is the main exception: it attains the best overall prompt Gap (0.42) and GRef (-0.12), but this cost advantage comes with lower validity (85.2%) and by far the largest valid formulas (AST 65.9 overall). A low cost gap is therefore not by itself a sufficient measure of parsimony. Interpreting the normalized scale: with $\approx$ 6–10 prompt worlds per instance (Table 1), a gap of ≈ 1.1 corresponds to roughly 7–11 extra abnormal elements per instance.

Several models "beat the reference" (negative reference-relative gap) on a nontrivial fraction of instances. Planted generator references are *generator anchors*, not ground truth; beating the reference often correlates with larger AST size. GPT-5.4 makes this especially clear: it beats the reference on 56.8% of valid ABD-Full/ABD-Partial instances, but those reference-beating formulas have mean AST 90.1, versus 52.1 for GPT-5.2, 22.4 for Opus-4.6, and 19.5 for Gemini-3.1. This signal is diagnostic, not evidence of "winning" the task: cost parsimony relative to the solver lower bound remains the primary semantic metric, while AST tracks compactness.

6.3 Scenario and Theory Breakdown

Table 2 stratifies results by observation regime (FULL, PARTIAL, SKEPTICAL), while Table 3 breaks down performance by background theory. Note that T1–T5 appear in all three scenarios, so their rows aggregate across regimes; T6–T7 are ABD-Skeptical only. They were added only to enrich the skeptical regime: T6 stresses existential witness existence when R facts may be unknown, while T7 stresses worst-case universal counterexamples created by unknown R facts.

Partial Observation Trades Feasibility for Stability. ABD-Partial often increases feasibility (higher validity) for some models by allowing existential completions of unknown atoms, but it does not uniformly improve parsimony. In fact, models can remain valid while drifting farther from the lower

Table 4: **Generalization Drop Profile.** Aggregated across all scenarios. PV%/HV% are instance-level prompt/holdout validity; PGap/HGap are normalized per-world gaps on their respective valid subsets; ΔGap is computed on survivors valid on both prompt and holdout; AST = mean formula size among prompt-valid formulas. Best values bold.

Model	PV%	HV%	PGap	HGap	ΔGap	AST
Opus-4.6	**98.8%**	62.2%	1.05	2.24	+0.97	14.7
GPT-5.2	92.2%	61.5%	1.25	2.45	+0.85	22.0
Gemini-3.1	98.0%	69.7%	1.30	2.37	+0.98	12.3
Grok4.1f	95.2%	**82.2%**	1.51	2.54	+0.94	11.0
DSR	95.9%	80.6%	1.55	2.64	+0.96	10.8
GPT-5.4	85.2%	24.8%	**0.42**	**1.37**	**+0.56**	65.9
Grok4	89.4%	59.1%	1.21	2.32	+0.95	17.1
Gemini-3	76.9%	49.6%	1.23	2.13	+0.80	16.6
Kimi-K2t	71.5%	64.8%	1.84	2.85	+0.93	11.1
Hermes4	2.3%	3.0%	2.80	4.71	+1.17	7.4
GPT-4o	19.8%	19.0%	3.04	4.01	+0.99	6.9

bound, reflecting that "having more ways to make the world consistent" does not automatically translate into *finding* parsimonious repairs.

Skeptical Completion Changes the Failure Mode. ABD-Skeptical requires hypotheses to remain valid under *all* completions of unknown atoms, not just some favorable one. Prompt validity under skeptical semantics is not uniformly lower—some models even achieve higher prompt validity in Skeptical than in Full (Table 2)—but the cost metric differs: models must minimize the worst-case exception count, which penalizes formulas that "get lucky" under particular completions. The main consequence of robust semantics emerges on holdout (Section 6): the primary failure mode shifts from parsimony inflation (in ABD-Full/ABD-Partial) to *validity brittleness*, where rules that satisfy universal-completion on prompt worlds often break on holdouts.

Model Families Respond Differently to Missingness. A notable qualitative contrast in Table 2 is that some models benefit from ABD-Partial (higher validity), while others lose validity under partial observation. This suggests that the partial setting genuinely probes reasoning about *what could be true* rather than merely making the task easier. Gemini-3.1 improves sharply over Gemini-3 in every regime, especially on validity. DSR and Grok4.1f combine high validity with low-teens valid-formula AST and later emerge as the strongest holdout-generalization pair. Kimi-K2t shows the opposite tradeoff from GPT-5.4: once it finds a prompt-valid rule, it is unusually robust on holdout, but its gaps remain large. GPT-5.4 drives cost down with much larger repair policies, which rarely survive holdout unchanged.

Theory Choice Changes Which Shortcuts Exist. Across theories (Table 3), T5 is consistently easier, while T3/T4 exhibit larger spread across models. A plausible explanation is that theories with stronger relational structure (e.g., nested quantification or additional dependencies in consequents) constrain the exception rule more tightly, reducing the effectiveness of compact heuristics. This theory-conditioned variation is valuable: it yields a natural set of controlled "difficulty knobs" without changing the output language or evaluation protocol. GPT-5.4 also attains the best GRef in every theory row without leading validity on any theory family, reinforcing that low prompt cost and robust validity are distinct capabilities. Additionally, T6 exhibits negative GRef values for most frontier models (Table 3), so T6 GRef should be read as a generator-reference diagnostic rather than absolute optimality.

6.4 Holdout Generalization

Prompt worlds are the adversarially hardened worlds shown in the instance; holdout worlds are fresh matched draws from the same sampler and theory parameters, subject to the same per-world acceptance checks but *without* competitor elimination (Section 4.4). Holdout is therefore an out-of-filter distribution refresh, not an i.i.d. split. Because prompt worlds are harder by construction, holdout validity can exceed prompt validity for some models; the primary signal is the change in parsimony among rules that survive on both sets.

Table 4 shows the overall generalization pattern: across models, holdout gaps are substantially larger than prompt gaps, with ΔGap typically around $+1$ extra exception per world among survivor instances.

Two Distinct Failure Modes. Table 4 aggregates across all three scenarios and mixes two effects: (i) whether a rule stays satisfiable on holdouts (*validity brittleness*), and (ii) how parsimonious it remains when satisfiable (*parsimony inflation*). Conditional holdout validity further disentangles these by reporting holdout validity conditioned on prompt validity. Because ΔGap is computed only on the survivor intersection, a small ΔGap does not by itself imply strong generalization. GPT-5.4 is the clearest example: it has the best overall ΔGap (+0.56) but only 28.5% conditional holdout validity. These two failure modes manifest differently across scenarios, as we show next.

Scenario-Level Holdout Analysis. Table 5 breaks down holdout generalization by scenario, revealing a striking contrast. In ABD-Full and ABD-Partial, the main failure mode is *parsimony inflation*: Grok4.1f and DSR achieve the strongest holdout validity, but even they pay roughly one extra exception per world on fresh samples. In ABD-Skeptical, the pattern differs: ΔGap is smaller, but holdout validity drops are more severe—many rules that satisfy universal-completion on prompt worlds fail outright on holdouts. Kimi-K2t reaches the highest skeptical holdout validity (71%) at substantially worse cost, while GPT-5.4 again shows that low ΔGap without survival is not enough (33% holdout validity). This suggests that robust semantics regularizes cost inflation but makes validity harder to preserve.

6.5 Complexity and Generalization

Table 6 relates formula size (AST bins: $<$15, 15–30, $\geq$30) to holdout degradation across all three tasks. Very small formulas tend to underfit (high ΔGap), while very large formulas reintroduce brittleness via case-splitting. Moderate-sized formulas generally degrade least. On paired problems where multiple models produced valid formulas, shorter-or-reference-length formulas have much higher conditional holdout validity (85% overall vs. 28% for longer formulas), while longer formulas achieve smaller survivor-conditioned ΔGap

Table 5: **Holdout Generalization by Scenario.** PV%/HV% are prompt/holdout validity; PGap/HGap are normalized per-world gaps on their respective valid subsets; ΔGap is computed on survivors valid on both prompt and holdout. Best HV% and ΔGap per scenario bold.

Model	FULL					PARTIAL					SKEPTICAL				
	PV%	HV%	PGap	HGap	ΔGap	PV%	HV%	PGap	HGap	ΔGap	PV%	HV%	PGap	HGap	ΔGap
Opus-4.6	100%	87%	1.26	2.66	+1.34	98%	51%	1.02	2.09	+0.92	99%	49%	0.86	1.59	+0.25
GPT-5.2	85%	54%	0.93	2.74	+1.49	92%	68%	1.30	2.52	+1.02	100%	61%	1.49	2.06	**-0.02**
Gemini-3.1	99%	87%	1.27	2.60	+1.35	97%	68%	1.40	2.44	+1.00	98%	52%	1.18	1.80	+0.21
Grok4.1f	96%	**88%**	1.32	2.59	+1.27	95%	**85%**	1.69	2.85	+1.14	96%	69%	1.46	1.83	-0.02
DSR	96%	85%	1.42	2.86	+1.42	96%	**85%**	1.54	2.72	+1.11	96%	70%	1.69	2.23	+0.09
GPT-5.4	76%	18%	0.13	1.07	**+0.68**	86%	25%	0.65	2.13	+0.96	94%	33%	0.37	0.73	+0.04
Grok4	87%	61%	0.90	1.86	+0.98	92%	72%	1.55	2.80	+1.12	89%	38%	1.08	1.85	+0.43
Gemini-3	77%	60%	1.23	2.42	+1.21	65%	40%	1.22	2.21	+0.98	93%	52%	1.25	1.66	+0.11
Kimi-K2t	60%	58%	1.45	2.89	+1.45	73%	66%	1.68	2.87	+1.16	83%	**71%**	2.41	2.76	+0.10
Hermes4	3%	5%	2.85	5.51	+1.40	1%	1%	1.78	3.00	+1.22	4%	4%	3.11	4.07	+0.96
GPT-4o	9%	17%	3.05	5.58	+2.03	23%	16%	3.18	3.31	**+0.86**	29%	25%	2.87	3.38	+0.83

(+0.52 vs. +0.95 overall). Lower cost can therefore be purchased by brittle case-splitting.

6.6 Discussion

Models differ qualitatively in their performance profiles. Opus-4.6, Gemini-3.1, DSR, and Grok4.1f form a high-validity cluster; within that group, DSR and Grok4.1f form the strongest compact holdout-generalization pair, with average valid-formula AST near 11, while Opus-4.6 and Gemini-3.1 lead in prompt validity. GPT-5.4 is the clearest low-gap outlier: it achieves the best prompt Gap and GRef, but does so with the largest valid formulas (AST 65.9 overall) and weak holdout survival (24.8% overall; 28.5% conditional on prompt validity). Kimi-K2t shows nearly the opposite trade-off, remaining compact and conditionally robust but with substantially larger gaps. Taken together, the results support three conclusions.

(i) ABD Is Not Solved. Among the high-validity systems, cost parsimony remains challenging: prompt-set gap stays around one extra abnormality per world beyond the solver lower bound, and this gap roughly doubles on holdouts in ABD-Full/Partial.

(ii) Holdout Evaluation Across All Three Regimes Is Essential. Prompt metrics alone are misleading because prompt sets are adversarially filtered. Holdouts reveal different patterns by scenario: parsimony inflation dominates in ABD-Full/Partial, while validity brittleness dominates in ABD-Skeptical, suggesting that robust semantics regularizes cost but makes validity harder to preserve. They also show that low survivor-conditioned ΔGap is not sufficient: GPT-5.4 has the smallest overall ΔGap, yet only 28.5% conditional holdout validity.

(iii) Cost- and Size-Aware Reporting Is Necessary. Binary validity saturates; cost gaps reveal one axis of semantic parsimony; AST reveals whether that cost is achieved with compact formulas or with bloated case-splitting. The strongest counterexample is GPT-5.4, whose best-in-paper gap scores come with AST 65.9 and weak holdout survival. More broadly, longer-than-reference formulas attain lower ΔGap than shorter ones (+0.52 vs. +0.95) while collapsing holdout validity (28% vs. 85%). Improved abductive reasoning therefore requires improvement in all three metrics: validity, cost gap, and formula size.

Table 6: **Holdout Generalization by Formula Complexity.** H|P columns report holdout validity conditional on prompt validity (percent); Δ columns report survivor-conditioned ΔGap. Hermes4 and GPT-4o are omitted due to too few valid responses.

	FULL						PARTIAL						SKEPTICAL					
	[0,15)		[15,30)		[30,∞)		[0,15)		[15,30)		[30,∞)		[0,15)		[15,30)		[30,∞)	
Model	H|P	Δ	H|P	Δ	H|P	Δ	H|P	Δ	H|P	Δ	H|P	Δ	H|P	Δ	H|P	Δ	H|P	Δ
Opus-4.6	98	+1.5	62	+0.6	50	**0.0**	80	+1.1	30	+0.5	0	—	73	+0.3	26	+0.1	10	+2.2
GPT-5.2	97	+1.7	65	+0.9	17	+0.8	89	+1.1	42	+0.7	12	+0.9	88	**-0.1**	24	+0.3	12	+1.7
Gemini-3.1	93	+1.6	71	+0.8	**100**	+0.1	78	**+1.0**	38	+0.7	—	—	73	+0.2	18	0.0	**33**	+0.5
Grok4.1f	99	+1.4	64	+0.7	89	+0.1	93	+1.2	33	**-0.1**	0	—	87	0.0	**31**	+0.2	—	—
DSR	96	+1.6	57	+0.7	75	+0.1	89	+1.1	**70**	+1.2	—	—	86	+0.1	26	0.0	0	—
GPT-5.4	**100**	+1.9	**77**	**+0.4**	15	+0.6	84	**+1.0**	32	+1.0	9	+0.8	67	**-0.1**	26	**-0.1**	16	+0.7
Grok4	89	+1.5	69	+0.7	50	+0.8	90	+1.1	17	+1.2	0	—	67	+0.5	26	+0.2	7	**-1.0**
Gemini-3	93	**+1.3**	61	+1.1	50	+0.7	82	+1.1	30	+0.4	**27**	**+0.7**	77	+0.2	23	**-0.1**	9	-0.9
Kimi-K2t	**100**	+1.5	73	+0.5	70	+1.8	**95**	+1.2	35	+0.9	0	—	**97**	+0.1	6	+0.5	0	—

7 Conclusion

We introduced ABD, a solver-checkable benchmark for multi-world default–exception abduction over finite first-order worlds, with exact SMT-based evaluation under full, partial, and skeptical observation regimes. This lets us assess not only whether a model can synthesize a valid repair rule across prompt worlds, but also how that rule trades off parsimony, syntactic complexity, and generalization.

Across tasks, frontier models often produce *valid* repairs on prompt worlds, but ABD remains unsaturated because *validity*, *cost parsimony*, and *syntactic complexity* do not saturate together. Among the high-validity models, the normalized gap to the solver lower bound remains roughly 1.0–1.6 extra abnormalities per world, and the more compact systems stay in the low teens of valid-formula AST, while GPT-5.4 reaches much lower prompt gaps only by using much larger formulas.

Holdout evaluation shows that generalization remains the key challenge, and that its failure mode depends on the completion semantics. In ABD-Full and ABD-Partial, the main failure is *parsimony inflation*: even when a rule remains valid on holdout worlds, its cost gap usually increases by about one additional exception per world, indicating overfitting to the filtered prompt worlds.

In contrast, ABD-Skeptical induces a different generalization profile. For rules that remain skeptically valid, gap inflation is often smaller than in ABD-Full/ABD-Partial, consistent with the worst-case objective acting as an implicit regularizer on exception marking. However, the main failure becomes *validity brittleness*: rules that satisfy the universal-completion criterion on prompt worlds frequently fail outright on holdouts. Taken together, these outcomes suggest that robust completion semantics shifts difficulty from "how few exceptions can I mark?" to "can I preserve a universally valid repair policy under distribution refresh?"

Finally, complexity matters: longer-than-reference formulas have only 28% holdout validity versus 85% for shorter formulas, even though their ΔGap is lower. ABD should therefore be read as a joint profile over validity, parsimony, and syntactic budget.

Limitations. Domains are small to permit exact SMT verification, which also makes extensional case-splitting feasible; we therefore report size-conditioned and holdout metrics to diagnose memorization. Our free-Ab baseline relaxes the single-formula constraint, so gap-to-opt is conservative, and planted generator references are anchors rather than uniquely optimal targets.

Software and Data. Code and data are available at https://github.com/SerafimBatzoglou/concept-synth, including benchmark files, prompts, cached outputs, evaluation scripts, and instructions to reproduce the reported tables and figures.

Acknowledgements

This work used Anthropic Claude Code and OpenAI Codex for software development and editorial support. All scientific decisions, experiments, verification, and final writing were performed and validated by the author.

A Task Prompts

Each benchmark instance is presented to the model as a single user message preceded by a fixed system prompt. The system prompt is shared across all three observation regimes (ABD-Full, ABD-Partial, ABD-Skeptical). The user message consists of:

1. A **task description** explaining the observation semantics, S-expression grammar, and validity/parsimony objectives;
2. The **predicate scope** (allowed and forbidden predicates for the hypothesis);
3. The **theory** (axioms in S-expression form);
4. The **prompt worlds** (domain, predicate interpretations, and—for partial/skeptical—unknown atoms);
5. An **output format** specification requesting JSON with the formula and a natural-language description.

Below we reproduce the system prompt followed by one concrete example prompt for each scenario, generated directly from the benchmark's prompt-building code.

A.1 System Prompt (All Scenarios)

The following system prompt is prepended to every API call, regardless of scenario.

```
You are an expert in first-order logic and abductive reasoning.
Your task is to find concise formulas that explain abnormal behavior in logical systems.
Always output valid JSON with the required fields.
```

A.2 ABD-Full Example Prompt

The following prompt was generated from a benchmark instance with theory `TH7` (T5) and 6 prompt worlds under the **closed-world** observation regime. All predicate truth values are fully observed; unlisted atoms are false.

```
# First-Order Logic Abduction (Full Observation)

## Task Overview

You are given a **default reasoning theory** and several finite "worlds" with **full observation**:
- A set of theory axioms involving an abnormality predicate Ab(x)
- Multiple worlds, each with a finite domain and complete predicate interpretations
- Your task: find a formula alpha(x) that defines which objects are "abnormal"

Each world contains:
- A finite domain of objects (named a0, a1, a2, ...)
- Complete interpretations of predicates P(x), Q(x), R(x,y), S(x,y)

**Closed World Assumption**: Only the facts explicitly listed as TRUE are true. Any predicate application not listed
     should be assumed FALSE.

## Abnormality Semantics

The abnormality predicate Ab(x) is defined by your formula:
- **Ab(x) <-> alpha(x)**: An object a is abnormal if and only if alpha(a) evaluates to TRUE

The theory axioms use Ab(x) to express default rules with exceptions. For example:
- "P(x) AND NOTAb(x) -> Q(x)" means: normally, P-objects have Q, unless they are abnormal

**Your goal**: Find a formula alpha(x) such that:
1. **Validity**: When Ab(x) is defined as alpha(x), all theory axioms are satisfied in every prompt world
2. **Parsimony**: Among valid formulas, prefer those that minimize the number of abnormal objects

## Scoring / Tie-breaks (important)

- alpha(x) is a SINGLE global formula shared across all worlds.
- Primary objective: **Validity** (must satisfy all axioms in every prompt world).
- Secondary objective: **Parsimony** -- minimize the total number of abnormal objects across prompt worlds:
  cost = SUM_world |{ a in Domain_world : alpha(a) is TRUE }|
- Tertiary objective: **Simplicity** -- if multiple formulas have the same validity + cost, prefer smaller formulas (
     lower AST / fewer operators / fewer quantifiers).

## Output Format
```

You must output your formula in S-expression syntax. The grammar is:

```
alpha ::= (P x)              -- unary predicate applied to variable
    | (R x y)            -- binary predicate applied to two variables
    | (= x y)            -- equality of two variables
    | (not alpha)            -- negation
    | (and alpha_1 alpha_2 ...)    -- conjunction (2 or more arguments)
    | (or alpha_1 alpha_2 ...)     -- disjunction (2 or more arguments)
    | (forall v alpha)       -- universal quantification
    | (exists v alpha)       -- existential quantification
```

Important constraints:
- Your formula must have exactly one free variable: x
- All other variables must be bound by quantifiers (forall or exists)
- Variable names should be: x (free), y, z, w (bound by quantifiers)
- **Do NOT mention Ab in your formula.**
- **Use ONLY the predicate symbols listed in AllowedAlphaPredicates for this instance.**
- **If your formula uses any predicate in ForbiddenAlphaPredicates, it is invalid.**
- **Do NOT use object names like a0, a1, ... inside alpha(x). Use variables only.**
- Prefer simpler formulas; aim for minimal abnormal set
- **No implication or biconditional**: Express these using other connectives:
  - "A implies B" should be written as `(or (not A) B)`
  - "A if and only if B" should be written as `(and (or (not A) B) (or (not B) A))`

Examples of Valid Formulas

Note: Examples below are syntactic illustrations; only use predicates that appear in AllowedAlphaPredicates for THIS instance.

- `(P x)` -- "x is abnormal if x satisfies P"
- `(and (P x) (not (Q x)))` -- "x is abnormal if x has P but not Q"
- `(exists y (and (R x y) (not (P y))))` -- "x is abnormal if x has an R-successor without P"
- `(not (exists y (R x y)))` -- "x is abnormal if x has no R-successors"
- `(forall y (or (not (R x y)) (Q y)))` -- "x is abnormal if all R-successors of x have Q"

Evaluation

Your formula will be evaluated on:
1. **Validity**: Does defining Ab(x) <-> alpha(x) satisfy all theory axioms in every world?
2. **Parsimony**: How many objects are marked as abnormal? (fewer is better)
3. **Generalization**: Does the formula also work on holdout worlds?

Note: A trivially true formula like `(or (P x) (not (P x)))` makes everything abnormal (valid but not parsimonious). A trivially false formula makes nothing abnormal (may violate axioms).

Problem Instance

AllowedAlphaPredicates: ["P", "R"]
ForbiddenAlphaPredicates: ["Ab", "Q", "S"]

Theory ID: TH7

Axioms:
1. `(forall x (implies (and (exists y (and (R x y) (P y))) (not (Ab x))) (exists z (and (S x z) (Q z)))))`

Prompt Worlds

World W0
Domain: {a0, a1, a2, a3, a4, a5, a6, a7, a8, a9, a10}

Predicates (Closed World Assumption: unlisted atoms are false):
- P: {a7, a3, a0, a9}
- Q: {a0, a6, a8}
- R: {(a10, a9), (a8, a2), (a5, a8), (a0, a2), (a7, a9), (a3, a1), (a4, a4), (a0, a8), (a4, a2), (a2, a1), (a0, a10), (a3, a4), (a9, a9), (a1, a1), (a6, a4), (a4, a0), (a4, a1), (a6, a1), (a10, a0), (a9, a7), (a8, a10), (a6, a10), (a5, a1), (a1, a6), (a1, a8), (a6, a5), (a8, a1)}
- S: {(a4, a7), (a10, a7), (a3, a1), (a4, a3), (a0, a0), (a5, a0), (a4, a6), (a8, a3), (a10, a8), (a4, a5), (a2, a3), (a6, a2), (a9, a10), (a8, a9), (a2, a9), (a5, a5), (a7, a9), (a3, a7), (a5, a6)}

World W1
Domain: {a0, a1, a2, a3, a4, a5, a6, a7, a8, a9, a10}

Predicates (Closed World Assumption: unlisted atoms are false):
- P: {a6, a1}
- Q: {a5, a10, a0, a2}
- R: {(a10, a7), (a4, a1), (a4, a9), (a9, a5), (a10, a8), (a6, a3), (a7, a7), (a10, a9), (a8, a6), (a4, a0), (a6, a9), (a2, a8), (a1, a1), (a2, a0), (a3, a10), (a7, a3), (a0, a5), (a3, a7), (a10, a5), (a0, a0)}
- S: {(a2, a3), (a10, a7), (a7, a1), (a4, a0), (a4, a7), (a5, a9), (a9, a5), (a2, a5), (a7, a9), (a7, a3), (a1, a1), (a6, a4), (a6, a2), (a6, a1), (a2, a1), (a1, a6), (a8, a4), (a7, a0), (a7, a4), (a8, a10)}

World W2
Domain: {a0, a1, a2, a3, a4, a5, a6, a7, a8, a9}

Predicates (Closed World Assumption: unlisted atoms are false):
- P: {a5, a9}
- Q: {a4, a6, a7, a3}
- R: {(a6, a5), (a2, a7), (a4, a8), (a1, a4), (a0, a5), (a5, a5), (a7, a2), (a4, a3), (a3, a4), (a7, a3), (a6, a3), (a5, a7), (a8, a6), (a7, a5), (a9, a1), (a1, a0), (a6, a7), (a8, a1), (a3, a2), (a4, a1), (a0, a0)}
- S: {(a1, a3), (a2, a1), (a7, a9), (a6, a2), (a0, a5), (a2, a4), (a4, a7), (a5, a2), (a6, a7), (a2, a8), (a1, a8), (a0, a8), (a7, a4), (a8, a1), (a7, a7), (a5, a0), (a3, a8)}

World W3
Domain: {a0, a1, a2, a3, a4, a5, a6, a7, a8, a9}

Predicates (Closed World Assumption: unlisted atoms are false):
- P: {a7, a1, a0}
- Q: {a5, a8}
- R: {(a1, a6), (a2, a6), (a8, a3), (a9, a4), (a8, a8), (a9, a6), (a6, a4), (a0, a0), (a3, a2), (a6, a2), (a3, a0), (a5, a1), (a3, a9), (a7, a5), (a7, a8), (a8, a9), (a0, a5), (a2, a3)}
- S: {(a6, a0), (a6, a3), (a4, a8), (a8, a1), (a6, a4), (a9, a8), (a9, a4), (a3, a7), (a0, a3), (a6, a1), (a0, a1), (a1, a5), (a5, a8), (a9, a3), (a3, a8)}

World W4
Domain: {a0, a1, a2, a3, a4, a5, a6, a7, a8, a9}

Predicates (Closed World Assumption: unlisted atoms are false):
- P: {a1, a4, a5, a7}
- Q: {a7, a2, a9}
- R: {(a5, a8), (a1, a6), (a4, a5), (a8, a3), (a9, a9), (a8, a8), (a0, a0), (a5, a6), (a3, a4), (a2, a9), (a6, a9), (a7, a6), (a1, a0)}
- S: {(a7, a6), (a0, a0), (a9, a2), (a1, a8), (a1, a3), (a2, a1), (a2, a3), (a5, a8), (a3, a1), (a1, a4), (a4, a7)}

World W5
Domain: {a0, a1, a2, a3, a4, a5, a6, a7, a8, a9}

Predicates (Closed World Assumption: unlisted atoms are false):
- P: {a8, a3, a7}
- Q: {a3, a6, a4}
- R: {(a8, a8), (a1, a0), (a9, a4), (a5, a8), (a9, a3), (a3, a5), (a0, a4), (a3, a6), (a8, a3), (a5, a1), (a3, a7), (a2, a4), (a3, a2), (a4, a0), (a5, a3), (a1, a3), (a7, a8), (a6, a4), (a9, a0), (a6, a5), (a1, a5)}
- S: {(a8, a8), (a3, a3), (a4, a9), (a7, a8), (a6, a3), (a6, a5), (a6, a6), (a8, a4), (a9, a4), (a9, a2), (a8, a3), (a1, a4), (a0, a5)}

```
## Your Task

Analyze the theory axioms and prompt worlds carefully. Find a formula alpha(x) that defines abnormality such that all
     axioms are satisfied.

**Solve method (reason internally; do NOT output your reasoning):**
1. Understand the theory axioms: What do they require? When might they be violated?
2. For each world, identify which objects would violate an axiom if they were NOT abnormal
3. Look for a pattern: What property characterizes objects that MUST be abnormal?
4. Check for parsimony: Is there a simpler formula that still satisfies all axioms?
5. Formulate your hypothesis as an S-expression formula
6. **Verify**: For each world, check that defining Ab(x) <-> alpha(x) makes all axioms true

**Reminder**: Validity comes first. Only after you have a valid alpha(x), optimize parsimony (fewer abnormal objects),
      then simplify the formula.

**Example reasoning** (assumes predicates shown are allowed for alpha in this instance):
- Theory: "P(x) AND NOTAb(x) -> Q(x)" (normally, P-objects have Q)
- World: P(a0)=T, Q(a0)=F, P(a1)=T, Q(a1)=T
- Analysis: a0 has P but not Q, so if a0 is not abnormal, the axiom fails
- Result: alpha(x) = (and (P x) (not (Q x))) makes a0 abnormal, satisfying the axiom

**Important**:
- Your formula must make the axioms true in ALL prompt worlds
- Minimize the number of abnormal objects when possible
- Do NOT mention Ab in your formula
- Use ONLY predicates from AllowedAlphaPredicates; using ForbiddenAlphaPredicates is invalid
- Do NOT use object names like a0, a1, ... inside alpha(x); use variables only

---

## Output

Your output must be exactly ONE LINE containing ONLY valid JSON with exactly these keys:
- "formula": a single S-expression formula string with one free variable x
- "description": a short plain-English description (one sentence)

Do NOT include code fences, prefixes, or any other text.

Example of correct output:
{"formula":"(and (P x) (not (Q x)))","description":"Objects with P but not Q are abnormal."}
```

A.3 ABD-Partial Example Prompt

The following prompt was generated from a benchmark instance with theory `TH11` (T6) and 6 prompt worlds under the **existential-completion** (partial observation) regime. Known true facts and unknown atoms are listed explicitly; unlisted atoms that are not unknown are known false. The model must find a formula valid under *at least one* completion of unknowns per world.

```
# First-Order Logic Abduction (Partial Observation)

## Task Overview

You are given a **default reasoning theory** and several finite "worlds" with **partial observation**:
- A set of theory axioms involving an abnormality predicate Ab(x)
- Multiple worlds, each with a finite domain and partially observed predicate interpretations
- Some predicate facts are **unknown** (not observed)
- Your task: find a formula alpha(x) that defines which objects are "abnormal"

Each world contains:
- A finite domain of objects (named a0, a1, a2, ...)
- Known facts: predicates whose truth values have been observed
- Unknown atoms: predicates whose truth values are hidden
```

Observation Rules:
- Atoms listed under "Known Facts" with TRUE values are TRUE
- Atoms listed under "Unknown Atoms" have unknown truth values
- Any atom NOT listed as TRUE or Unknown is FALSE

Abnormality Semantics

The abnormality predicate Ab(x) is defined by your formula:
- **Ab(x) <-> alpha(x)**: An object a is abnormal if and only if alpha(a) evaluates to TRUE

The theory axioms use Ab(x) to express default rules with exceptions.

Your goal: Find a formula alpha(x) such that:
1. **Validity**: For each world, there EXISTS a completion of unknown atoms such that all theory axioms are satisfied when Ab(x) <-> alpha(x)
2. **Parsimony**: Among valid formulas, prefer those that minimize the number of abnormal objects in the best-case completion

Scoring / tie-breaks (important)

- alpha(x) is a SINGLE global formula shared across all worlds.
- **Completions are per-world**: each world may use a different completion of its unknown atoms.
- Primary objective: **Validity** (for every prompt world, there exists at least one completion that makes all axioms true).
- Secondary objective: **Parsimony** -- for each world, we consider the best-case completion (most favorable) and count abnormal objects under that completion.
  Total cost is the sum of these per-world best-case abnormal counts.
- Tertiary objective: **Simplicity** -- if multiple formulas have the same validity + cost, prefer smaller formulas (lower AST / fewer operators / fewer quantifiers).

Output Format

You must output your formula in S-expression syntax. The grammar is:

```
alpha ::= (P x)              -- unary predicate applied to variable
    | (R x y)            -- binary predicate applied to two variables
    | (= x y)            -- equality of two variables
    | (not alpha)            -- negation
    | (and alpha_1 alpha_2 ...)    -- conjunction (2 or more arguments)
    | (or alpha_1 alpha_2 ...)     -- disjunction (2 or more arguments)
    | (forall v alpha)       -- universal quantification
    | (exists v alpha)       -- existential quantification
```

Important constraints:
- Your formula must have exactly one free variable: x
- All other variables must be bound by quantifiers (forall or exists)
- Variable names should be: x (free), y, z, w (bound by quantifiers)
- **Do NOT mention Ab in your formula.**
- **Use ONLY the predicate symbols listed in AllowedAlphaPredicates for this instance.**
- **If your formula uses any predicate in ForbiddenAlphaPredicates, it is invalid.**
- **Do NOT use object names like a0, a1, ... inside alpha(x). Use variables only.**
- Prefer simpler formulas; aim for minimal abnormal set
- **No implication or biconditional**: Express these using other connectives:
  - "A implies B" should be written as `(or (not A) B)`
  - "A if and only if B" should be written as `(and (or (not A) B) (or (not B) A))`

Examples of Valid Formulas

Note: Examples below are syntactic illustrations; only use predicates that appear in AllowedAlphaPredicates for THIS instance.

- `(P x)` -- "x is abnormal if x satisfies P"
- `(and (P x) (not (Q x)))` -- "x is abnormal if x has P but not Q"

- `(exists y (and (R x y) (not (P y))))` -- "x is abnormal if x has an R-successor without P"
- `(not (exists y (R x y)))` -- "x is abnormal if x has no R-successors"
- `(forall y (or (not (R x y)) (Q y)))` -- "x is abnormal if all R-successors of x have Q"

Evaluation

Your formula will be evaluated on:
1. **Validity**: For each world, does there exist a completion of unknowns such that Ab(x) <-> alpha(x) satisfies all axioms?
2. **Parsimony**: In the best-case completion, how many objects are abnormal? (fewer is better)
3. **Generalization**: Does the formula work on holdout worlds?

Key insight: Unknown atoms can be assigned TRUE or FALSE to help satisfy the axioms. Your formula should work with the known facts while allowing unknowns to be completed favorably.

Problem Instance

AllowedAlphaPredicates: ["P", "Q", "R"]
ForbiddenAlphaPredicates: ["Ab", "S"]

Theory ID: TH11

Axioms:
1. `(forall x (implies (and (exists y (and (R x y) (P y))) (not (Ab x))) (exists z (and (S x z) (forall w (implies (R z w) (P w)))))))`

Prompt Worlds

World W0
Domain: {a0, a1, a2, a3, a4, a5, a6, a7, a8}

Known Facts (unlisted atoms that are not Unknown are known FALSE):
- P: {a3, a7, a5, a8}
- Q: {a6, a7, a0}
- R: {(a3, a8), (a0, a8), (a4, a5), (a0, a3), (a4, a4), (a4, a0), (a8, a6), (a4, a8), (a0, a2)}
- S: {(a7, a2), (a4, a0), (a8, a2), (a7, a5), (a3, a6), (a2, a3), (a1, a4), (a2, a4), (a2, a6), (a6, a5)}

Unknown Atoms (truth value not observed, can be completed either way):
- R: {(a0, a0), (a6, a0), (a8, a7), (a8, a1), (a8, a5), (a3, a4), (a4, a7), (a6, a4), (a8, a3)}
- S: {(a2, a8), (a4, a4), (a8, a5), (a5, a5), (a6, a8), (a7, a8)}

World W1
Domain: {a0, a1, a2, a3, a4, a5, a6, a7, a8}

Known Facts (unlisted atoms that are not Unknown are known FALSE):
- P: {a2, a8, a4, a7}
- Q: {a7, a4}
- R: {(a4, a0), (a8, a8), (a3, a4), (a0, a8), (a8, a7), (a5, a2), (a1, a0), (a3, a6), (a3, a1), (a8, a1)}
- S: {(a6, a4), (a5, a6), (a5, a5), (a8, a6), (a8, a4), (a0, a1)}

Unknown Atoms (truth value not observed, can be completed either way):
- R: {(a8, a0), (a5, a8), (a6, a7), (a4, a7), (a3, a0), (a4, a6), (a5, a7), (a3, a7), (a7, a8)}
- S: {(a6, a7), (a7, a0), (a6, a5), (a6, a6), (a1, a8), (a7, a4)}

World W2
Domain: {a0, a1, a2, a3, a4, a5, a6, a7, a8}

Known Facts (unlisted atoms that are not Unknown are known FALSE):
- P: {a0, a7, a4, a8}
- Q: {a6, a1}
- R: {(a1, a8), (a4, a8), (a2, a5), (a6, a4), (a4, a3), (a0, a6), (a3, a6), (a1, a1), (a6, a8)}
- S: {(a3, a3), (a1, a2), (a5, a0), (a6, a4), (a2, a6), (a5, a3), (a5, a2), (a4, a2), (a4, a0), (a4, a6), (a8, a1), (

a8, a8), (a3, a0)}

Unknown Atoms (truth value not observed, can be completed either way):
- R: {(a3, a1), (a6, a1), (a5, a4), (a8, a7), (a8, a4), (a1, a7), (a5, a0), (a0, a3), (a5, a5)}
- S: {(a1, a3), (a7, a0), (a0, a5), (a5, a4), (a2, a5), (a7, a2)}

World W3
Domain: {a0, a1, a2, a3, a4, a5, a6, a7, a8}

Known Facts (unlisted atoms that are not Unknown are known FALSE):
- P: {a3, a4, a2, a1, a0}
- Q: {a8, a3, a5, a1}
- R: {(a8, a2), (a3, a7), (a3, a2), (a7, a0), (a0, a3), (a5, a4), (a7, a6), (a1, a0), (a5, a2)}
- S: {(a5, a0), (a7, a8), (a4, a0), (a8, a8), (a1, a4), (a2, a3), (a8, a7), (a0, a0), (a2, a8), (a6, a5), (a2, a5), (a2, a0)}

Unknown Atoms (truth value not observed, can be completed either way):
- R: {(a4, a1), (a1, a1), (a2, a2), (a8, a5), (a7, a2), (a0, a6), (a0, a1), (a0, a4), (a7, a8)}
- S: {(a6, a1), (a0, a5), (a0, a2), (a6, a6), (a5, a1), (a4, a2)}

World W4
Domain: {a0, a1, a2, a3, a4, a5, a6, a7, a8}

Known Facts (unlisted atoms that are not Unknown are known FALSE):
- P: {a4, a0, a8, a3}
- Q: {a0, a8}
- R: {(a5, a0), (a0, a2), (a7, a6), (a6, a4), (a1, a3), (a5, a2), (a2, a0), (a3, a8)}
- S: {(a8, a8), (a1, a3), (a4, a6), (a2, a4), (a2, a5), (a6, a3)}

Unknown Atoms (truth value not observed, can be completed either way):
- R: {(a2, a3), (a8, a1), (a3, a3), (a8, a8), (a4, a6), (a0, a6), (a0, a7), (a1, a0), (a3, a7)}
- S: {(a4, a0), (a7, a7), (a2, a7), (a0, a4), (a4, a7), (a6, a0)}

World W5
Domain: {a0, a1, a2, a3, a4, a5, a6, a7, a8}

Known Facts (unlisted atoms that are not Unknown are known FALSE):
- P: {a2, a1}
- Q: {a2, a7}
- R: {(a0, a4), (a5, a7), (a3, a6), (a0, a6), (a0, a5), (a8, a2), (a2, a5), (a1, a4), (a0, a1), (a2, a8)}
- S: {(a8, a3), (a5, a0), (a5, a5), (a5, a2), (a4, a5), (a5, a8), (a3, a7), (a6, a7)}

Unknown Atoms (truth value not observed, can be completed either way):
- R: {(a6, a7), (a1, a1), (a7, a3), (a5, a0), (a7, a7), (a7, a6), (a5, a1), (a3, a5), (a4, a2)}
- S: {(a6, a3), (a2, a0), (a8, a2), (a5, a4), (a3, a5), (a2, a1)}

Your Task

Analyze the theory axioms and prompt worlds carefully, keeping in mind that some facts are unknown.

Solve method (reason internally; do NOT output your reasoning):
1. Understand the theory axioms: What do they require? When might they be violated?
2. For each world, note which predicate facts are known vs. unknown
3. Identify which objects would violate an axiom based on KNOWN facts alone
4. Consider how unknown atoms might be completed to satisfy axioms
5. Look for a pattern: Which objects are **forced** to be abnormal by the KNOWN facts (i.e., no completion can make them normal while keeping axioms true)?
6. Formulate your hypothesis as an S-expression formula
7. **Verify**: For each world, check that there exists some completion of unknowns such that Ab(x) <-> alpha(x) satisfies all axioms

Example reasoning (assumes predicates shown are allowed for alpha in this instance):
- Theory: "P(x) AND NOTAb(x) -> Q(x)" (normally, P-objects have Q)

```
- World:
  - Known: P(a0)=T
  - Known: P(a1)=T
  - Unknown: Q(a1)
- Analysis:
  - a0 is forced abnormal: since Q(a0)=F is KNOWN because it is not listed to be
  true or unknown. No completion can fix Q(a0) to be true; if a0 were normal, the default would fail.
  - a1 does NOT need to be abnormal: we can complete Q(a1)=T, making the default true while keeping a1 normal.
- Candidate: alpha(x) = (and (P x) (not (Q x))) marks exactly the "P but not Q" objects abnormal.
  Under a favorable completion, Q(a1)=T so a1 is not abnormal, but a0 remains abnormal because Q(a0)=F is fixed.

**Key insight**: Focus on patterns in the KNOWN facts: those listed to be true, or
those not listed to be true or unknown, and therefore known to be false.
Unknown atoms provide flexibility and can be completed to help satisfy axioms and reduce unnecessary abnormality --
     but they cannot override known TRUE/FALSE facts.

**Important**:
- Your formula must allow axioms to be satisfied in ALL prompt worlds (for some completion)
- Minimize the number of abnormal objects when possible
- Do NOT mention Ab in your formula
- Use ONLY predicates from AllowedAlphaPredicates; using ForbiddenAlphaPredicates is invalid
- Do NOT use object names like a0, a1, ... inside alpha(x); use variables only

---

## Output

Your output must be exactly ONE LINE containing ONLY valid JSON with exactly these keys:
- "formula": a single S-expression formula string with one free variable x
- "description": a short plain-English description (one sentence)

Do NOT include code fences, prefixes, or any other text.

Example of correct output:
{"formula":"(and (P x) (not (Q x)))","description":"Objects with P but not Q are abnormal."}
```

A.4 ABD-Skeptical Example Prompt

The following prompt was generated from a benchmark instance with theory `TH11` (T6) and 5 prompt worlds under the **universal-completion** (skeptical) regime. The world format is identical to ABD-Partial but the validity criterion is stricter: the formula must satisfy the axioms under *all* completions of unknowns, and parsimony is measured by the worst-case abnormal count.

```
# First-Order Logic Abduction (Skeptical/Universal Completion)

## Task Overview

You are given a **default reasoning theory** and several finite "worlds" with **partial observation**:
- A set of theory axioms involving an abnormality predicate Ab(x)
- Multiple worlds, each with a finite domain and partially observed predicate interpretations
- Some predicate facts are **unknown** (not observed)
- Your task: find a formula alpha(x) that defines which objects are "abnormal"

Each world contains:
- A finite domain of objects (named a0, a1, a2, ...)
- Known facts: predicates whose truth values have been observed
- Unknown atoms: predicates whose truth values are hidden

**Observation Rules**:
- Atoms listed under "Known Facts" with TRUE values are TRUE
- Atoms listed under "Unknown Atoms" have unknown truth values
- Any atom NOT listed as TRUE or Unknown is FALSE

## Abnormality Semantics
```

The abnormality predicate Ab(x) is defined by your formula:
- **Ab(x) <-> alpha(x)**: An object a is abnormal if and only if alpha(a) evaluates to TRUE

The theory axioms use Ab(x) to express default rules with exceptions.

SKEPTICAL Validity (IMPORTANT DIFFERENCE)

Unlike partial observation with existential completion, this task uses **skeptical (universal) completion**:

Your goal: Find a formula alpha(x) such that:
1. **Skeptical Validity**: For each world, **FOR ALL completions** of unknown atoms, the theory axioms are satisfied when Ab(x) <-> alpha(x)
2. **Parsimony**: Among valid formulas, prefer those that minimize the **WORST-CASE** number of abnormal objects

Your formula must be **robust** -- it must ensure the axioms hold regardless of how the unknown atoms are resolved.

Scoring / tie-breaks (important)

- alpha(x) is a SINGLE global formula shared across all worlds.
- **Completions are per-world**: each world considers ALL possible completions of its unknown atoms.
- Primary objective: **Skeptical Validity** (for every prompt world, ALL completions must satisfy all axioms).
- Secondary objective: **Parsimony** -- for each world, we consider the worst-case completion (least favorable) and count abnormal objects under that completion.
  Total cost is the sum of these per-world worst-case abnormal counts.
- Tertiary objective: **Simplicity** -- if multiple formulas have the same validity + cost, prefer smaller formulas (lower AST / fewer operators / fewer quantifiers).

Output Format

You must output your formula in S-expression syntax. The grammar is:

```
alpha ::= (P x)              -- unary predicate applied to variable
    | (R x y)            -- binary predicate applied to two variables
    | (= x y)            -- equality of two variables
    | (not alpha)            -- negation
    | (and alpha_1 alpha_2 ...)    -- conjunction (2 or more arguments)
    | (or alpha_1 alpha_2 ...)     -- disjunction (2 or more arguments)
    | (forall v alpha)       -- universal quantification
    | (exists v alpha)       -- existential quantification
```

Important constraints:
- Your formula must have exactly one free variable: x
- All other variables must be bound by quantifiers (forall or exists)
- Variable names should be: x (free), y, z, w (bound by quantifiers)
- **Do NOT mention Ab in your formula.**
- **Use ONLY the predicate symbols listed in AllowedAlphaPredicates for this instance.**
- **If your formula uses any predicate in ForbiddenAlphaPredicates, it is invalid.**
- **Do NOT use object names like a0, a1, ... inside alpha(x). Use variables only.**
- Prefer simpler formulas; aim for minimal worst-case abnormal set
- **No implication or biconditional**: Express these using other connectives:
  - "A implies B" should be written as `(or (not A) B)`
  - "A if and only if B" should be written as `(and (or (not A) B) (or (not B) A))`

Examples of Valid Formulas

Note: Examples below are syntactic illustrations; only use predicates that appear in AllowedAlphaPredicates for THIS instance.

- `(P x)` -- "x is abnormal if x satisfies P"
- `(and (P x) (not (Q x)))` -- "x is abnormal if x has P but not Q"
- `(exists y (and (R x y) (not (P y))))` -- "x is abnormal if x has an R-successor without P"
- `(not (exists y (R x y)))` -- "x is abnormal if x has no R-successors"

- `(forall y (or (not (R x y)) (Q y)))` -- "x is abnormal if all R-successors of x have Q"
- `(exists y (exists z (and (R x y) (R x z) (not (= y z)))))` - "x has at least two distinct R-successors"

Evaluation

Your formula will be evaluated on:
1. **Skeptical Validity**: For each world, do ALL completions of unknowns lead to Ab(x) <-> alpha(x) satisfying all axioms?
2. **Parsimony**: In the worst-case completion, how many objects are abnormal? (fewer is better)
3. **Generalization**: Does the formula work on holdout worlds?

Key insight: Your formula must be robust enough that no matter how the unknown atoms are resolved, the axioms still hold. This is more demanding than existential completion -- you cannot rely on a "helpful" assignment of unknowns.

Problem Instance

AllowedAlphaPredicates: ["P", "Q", "R"]
ForbiddenAlphaPredicates: ["Ab", "S"]

Theory ID: TH11
Description: existsy(R(x,y)ANDP(y)) AND NOTAb(x) -> existsz(S(x,z)ANDforallw(R(z,w)->P(w))): Nested universal in consequent

Axioms:
1. `(forall x (implies (and (exists y (and (R x y) (P y))) (not (Ab x))) (exists z (and (S x z) (forall w (implies (R z w) (P w)))))))`

Prompt Worlds

World W0
Domain: {a0, a1, a2, a3, a4, a5, a6, a7, a8, a9}

Known Facts (unlisted atoms that are not Unknown are known FALSE):
- P: {a3, a8}
- Q: {a5, a6, a7, a8, a9}
- R: {(a0, a0), (a0, a5), (a0, a6), (a1, a4), (a1, a7), (a2, a3), (a2, a4), (a2, a5), (a2, a8), (a3, a1), (a3, a3), (a3, a4), (a5, a4), (a5, a6), (a5, a8), (a6, a4), (a7, a3), (a7, a4), (a7, a5), (a7, a6), (a7, a9), (a8, a5), (a8, a6), (a8, a8), (a8, a9), (a9, a1), (a9, a4), (a9, a9)}
- S: {(a0, a4), (a1, a1), (a1, a2), (a1, a5), (a3, a8), (a4, a6), (a4, a7), (a5, a1), (a5, a4), (a5, a7), (a5, a8), (a6, a4), (a6, a5), (a7, a3), (a7, a4), (a7, a6), (a8, a0), (a8, a1), (a8, a8), (a9, a2), (a9, a3), (a9, a4), (a9, a6), (a9, a7), (a9, a8)}

Unknown Atoms (truth value not observed, can be completed either way):
- R: {(a8, a3), (a2, a1), (a6, a3), (a3, a8), (a6, a9)}
- S: {(a6, a8), (a7, a9), (a6, a2), (a6, a9), (a5, a6)}

World W1
Domain: {a0, a1, a2, a3, a4, a5, a6, a7, a8, a9}

Known Facts (unlisted atoms that are not Unknown are known FALSE):
- P: {a4, a7}
- Q: {a3, a4}
- R: {(a0, a2), (a0, a6), (a1, a6), (a2, a5), (a2, a6), (a3, a1), (a3, a6), (a3, a8), (a3, a9), (a4, a0), (a4, a3), (a4, a4), (a4, a5), (a4, a6), (a4, a8), (a4, a9), (a5, a3), (a5, a5), (a5, a6), (a7, a1), (a7, a5), (a7, a6), (a8, a3), (a8, a5), (a8, a6), (a9, a3), (a9, a4), (a9, a5), (a9, a6), (a9, a7)}
- S: {(a0, a0), (a0, a1), (a0, a2), (a0, a5), (a0, a6), (a0, a7), (a0, a8), (a0, a9), (a1, a1), (a1, a2), (a1, a3), (a1, a4), (a1, a5), (a1, a6), (a1, a7), (a1, a8), (a1, a9), (a2, a0), (a2, a1), (a2, a2), (a2, a3), (a2, a6), (a4, a9), (a9, a6)}

Unknown Atoms (truth value not observed, can be completed either way):
- R: {(a4, a1), (a1, a9), (a8, a8)}

- S: {(a7, a0), (a5, a1), (a9, a4), (a4, a3), (a5, a7), (a7, a1), (a2, a4)}

World W2
Domain: {a0, a1, a2, a3, a4, a5, a6, a7, a8, a9}

Known Facts (unlisted atoms that are not Unknown are known FALSE):
- P: {a4, a7}
- Q: {a3, a4}
- R: {(a0, a2), (a0, a6), (a1, a6), (a2, a5), (a2, a6), (a3, a1), (a3, a6), (a3, a8), (a3, a9), (a4, a0), (a4, a3), (a4, a5), (a4, a6), (a4, a8), (a4, a9), (a5, a3), (a5, a5), (a5, a6), (a7, a1), (a7, a5), (a7, a6), (a8, a3), (a8, a5), (a8, a6), (a9, a3), (a9, a4), (a9, a5), (a9, a6), (a9, a7)}
- S: {(a0, a0), (a0, a1), (a0, a2), (a0, a5), (a0, a6), (a0, a7), (a0, a8), (a0, a9), (a1, a1), (a1, a2), (a1, a3), (a1, a5), (a1, a6), (a1, a7), (a1, a8), (a1, a9), (a2, a0), (a2, a1), (a2, a2), (a2, a3), (a2, a4), (a2, a6), (a4, a9), (a9, a6)}

Unknown Atoms (truth value not observed, can be completed either way):
- R: {(a4, a4), (a8, a9), (a0, a0), (a8, a2), (a3, a2)}
- S: {(a1, a4), (a5, a8), (a6, a7), (a3, a9), (a7, a7)}

World W3
Domain: {a0, a1, a2, a3, a4, a5, a6, a7, a8, a9}

Known Facts (unlisted atoms that are not Unknown are known FALSE):
- P: {a4, a7}
- Q: {a3, a4}
- R: {(a0, a2), (a0, a6), (a1, a6), (a2, a5), (a2, a6), (a3, a1), (a3, a6), (a3, a8), (a3, a9), (a4, a0), (a4, a3), (a4, a4), (a4, a5), (a4, a6), (a4, a8), (a4, a9), (a5, a3), (a5, a5), (a5, a6), (a7, a1), (a7, a5), (a8, a3), (a8, a5), (a8, a6), (a9, a3), (a9, a4), (a9, a5), (a9, a6), (a9, a7)}
- S: {(a0, a0), (a0, a1), (a0, a2), (a0, a5), (a0, a6), (a0, a7), (a0, a8), (a0, a9), (a1, a1), (a1, a3), (a1, a4), (a1, a5), (a1, a6), (a1, a7), (a1, a8), (a1, a9), (a2, a0), (a2, a1), (a2, a2), (a2, a3), (a2, a4), (a2, a6), (a4, a9), (a9, a6)}

Unknown Atoms (truth value not observed, can be completed either way):
- R: {(a5, a2), (a7, a6)}
- S: {(a1, a2), (a8, a0), (a6, a9), (a8, a4), (a3, a2), (a5, a8), (a5, a2), (a5, a4)}

World W4
Domain: {a0, a1, a2, a3, a4, a5, a6, a7, a8, a9}

Known Facts (unlisted atoms that are not Unknown are known FALSE):
- P: {a4, a7}
- Q: {a3, a4}
- R: {(a0, a2), (a0, a6), (a1, a6), (a2, a5), (a2, a6), (a3, a1), (a3, a6), (a3, a8), (a3, a9), (a4, a0), (a4, a3), (a4, a4), (a4, a5), (a4, a6), (a4, a8), (a4, a9), (a5, a3), (a5, a5), (a5, a6), (a7, a1), (a7, a5), (a7, a6), (a8, a3), (a8, a5), (a8, a6), (a9, a3), (a9, a4), (a9, a5), (a9, a6), (a9, a7)}
- S: {(a0, a0), (a0, a1), (a0, a2), (a0, a6), (a0, a7), (a0, a8), (a0, a9), (a1, a2), (a1, a3), (a1, a4), (a1, a5), (a1, a6), (a1, a7), (a1, a8), (a1, a9), (a2, a0), (a2, a1), (a2, a2), (a2, a3), (a2, a4), (a2, a6), (a4, a9), (a9, a6)}

Unknown Atoms (truth value not observed, can be completed either way):
- R: {(a2, a1), (a7, a0), (a4, a2)}
- S: {(a3, a1), (a3, a9), (a7, a4), (a9, a5), (a0, a5), (a1, a1), (a3, a6)}

Your Task

Analyze the theory axioms and prompt worlds carefully. Remember: your formula must work for ALL possible completions of unknown atoms, not just some favorable completion.

Solve method (reason internally; do NOT output your reasoning):
1. Understand the theory axioms: What do they require? When might they be violated?
2. For each world, note which predicate facts are known vs. unknown
3. Identify which objects would violate an axiom in ANY completion
4. Consider the WORST-CASE completion: which objects might be forced abnormal when unknowns are resolved unfavorably?

```
5. Look for a pattern: Which objects must be abnormal to ensure axioms hold NO MATTER HOW unknowns are completed?
6. Formulate your hypothesis as an S-expression formula
7. **Verify**: For each world, check that for ALL completions of unknowns, Ab(x) <-> alpha(x) satisfies all axioms

**Example reasoning** (assumes predicates shown are allowed for alpha in this instance):
- Theory: "P(x) AND NOTAb(x) -> Q(x)" (normally, P-objects have Q)
- World:
  - Known: P(a0)=T
  - Known: P(a1)=T
  - Unknown: Q(a1)
- Analysis for SKEPTICAL semantics:
  - a0 is forced abnormal: Q(a0)=F is KNOWN, so if a0 were normal, the default would fail.
  - a1 MUST ALSO be abnormal under skeptical semantics: Q(a1) is unknown, so in the WORST-CASE completion where Q(a1)=
    F, the default would fail if a1 were normal.
  - Unlike existential completion where we could rely on Q(a1)=T, skeptical requires robustness against Q(a1)=F.
- Candidate: alpha(x) = (P x) marks ALL P-objects abnormal, ensuring robustness.

**Key insight**: Focus on what might go WRONG if unknowns are completed adversarially:
- An object with unknown predicates in the antecedent/consequent of a default might be forced abnormal to handle the
    worst case.
- You cannot assume a "helpful" completion of unknowns -- you must be prepared for the least favorable assignment.

**Important**:
- Your formula must make axioms satisfied in ALL prompt worlds for ALL completions
- Minimize the worst-case number of abnormal objects when possible
- Do NOT mention Ab in your formula
- Use ONLY predicates from AllowedAlphaPredicates; using ForbiddenAlphaPredicates is invalid
- Do NOT use object names like a0, a1, ... inside alpha(x); use variables only

---

## Output

Your output must be exactly ONE LINE containing ONLY valid JSON with exactly these keys:
- "formula": a single S-expression formula string with one free variable x
- "description": a short plain-English description (one sentence)

Do NOT include code fences, prefixes, or any other text.

Example of correct output:
{"formula":"(P x)","description":"All P-objects are abnormal to ensure robustness against any completion."}
```

B Output Language Specification

B.1 S-Expression Grammar

Models must output hypotheses in the following S-expression syntax (EBNF). The grammar is shared across all three observation regimes.

```
alpha  ::= atom
         | "(not" alpha ")"
         | "(and" alpha alpha+ ")"
         | "(or"  alpha alpha+ ")"
         | "(forall" var alpha ")"
         | "(exists" var alpha ")"

atom   ::= "(" pred var ")"              (* unary *)
         | "(" pred var var ")"           (* binary *)
         | "(=" var var ")"               (* equality *)

pred   ::= "P" | "Q" | "R" | "S"

var    ::= "x" | "y" | "z" | "w"
```

Implication and biconditional are not primitive; models are instructed to encode them:
- $A \to B$ as `(or (not A) B)`
- $A \leftrightarrow B$ as `(and (or (not A) B) (or (not B) A))`

Variable discipline. The hypothesis $\alpha(x)$ must contain exactly one free variable x; all other variables must be bound by quantifiers, and no object constants (`a0`, `a1`, ...) may appear in α.

B.2 Predicate Scoping Rules

Each benchmark instance specifies two predicate sets:

`AllowedAlphaPredicates`: The predicates that $\alpha(x)$ may reference. These always form a subset of $\{P, Q, R, S\}$ and vary by theory (see Appendix C).

`ForbiddenAlphaPredicates`: Predicates that must *not* appear in $\alpha(x)$. This set always includes Ab (since α *defines* Ab) and also includes the *repaired predicates*—those appearing in the consequent of the default rule—to prevent trivial shortcut solutions such as $\alpha(x) = \neg Q(x)$.

Additional constraints. The hypothesis $\alpha(x)$ must have exactly one free variable x. All other variables must be bound by quantifiers. Object names (`a0`, `a1`, ...) may not appear inside α; only logical variables are permitted.

B.3 Complexity Measures

We use two standard measures of formula complexity throughout the paper.

AST Size ($|\alpha|$). The number of nodes in the abstract syntax tree of α:
- Each predicate application counts as $1+$ the number of its arguments (each argument variable counts as 1).
- Equality counts as $1 + 2 = 3$.
- Negation: $1 + |\varphi|$.
- Binary connectives ($\wedge$, $\vee$): $1 + |\varphi_1| + |\varphi_2|$.
- Quantifiers ($\forall v$, $\exists v$): 1 (quantifier) + 1 (bound variable) + $|\varphi|$.

For example, `(exists y (and (R x y) (P y)))` has AST size $= 2 + (1 + (3 + 2)) = 8$: `exists`(1) + `y`(1) + `and`(1) + `R`(1) + `x`(1) + `y`(1) + `P`(1) + `y`(1).

Quantifier Depth ($\mathrm{QD}(\alpha)$). The maximum nesting depth of quantifiers, ignoring connectives:
- Atomic formulas and equality: 0.
- Negation, conjunction, disjunction: inherited from children ($\max$ for binary connectives).
- $\forall v.\, \varphi$ or $\exists v.\, \varphi$: $1 + \mathrm{QD}(\varphi)$.

For example, `(P x)` has $\mathrm{QD} = 0$; `(exists y (R x y))` has $\mathrm{QD} = 1$; `(forall y (exists z (R y z)))` has $\mathrm{QD} = 2$.

C Theory Library

The benchmark uses seven default-exception theories of the form $\forall x\big(\phi(x) \wedge \neg \mathsf{Ab}(x) \to \psi(x)\big)$, where ϕ is the antecedent (trigger condition) and ψ is the consequent (default conclusion). Theories T1–T5 are used in all three scenarios; T6–T7 are used only in ABD-Skeptical.

C.1 Theory Formulas

Table C.1 lists each theory with its full S-expression axiom.

Table C.1: Theory library. Each theory is a single default rule $\phi(x) \wedge \neg \mathsf{Ab}(x) \to \psi(x)$.

ID	Antecedent $\phi(x)$	Consequent $\psi(x)$
T1	$\exists y\big(R(x,y) \wedge P(y)\big)$	$Q(x)$
T2	$\exists y\big(R(x,y) \wedge P(y)\big)$	$\exists z\big(S(x,z) \wedge Q(z)\big)$
T3	$\exists y\big(S(x,y) \wedge P(y)\big)$	$\exists z\big(R(x,z) \wedge Q(z)\big)$
T4	$\exists y\big(R(x,y) \wedge P(y)\big)$	$\exists z\big(S(x,z) \wedge \forall w(R(z,w) \to P(w))\big)$
T5	$\exists y\big(R(x,y) \wedge P(y)\big)$	$\forall z\big(S(x,z) \to Q(z)\big)$
T6	$P(x)$	$\exists y\big(R(x,y)\big)$
T7	$P(x)$	$\forall y\big(R(x,y) \to Q(y)\big)$

C.2 Allowed and Forbidden Predicates

Table C.2 shows, for each theory, which predicates may appear in the hypothesis $\alpha(x)$ and which are forbidden. Forbidden predicates include Ab (always) plus the *repaired predicates*—those appearing in the consequent—to block trivial shortcut solutions.

Table C.2: Predicate scoping per theory. *Repaired* = consequent predicates excluded from α to prevent shortcuts.

ID	Allowed in α	Forbidden in α	Repaired
T1	P, R, S	Ab, Q	Q
T2	P, R	Ab, S, Q	S, Q
T3	P, S	Ab, R, Q	R, Q
T4	P, Q, R	Ab, S	S
T5	P, R, S	Ab, Q	Q
T6	P, Q, S	Ab, R	R
T7	P, R, S	Ab, Q	Q

C.3 Theory Commentary

T1 (relational antecedent, unary consequent). The simplest relational theory; checking the default reduces to verifying $Q(x)$ for each domain element triggered by an R-neighbor with P. Models frequently achieve near-optimal cost.

T2 (relational antecedent, existential consequent). The consequent requires *finding* an S-witness with Q, making validity checking harder and parsimony gaps wider. T2 drives the largest ΔGap values in holdout evaluation, suggesting models learn prompt-specific S-patterns that fail to transfer.

T3 (swapped relations: S-antecedent, R-consequent). Mirrors T2 with S and R exchanged. Since R is repaired (forbidden in α), the hypothesis space is restricted to $\{P, S\}$, which limits the available structural patterns and often forces models toward simple unary heuristics.

T4 (nested universal in consequent). The consequent demands an S-witness z all of whose R-successors satisfy P—a second-order-flavored constraint. This makes validity checking expensive and produces the highest syntactic complexity in planted reference rules.

T5 (universal consequent over S). Requires that *all* S-successors of the non-abnormal object have Q. A single counterexample (an S-successor without Q) is enough to trigger abnormality, giving models a relatively readable signal.

T6 (unary antecedent, existential consequent; Skeptical-only). The simplest consequent structure: $P(x)$ just requires some R-successor. Model predictions frequently beat the planted generator reference (negative GRef), suggesting that the reference formulas are not optimal for this constraint.

T7 (unary antecedent, universal consequent; Skeptical-only). Under skeptical semantics the universal consequent $\forall y(R(x, y) \rightarrow Q(y))$ must hold for *every* completion of unknown R and Q atoms, producing tighter abnormality requirements and smaller ΔGap than the existential theories.

D Dataset Generation Details

This section describes the generation pipeline as pseudocode and parameter tables. All parameters correspond to the released benchmark (`abd_b1_v1`).

D.1 World Sampler

Each world is a finite first-order structure over four predicates (P, Q unary; R, S binary) plus the derived Ab.

Domain size. Domain sizes vary by scenario. In **ABD-Full** and **ABD-Partial**, each world's domain is drawn uniformly from $\{9, 10, 11\}$; different worlds within the same instance may have different sizes (and in practice nearly all instances span the full 9–11 range). In **ABD-Skeptical**, domains range over $\{10, 11, 12\}$; all worlds within a single instance share the same domain size. Objects are named $a_0, a_1, \ldots$

Predicate densities. For each world the sampler draws a density ρ per predicate uniformly from the ranges in Table D.1, then marks $\max(1, \lfloor n^k \cdot \rho \rfloor)$ ground atoms true (chosen uniformly at random), where n is the domain size and k is the arity.

Table D.1: Predicate density ranges by scenario. For each world the sampler draws ρ uniformly from the listed interval; $n = |D|$. ABD-Full and ABD-Partial share the same ranges; ABD-Skeptical uses denser binary predicates and a narrower P range.

Predicate	Arity	Full / Partial	Skeptical	True atoms
P	1	$[0.20,\ 0.60]$	$[0.40,\ 0.60]$	$\lfloor n \cdot \rho_P \rfloor$
Q	1	$[0.20,\ 0.60]$	$[0.20,\ 0.50]$	$\lfloor n \cdot \rho_Q \rfloor$
R	2	$[0.12,\ 0.25]$	$[0.15,\ 0.30]$	$\lfloor n^2 \cdot \rho_R \rfloor$
S	2	$[0.08,\ 0.18]$	$[0.10,\ 0.25]$	$\lfloor n^2 \cdot \rho_S \rfloor$

Unknown atoms (ABD-Partial / Skeptical). After sampling a complete world, a fraction of R and S atoms are masked as *unknown*. Table D.2 lists the per-theory masking rates used for each scenario. Unary predicates P, Q are never masked.

Table D.2: Unknown-atom rates (fraction of binary atoms masked). ABD-Partial uses the default column; ABD-Skeptical uses theory-specific rates to keep universal validity tractable.

	ABD-Partial		ABD-Skeptical	
Theory	R	S	R	S
T1 (TH2)	0.20	0.10	0.05	0.08
T2 (TH7)	0.20	0.10	0.05	0.05
T3 (TH10)	0.20	0.10	0.05	0.05
T4 (TH11)	0.20	0.10	0.05	0.05
T5 (TH12)	0.20	0.10	0.05	0.05
T6 (TH3)	—	—	0.04	0.08
T7 (TH5)	—	—	0.05	0.08

D.2 Filtering Criteria

Every candidate instance must pass the following filters before inclusion.

Validity floor. The planted reference hypothesis α^* must be valid on all prompt worlds (all scenarios) and achieve $\mathrm{opt_cost}(\alpha^*) \geq 1$ per world (the problem is non-trivial—abnormality is necessary).

Gap constraint. $\mathrm{cost}(\alpha^*) \leq \mathrm{OptCost} + 1$ per world, ensuring the planted reference is close to optimal.

Exception-rate cap. Per-world abnormal fraction ≤ 0.20:

$$\frac{\mathrm{OptCost}(W)}{|D_W|} \ \leq\ 0.20.$$

This keeps "abnormality" a minority phenomenon as intended by default-exception logic.

Holdout worlds. For each instance $k{=}5$ holdout worlds are sampled from the same distribution and parameters, but *without* adversarial competitor elimination, to test generalization under distributional shift.

D.3 Competitor Pool and CEGIS-like Elimination

Instances are hardened by adversarial world addition. The procedure ensures that no simple shortcut formula can achieve the same cost as the planted reference.

1. **Initialize pool.** Collect up to $C_{\max} = 30$ competitor formulas:
   - Hand-curated Tier-1 shortcuts (e.g. $P(x)$, $\neg Q(x)$, $\exists y\, R(x, y)$);
   - Mined Tier-2 shortcuts observed from model outputs;
   - Up to 10 syntactic mutants of the reference formula.

   Only competitors whose predicates are a subset of the allowed set are retained.

2. **Test on current worlds.** Evaluate every surviving competitor on the current prompt-world set. A competitor is *beaten* if it is invalid in ≥ 1 world or its cost exceeds the reference cost by at least a margin m (typically m=2).
3. **Add adversarial world.** If survivors remain and the world budget is not exhausted, sample a new world that invalidates ≥ 1 surviving competitor. Go to step 2.
4. **Terminate.** Accept the instance when all competitors are beaten. Reject if the world budget (typically 12–15) is exhausted with survivors remaining.

Algorithm D.1 gives the pseudocode.

```
function GENERATE_INSTANCE(theory, α*, W_budget, m):
    worlds = [sample_world(theory)]
    pool = build_pool(theory, α*, C_max)
    while |worlds| < W_budget:
        survivors = [c in pool :
            is_valid(c, worlds) and
            cost(c, worlds) < cost(α*, worlds) + m]
        if survivors == []:
            return ACCEPT(worlds)
        w = sample_adversarial(theory, survivors)
        if w is None: break          // cannot separate
        worlds.append(w)
    return REJECT("survivors remain")
```

Figure D.1: CEGIS-like adversarial world addition.

D.4 Reference Refinement and Cheater Filtering

After competitor elimination, a second filter guards against *cheater* formulas—trivial patterns that happen to match the reference cost but have no semantic depth.

Cheater pool. The cheater pool comprises two tiers:

Tier 1 ($\sim$25 formulas): Constants ($\top$, $\bot$), atomic unary/binary predicates and their negations, self-loops ($R(x,x)$, $S(x,x)$), simple quantified existence ($\exists y\, R(x,y)$), and unary conjunctions/disjunctions.

Tier 2 ($\sim$15 formulas): Common model shortcuts mined from preliminary runs, e.g. $\exists y(R(x,y) \wedge P(y))$, $P(x) \wedge \exists y(R(x,y))$.

Rejection criterion. An instance is rejected if the best cheater achieves cost $\leq \mathrm{cost}(\alpha^*) - 1$ on the current prompt worlds. This ensures that the planted reference rule is not trivially beatable by a simple heuristic.

Reference refinement (optional). When enabled, the pipeline evaluates $\sim$20 alternative reference candidates (sampled from the theory's template library plus mutants of the seed) and selects the one with the best validity-cost-cheater-margin trade-off.

E Completion Semantics and Solver Encoding

All validity checks and cost computations are implemented via reduction to SMT using the Z3 solver. This section formalises the encoding.

E.1 World Representation

A world W over domain $D_W = \{a_0, \ldots, a_{n-1}\}$ is specified by:

- $\mathrm{True}_P(W)$: set of atoms known true for each predicate P;
- $\mathrm{Unk}_P(W)$: set of atoms whose truth value is unknown;
- all remaining atoms are known false (closed-world assumption on the "rest").

A *completion* $\mathcal{C}$ assigns each unknown atom a truth value. The completed world $W^{\mathcal{C}}$ has:

$$\mathrm{Facts}(W^{\mathcal{C}}) \;=\; \mathrm{True}_P(W) \;\cup\; \{\, a \in \mathrm{Unk}_P(W) : \mathcal{C}(a) = \top \,\}.$$

Z3 encoding. Each unknown atom $u \in \mathrm{Unk}_P(W)$ is represented by a free Z3 boolean variable. Known-true atoms are replaced by `True`, known-false atoms by `False`, and unknown atoms by their variable. The abnormality predicate $\mathsf{Ab}(a)$ is replaced *inline* by the Z3 grounding of $\alpha(a)$, making the definitional substitution explicit in the solver encoding.

E.2 Validity Queries

Let Θ denote the theory axioms and $\Phi_W(\alpha, \mathcal{C})$ abbreviate the conjunction of grounded axioms after substituting $\mathsf{Ab} \mapsto \alpha$ and applying completion $\mathcal{C}$.

ABD-Full. No unknown atoms exist. Validity reduces to direct evaluation:

$$\mathrm{Valid}_{\mathrm{full}}(\alpha, W) \stackrel{\text{def}}{=} W \models \Theta[\mathsf{Ab} \mapsto \alpha].$$

ABD-Partial (existential completion).

$$\mathrm{Valid}_{\exists}(\alpha, W) \stackrel{\text{def}}{=} \exists \mathcal{C}.\ W^{\mathcal{C}} \models \Theta[\mathsf{Ab} \mapsto \alpha].$$

Encoding. Assert all grounded axioms with unknown atoms as free variables. The query is SAT; a satisfying assignment witnesses a valid completion.

ABD-Skeptical (universal completion).

$$\mathrm{Valid}_{\forall}(\alpha, W) \stackrel{\text{def}}{=} \forall \mathcal{C}.\ W^{\mathcal{C}} \models \Theta[\mathsf{Ab} \mapsto \alpha].$$

Encoding. Assert the *negation* of the grounded axiom conjunction with unknown atoms as free variables. The query checks for a *counterexample* completion:

- UNSAT $\Rightarrow$ no counterexample exists $\Rightarrow$ α is valid.
- SAT $\Rightarrow$ counterexample found $\Rightarrow$ α is invalid.

E.3 Cost Computation

The abnormality cost in world W under completion $\mathcal{C}$ is

$$\mathrm{cost}(\alpha, W^{\mathcal{C}}) \;=\; \big|\{\, a \in D_W : W^{\mathcal{C}} \models \alpha(a) \,\}\big|.$$

ABD-Full. Direct evaluation; no optimization needed.

ABD-Partial (best-case cost).

$$\mathrm{cost}_{\exists}(\alpha, W) \;=\; \min_{\mathcal{C}:\ \mathrm{Valid}_{\exists}(\alpha, W^{\mathcal{C}})} \mathrm{cost}(\alpha, W^{\mathcal{C}}).$$

Encoding. Use Z3's OPTIMIZE solver: assert the grounded axioms (with unknown atoms as free variables) and minimise $\sum_{a \in D_W} \mathbf{1}[\alpha(a)]$.

ABD-Skeptical (worst-case cost).

$$\mathrm{cost}_{\forall}(\alpha, W) \;=\; \max_{\mathcal{C}:\ \mathrm{Valid}_{\forall}(\alpha, W^{\mathcal{C}})} \mathrm{cost}(\alpha, W^{\mathcal{C}}).$$

Encoding. Use Z3's OPTIMIZE solver: assert the grounded axioms and *maximise* $\sum_{a \in D_W} \mathbf{1}[\alpha(a)]$. This finds the least favorable valid completion.

E.4 Baselines: Free-Ab Lower Bound

The conservative baseline $\mathrm{OptCost}$ relaxes the single-formula constraint:

ABD-Full.

$$\mathrm{OptCost}(W) \;=\; \min_{\mathsf{Ab}: D_W \to \{0,1\}} \big|\{a : \mathsf{Ab}(a) = 1\}\big| \quad \text{s.t.} \quad W \models \Theta.$$

Here Ab is a free function (one independent boolean per domain element), not constrained to be definable by a single formula. The solution is computed via Z3 OPTIMIZE.

ABD-Partial.

$$\mathrm{OptCost}_{\exists}(W) \;=\; \min_{\mathsf{Ab},\ \mathcal{C}} \big|\{a : \mathsf{Ab}(a) = 1\}\big| \quad \text{s.t.} \quad W^{\mathcal{C}} \models \Theta.$$

Both Ab and the completion $\mathcal{C}$ are jointly optimized.

ABD-Skeptical.

$$\mathrm{OptCost}_{\forall}(W) \;=\; \min_{\mathsf{Ab}} \; \max_{\mathcal{C}} \left|\{a : \mathsf{Ab}(a) = 1\}\right| \quad \text{s.t.} \quad \forall \mathcal{C}.\, W^{\mathcal{C}} \models \Theta.$$

Conservativeness. Because the single-formula definability constraint that one shared α must determine Ab for every object is dropped, $\mathrm{OptCost}$ is a *lower bound* on the achievable cost of any formula. The gap $\mathrm{Gap} = \mathrm{cost}(\alpha) - \mathrm{OptCost}$ therefore over-estimates actual sub-optimality but provides a consistent, formula-independent baseline across all models and instances.

F Holdout Construction and Distribution Match

F.1 Holdout Generation Algorithm

Each benchmark instance includes k=5 *holdout* worlds used exclusively for post-hoc generalization evaluation; no model ever sees holdout worlds during inference. The holdout sampler reuses the *identical* Z3-based world-generation procedure described in Appendix D.1:

1. **Extract parameters** from the prompt worlds of each instance: domain-size range, predicate-density intervals (Table D.1), unknown-atom rates (Table D.2), cost bounds, and gap constraints.
2. **Sample a world** using the same Z3 density-constrained generator. For ABD-Partial and ABD-Skeptical, unknown atoms are masked using the same per-theory rates.
3. **Filter.** Accept the world only if it (i) is not equivalent to any prompt world (identical domain size and predicate extensions), (ii) satisfies the validity semantics of the scenario, and (iii) has per-world cost and gap within the range observed across prompt worlds.
4. **Repeat** until k distinct holdout worlds are accepted (up to 150 attempts per world, 30 s per instance).

This filtering ensures that the planted reference formula α^* remains valid on every holdout world and that per-world cost and gap fall within the prompt-world regime, so holdout evaluation measures generalization under distributional match rather than distribution shift.

Deterministic seeding. Each holdout world is seeded via $\texttt{SHA-256}(dataset_path \parallel instance_id \parallel holdout_idx \parallel global_seed) \bmod 2^{31}$, ensuring full reproducibility.

F.2 What Is Not Done on Holdout

The following prompt-time hardening procedures (Appendix D.3–D.4) are *deliberately omitted* from holdout generation:

- **No competitor elimination.** Prompt worlds are iteratively added to defeat shortcut formulas (§D.3); holdout worlds are drawn independently with no adversarial loop.
- **No cheater filtering.** Prompt-set construction rejects instances where a trivial pattern matches the reference cost (§D.4); holdout applies no such filter.
- **No reference refinement.** The planted reference formula is fixed at benchmark-creation time and not re-optimized for holdout.

These omissions are intentional: holdout worlds provide a pure i.i.d. distributional test, measuring whether a model's formula captures the true pattern rather than artifacts of adversarial prompt-set construction.

F.3 Prompt vs. Holdout Distribution Match

Table F.1 compares per-world summary statistics across prompt and holdout sets. Per-world reference cost and OptCost are computed as sum/k where k is the number of worlds. The close agreement confirms that holdout worlds are drawn from the same generating distribution.

Table F.1: Per-world summary statistics for prompt and holdout sets. Values are means across all instances in each scenario. Reference Gap = Reference Cost − OptCost, per world.

	ABD-Full		ABD-Partial		ABD-Skeptical	
Metric	Prompt	Hold	Prompt	Hold	Prompt	Hold
Worlds/inst	9.0	5.0	8.3	5.0	6.0	5.0
PW Ref Cost	1.78	1.88	1.68	1.75	2.05	2.10
PW OptCost	1.27	1.33	1.20	1.32	1.36	1.54
PW Ref Gap	0.51	0.55	0.48	0.43	0.69	0.56

G Additional Results

G.1 Full Per-Theory Results

Tables G.1–G.3 report per-theory Prompt-Gap (P) and Holdout-Gap (H) for every model, with the number of *survivors* (instances valid on both prompt and holdout) in parentheses. Cells marked "—" indicate zero prompt validity. Gaps are normalized per world.

Table G.1: Per-theory results: **ABD-Full**. Each cell shows PGap / HGap (survivors).

	Opus-4.6	GPT-5.2	Gemini-3.1	Grok4.1f	DSR	GPT-5.4	Grok4	Gemini-3	Kimi-K2t	Hermes4	GPT-4o
T1	1.3/3.2 (4)	1.3/3.7 (3)	1.5/3.2 (4)	1.4/3.2 (5)	1.5/3.3 (5)	0.1/0.4 (1)	1.1/3.2 (3)	1.1/1.9 (3)	1.5/3.1 (4)	—	—
T2	1.4/3.2 (56)	1.1/3.2 (34)	1.4/3.2 (54)	1.4/2.9 (44)	1.6/3.6 (44)	0.3/1.3 (16)	0.8/1.5 (31)	1.2/2.8 (36)	1.3/3.5 (33)	—	2.0/5.3 (6)
T3	1.4/2.8 (37)	0.9/2.6 (17)	1.2/2.5 (39)	1.5/2.8 (39)	1.5/2.8 (34)	0.2/1.2 (7)	1.1/2.6 (27)	1.4/2.5 (19)	1.5/2.9 (26)	—	1.4/3.6 (4)
T4	1.3/2.3 (46)	0.9/2.4 (22)	1.4/2.1 (44)	1.3/2.3 (45)	1.5/2.6 (41)	0.0/0.4 (6)	0.9/1.4 (25)	1.4/2.2 (21)	1.7/2.6 (24)	—	5.3/6.4 (0)
T5	0.8/1.8 (27)	0.6/2.2 (17)	0.9/2.0 (27)	0.9/2.0 (23)	0.9/1.9 (27)	0.1/1.1 (4)	0.7/1.9 (21)	0.9/2.0 (24)	1.3/2.3 (24)	2.8/5.5 (5)	5.8/7.1 (1)

Table G.2: Per-theory results: **ABD-Partial**. Format as Table G.1.

	Opus-4.6	GPT-5.2	Gemini-3.1	Grok4.1f	DSR	GPT-5.4	Grok4	Gemini-3	Kimi-K2t	Hermes4	GPT-4o
T1	1.0/2.5 (5)	0.9/3.3 (2)	1.0/3.0 (3)	1.4/3.0 (5)	1.4/3.0 (5)	0.5/2.2 (2)	1.3/3.1 (4)	2.3/2.4 (3)	1.3/2.9 (4)	—	—
T2	1.2/2.6 (39)	1.4/2.8 (49)	1.5/2.8 (56)	1.6/3.0 (63)	1.6/2.9 (59)	0.7/2.6 (21)	1.6/2.9 (52)	1.3/2.3 (26)	1.7/3.0 (54)	1.8/3.0 (2)	2.9/3.2 (9)
T3	1.2/2.1 (36)	1.3/2.5 (32)	1.3/2.3 (38)	1.6/2.7 (46)	1.5/2.6 (40)	0.8/2.6 (11)	1.5/2.7 (39)	1.2/2.1 (20)	1.7/2.6 (34)	—	1.7/2.6 (24)
T4	0.9/1.7 (23)	1.4/2.6 (44)	1.5/2.3 (37)	2.0/3.1 (52)	1.7/2.9 (57)	0.6/1.8 (12)	1.7/2.9 (39)	1.3/2.2 (18)	1.9/3.2 (41)	—	5.0/5.8 (4)
T5	0.6/1.4 (21)	0.9/1.8 (25)	1.0/1.9 (31)	1.4/2.4 (26)	1.2/2.2 (30)	0.4/1.3 (12)	1.4/2.6 (26)	0.9/2.1 (19)	1.3/2.3 (25)	—	5.8/6.6 (0)

Table G.3: Per-theory results: **ABD-Skeptical**. Format as Table G.1. Note negative ΔGap for T5: holdout gap is often *lower* than prompt gap because worst-case costs can be smaller in holdout worlds for this theory.

	Opus-4.6	GPT-5.2	Gemini-3.1	Grok4.1f	DSR	GPT-5.4	Grok4	Gemini-3	Kimi-K2t	Hermes4	GPT-4o
T1	1.0/1.8 (10)	1.4/2.3 (14)	1.2/2.3 (10)	1.8/2.0 (13)	1.8/2.5 (14)	0.2/1.6 (4)	1.2/2.4 (4)	1.0/2.0 (10)	2.5/2.3 (18)	4.0/4.0 (1)	5.5/9.2 (1)
T2	1.1/2.6 (14)	1.8/3.2 (14)	1.5/2.9 (16)	1.8/2.4 (16)	1.8/3.4 (16)	0.6/0.7 (8)	1.4/2.8 (11)	1.2/2.1 (14)	3.3/4.4 (18)	1.9/5.8 (1)	2.6/4.9 (7)
T3	1.4/2.4 (13)	1.7/3.1 (12)	1.4/2.2 (10)	1.8/2.7 (14)	1.7/2.7 (14)	0.2/0.5 (6)	1.3/2.4 (10)	1.2/1.8 (11)	3.2/4.3 (16)	2.0/4.4 (1)	2.8/4.3 (5)
T4	0.9/1.7 (13)	2.1/2.7 (17)	1.5/2.0 (15)	1.5/1.9 (8)	2.3/2.9 (24)	0.6/1.0 (12)	1.2/2.6 (11)	1.7/2.6 (14)	2.8/3.3 (20)	—	—
T5	1.1/1.0 (13)	2.4/1.3 (22)	1.7/1.2 (15)	2.0/1.6 (15)	2.5/1.5 (21)	0.8/0.9 (11)	1.6/1.3 (12)	1.7/1.3 (18)	2.5/1.5 (19)	4.9/4.2 (2)	6.9/— (0)
T6	0.2/0.1 (3)	0.2/0.7 (4)	0.2/0.1 (2)	0.4/1.3 (7)	0.3/0.8 (5)	0.0/0.3 (3)	0.2/0.1 (2)	0.2/0.2 (4)	0.2/0.5 (2)	—	2.9/3.2 (12)
T7	0.1/0.4 (13)	0.3/0.7 (16)	0.3/0.7 (16)	0.7/0.9 (19)	0.7/1.0 (19)	0.0/0.1 (9)	0.3/0.5 (12)	1.2/0.6 (12)	1.3/1.4 (21)	1.0/1.8 (1)	1.7/2.1 (15)

Key observations. (i) Hermes4 and GPT-4o achieve near-zero prompt validity for most theories in ABD-Full and ABD-Partial, reflecting their difficulty with multi-world abduction. (ii) In ABD-Skeptical, holdout gaps can be *lower* than prompt gaps (T5), because worst-case cost under universal completion varies across world draws. (iii) Opus-4.6 combines the highest survivor counts with near-perfect prompt validity across all scenarios, while GPT-5.4 is the clearest cost outlier: many per-theory prompt gaps are near zero, but survivor counts are much smaller.

G.2 Gap Distribution Summaries

Table G.4 reports the distribution of normalized prompt-set gap (cost above OptCost, per world) for each model. Only prompt-valid predictions are included.

Table G.4: Normalized prompt-gap distribution by scenario. N = predictions valid on all prompt worlds. >3 and >5 are tail percentages.

Model	N	Mean	Med	P90	Max	>3	>5
ABD-Full							
Opus-4.6	195	1.26	1.25	2.00	2.7	0.0%	0.0%
GPT-5.2	152	0.93	0.90	1.99	2.4	0.0%	0.0%
Gemini-3.1	194	1.27	1.20	1.97	7.9	1.0%	1.0%
Grok4.1f	171	1.32	1.25	2.12	3.0	0.0%	0.0%
DSR	172	1.42	1.39	2.16	3.2	0.6%	0.0%
GPT-5.4	149	0.13	0.00	0.38	2.4	0.0%	0.0%
Grok4	156	0.90	0.89	1.62	2.4	0.0%	0.0%
Gemini-3	138	1.23	1.12	2.00	8.2	1.4%	0.7%
Kimi-K2t	117	1.45	1.40	2.18	2.9	0.0%	0.0%
Hermes4	5	2.84	2.20	4.40	5.8	20.0%	20.0%
GPT-4o	17	3.05	1.88	5.84	6.5	41.2%	35.3%
ABD-Partial							
Opus-4.6	238	1.02	1.00	1.67	2.9	0.0%	0.0%
GPT-5.2	208	1.30	1.22	2.11	3.0	0.0%	0.0%
Gemini-3.1	236	1.40	1.22	2.13	7.0	2.5%	0.8%
Grok4.1f	213	1.69	1.62	2.61	5.9	3.3%	0.5%
DSR	216	1.54	1.56	2.33	3.5	0.9%	0.0%
GPT-5.4	210	0.65	0.50	1.62	3.0	0.0%	0.0%
Grok4	205	1.55	1.56	2.33	5.4	2.0%	0.5%
Gemini-3	147	1.22	1.11	1.93	7.8	2.7%	0.7%
Kimi-K2t	178	1.68	1.62	2.58	4.0	3.4%	0.0%
Hermes4	2	1.78	1.78	2.22	2.3	0.0%	0.0%
GPT-4o	55	3.18	2.33	5.72	6.8	41.8%	23.6%
ABD-Skeptical							
Opus-4.6	160	0.86	0.18	2.62	6.7	9.4%	1.9%
GPT-5.2	162	1.49	0.63	4.00	8.0	20.4%	4.3%
Gemini-3.1	158	1.18	0.37	3.58	7.8	12.0%	3.8%
Grok4.1f	128	1.46	0.71	3.83	8.0	17.2%	5.5%
DSR	155	1.69	1.29	4.37	6.8	21.3%	4.5%
GPT-5.4	152	0.37	0.00	0.98	5.2	3.9%	0.7%
Grok4	144	1.08	0.33	3.12	6.7	10.4%	2.1%
Gemini-3	150	1.25	0.43	3.60	6.7	15.3%	4.7%
Kimi-K2t	134	2.41	2.08	4.64	10.0	27.6%	6.0%
Hermes4	6	3.11	2.50	5.40	6.8	33.3%	16.7%
GPT-4o	47	2.87	2.20	5.80	7.4	27.7%	14.9%

Discussion. In ABD-Full, the high-validity models still show fairly tight prompt-gap distributions, with P90 usually around 2.0–2.2 and almost no mass above gap 3. ABD-Partial is slightly heavier-tailed due to cost computation under existential completion. GPT-5.4 is the dominant low-gap outlier in every scenario (mean 0.13 in ABD-Full, 0.65 in ABD-Partial, 0.37 in ABD-Skeptical), but this concentration is a prompt-set parsimony effect rather than a sign of strong holdout robustness. Among the remaining high-validity models, ABD-Skeptical exhibits the widest tails: Opus-4.6 still has the tightest skeptical distribution (median 0.18, only 9.4% above gap 3), while GPT-5.2, DSR, and Kimi-K2t all exceed 20% above gap 3.

G.3 Holdout ΔGap by Theory

Table G.5 reports survivor-conditioned holdout ΔGap by theory. T1–T5 aggregate across scenarios; T6–T7 are ABD-Skeptical only.

G.4 Instance-Level Holdout Survival Counts

Table G.6 reports, for each model, the number of prompt-valid instances, the number valid on both prompt and holdout, and holdout validity conditional on prompt validity.

G.5 Beats-Reference Breakdown

A prediction *beats the reference* when its total prompt cost is strictly below the planted reference formula's cost on the same prompt worlds. This is possible because the reference formula is a fixed template that may not be optimal for every world draw.

Table G.5: **Holdout ΔGap by Theory.** T1–T5 aggregate across all three scenarios; T6–T7 are ABD-Skeptical only. Each cell shows ΔGap = HGap − PGap over *survivors* valid on both prompt and holdout, with N = survivor count. Cells with fewer than 5 survivors are suppressed. Lower is better. Best per theory bold.

Theory	Opus-4.6	GPT-5.2	Gemini-3.1	Grok4.1f	DSR	GPT-5.4	Grok4	Gemini-3	Kimi-K2t	Hermes4	GPT-4o
T1	+0.5 (19)	+0.4 (19)	+0.6 (17)	+0.5 (23)	+0.4 (24)	+0.6 (7)	+1.5 (11)	+0.5 (16)	**+0.2** (26)	—	—
T2	+1.5 (109)	+1.3 (97)	+1.5 (126)	+1.3 (123)	+1.5 (119)	**+0.8** (45)	+1.1 (94)	+1.2 (76)	+1.5 (105)	—	+1.7 (22)
T3	+1.0 (86)	+1.0 (61)	+1.0 (87)	+1.0 (99)	+1.1 (88)	+0.8 (24)	+1.1 (76)	**+0.7** (50)	+1.0 (76)	—	+1.0 (33)
T4	+0.8 (82)	+0.8 (83)	+0.9 (96)	+0.9 (105)	+0.9 (122)	**+0.4** (30)	+1.0 (75)	+0.9 (53)	+1.0 (85)	—	—
T5	+0.6 (61)	+0.3 (64)	+0.5 (73)	+0.5 (64)	+0.4 (78)	**+0.2** (27)	+0.6 (59)	+0.5 (61)	+0.5 (68)	+0.8 (7)	—
T6	—	—	—	+0.4 (7)	**+0.1** (5)	—	—	—	—	—	+0.2 (12)
T7	+0.3 (13)	+0.3 (16)	+0.3 (16)	**+0.2** (19)	+0.4 (19)	**+0.1** (9)	+0.3 (12)	+0.4 (12)	+0.2 (21)	—	+0.4 (15)

Table G.6: **Instance-Level Holdout Survival Counts.** All counts are benchmark instances. H|P% = holdout validity conditional on prompt validity. Best values bold.

Model	PV	P+H	H|P	H|P%
Opus-4.6	**593**	**591**	373	63.1%
GPT-5.2	522	521	344	66.0%
Gemini-3.1	588	586	417	71.2%
Grok4.1f	512	511	440	86.1%
DSR	543	542	**455**	83.9%
GPT-5.4	511	509	145	28.5%
Grok4	505	504	329	65.3%
Gemini-3	435	435	272	62.5%
Kimi-K2t	429	427	383	89.7%
Hermes4	13	13	13	**100.0%**
GPT-4o	119	119	88	73.9%

By theory. Table G.7 aggregates reference-beating rates across all models.

Table G.7: Reference-beating rate by theory (all models pooled). k/N = predictions beating the planted generator reference out of prompt-valid predictions.

Theory	ABD-Full	ABD-Partial	ABD-Skeptical
T1	4/42 (9.5%)	5/42 (11.9%)	14/239 (5.9%)
T2	63/437 (14.4%)	32/604 (5.3%)	36/186 (19.4%)
T3	62/323 (19.2%)	16/431 (3.7%)	53/185 (28.6%)
T4	77/417 (18.5%)	48/573 (8.4%)	75/219 (34.2%)
T5	63/247 (25.5%)	30/258 (11.6%)	58/208 (27.9%)
T6	—	—	86/129 (66.7%)
T7	—	—	9/230 (3.9%)

T6 in ABD-Skeptical stands out at 66.7%: the planted reference for TH3 ($P(x) \wedge \neg \mathsf{Ab}(x) \rightarrow \exists y\, R(x, y)$) yields a generous reference cost under worst-case completion, and models often find tighter formulas. Across ABD-Full and ABD-Skeptical, reference-beating is now common for several theories, especially T3–T6. ABD-Partial remains the least permissive regime, but once GPT-5.4 is included some theories still reach 8–12%, so the gap between scenarios is quantitative rather than absolute.

By reference AST size. Table G.8 groups reference-beating rate by the AST size of the planted reference formula.
In ABD-Full, the highest reference-beating rate occurs for the most complex references (AST $\geq$ 30, 34.1%), where the planted formula contains redundancy that models can trim. In ABD-Skeptical, reference-beating is strongest for mid-range and moderately large references, peaking at AST 15–20 (35.8%) and remaining high at AST 20–30 (32.6%), suggesting these formulas are complex enough to leave room for simpler alternatives under worst-case semantics. ABD-Partial remains comparatively low across AST bins, never exceeding 9.2%.

Top reference-beating examples. The largest single-instance margin is $+14$ in ABD-Skeptical (instance `abd_skeptical_v2_TH3_00`, T6): nine models independently discover a formula with prompt cost 7 vs. reference cost 21. In ABD-Full, the largest margin is now $+8$ (GPT-5.4 on `ABD_FULL_TH2_094`, T1: predicted cost 12 vs.

Table G.8: Reference-beating rate by planted generator reference AST size (all models pooled).

Ref AST	ABD-Full	ABD-Partial	ABD-Skeptical
$[0, 5)$	—	—	0/9 (0.0%)
$[5, 10)$	—	—	51/254 (20.1%)
$[10, 15)$	45/198 (22.7%)	33/510 (6.5%)	111/629 (17.6%)
$[15, 20)$	112/546 (20.5%)	60/636 (9.4%)	67/187 (35.8%)
$[20, 30)$	59/565 (10.4%)	30/595 (5.0%)	91/279 (32.6%)
$[30, +)$	53/157 (33.8%)	5/155 (3.2%)	11/38 (28.9%)

reference cost 20). ABD-Partial reaches the same maximum $+8$ (GPT-5.4 on `ABD_PARTIAL_TH7_024b`, T2: predicted cost 10 vs. reference cost 18), showing that once GPT-5.4 is included, substantial single-instance improvements over the reference are no longer confined to the other two regimes.

H Qualitative Examples

H.1 Worked Examples per Regime

For each observation regime we select one instance that has both a *robust* prediction (valid on prompt and holdout) and a *brittle* prediction (valid on prompt, fails on holdout), illustrating the generalization challenge. Formulas are shown in S-expression notation.

ABD-Full: instance `ABD_FULL_TH7_021` (T2). Theory T2: $\exists y(R(x,y) \wedge P(y)) \wedge \neg \mathsf{Ab}(x) \rightarrow \exists z(S(x,z) \wedge Q(z))$. 10 prompt worlds, 5 holdout worlds. 2 models produce robust formulas; 7 are brittle.

Robust (Opus-4.6, AST = 8):

```
(exists y (and (R x y) (P y)))
```

Prompt cost = 34 (gap 2.20/world); holdout cost = 25, all valid.

Interpretation: the formula marks any x with an R-related P-witness as abnormal. It is less cost-efficient on the prompt set than specialized alternatives, but it generalizes because it avoids accidental cardinality assumptions about the local R-neighborhood.

Brittle (Gemini-3, AST = 22):

```
(exists y (and (R x y) (P y) (forall z (or (not (and (R x z) (P z))) (= y z)))))
```

Prompt cost = 30; holdout: 3/5 worlds invalid (40% valid).

Failure: insists on a *unique* P-witness ($\forall z$-clause forces $z = y$). Three holdout worlds happen to have two or more R-related P-elements, violating uniqueness.

ABD-Partial: instance `ABD_PARTIAL_TH7_014` (T2). Same theory as above, but under partial observation. 8 prompt worlds, 5 holdout worlds. 1 robust, 8 brittle predictions.

Robust (GPT-4o, AST = 8):

```
(exists y (and (R x y) (P y)))
```

Prompt cost = 22 (gap 1.50/world); holdout cost = 23, all valid.

Interpretation: the simplest sufficient condition—x has an R-related P-element. Generalises precisely because it avoids over-specification.

Brittle (Gemini-3, AST = 12):

```
(and (exists y (and (R x y) (P y))) (not (P x)))
```

Prompt cost = 17; holdout: 0/5 worlds valid (catastrophic failure).

Failure: the conjunct $\neg P(x)$ happens to hold for all abnormal elements in the 8 prompt worlds, but is not a consequence of the theory. All 5 holdout worlds contain P-positive abnormals, breaking the formula.

ABD-Skeptical: instance `abd_skeptical_v2_TH2_11` (T1). Theory T1: $\exists y(R(x,y) \wedge P(y)) \wedge \neg \mathsf{Ab}(x) \rightarrow Q(x)$. 5 prompt worlds, 5 holdout worlds. 6 robust, 3 brittle predictions.

Robust (Opus-4.6, AST = 18):

```
(and (P x) (exists y (R x y)) (forall z (or (not (R x z)) (P z))))
```

Prompt cost = 5 (gap 0.00/world); holdout cost = 5, all valid.

Interpretation: x is abnormal if $P(x)$ holds, x has some R-successor, and all R-successors of x satisfy P. The universal quantifier over R-successors makes this robust under worst-case completion of unknown R-atoms.

Brittle (GPT-5.2, AST = 19):

```
(exists y (and (R x y) (P y) (forall z (or (not (R x z)) (= z y)))))
```

Prompt cost = 5; holdout: 1/5 worlds invalid (80% valid).

Failure: the uniqueness constraint ($\forall z$-clause forces $z = y$) insists that x has exactly one R-related P-element. Under skeptical completion, even a single holdout world with multiple P-witnesses is enough to invalidate the rule.

H.2 Common Failure Patterns

Table H.1 gives the per-model failure breakdown aggregated across all tasks.

Table H.2 categorises all model predictions by failure mode. A prediction is classified into the first matching category:

1. **Auto-repaired parse**: the only parse failure is missing trailing right parentheses, so we conservatively close the suffix and evaluate the repaired formula.
2. **Parse error**: formula still does not parse as a valid S-expression after the conservative suffix-repair attempt.
3. **All invalid (prompt)**: valid on zero prompt worlds.
4. **Partial invalid (prompt)**: valid on some but not all prompt worlds.
5. **Brittle**: valid on all prompt worlds, invalid on ≥ 1 holdout world. Sub-category *catastrophic*: $< 50\%$ holdout worlds valid.
6. **Parsimony inflation**: valid on both prompt and holdout, but ΔGap (holdout gap $-$ prompt gap, normalized per world) exceeds 2.
7. **Success**: valid on both, $\Delta\text{Gap} \leq 2$.

Table H.1: **Failure Mode Breakdown.** Aggregated across all abduction tasks. Parse = unrepaired parse failure; Repair = trailing right-parenthesis repair applied before evaluation; Invalid = parsed but not valid on all prompt worlds; Valid = prompt-valid predictions. Best values bold.

Model	Valid%	Invalid%	Parse%	Repair%	Missing%
Opus-4.6	**98.8%**	**0.5%**	**0.0%**	0.0%	0.7%
GPT-5.2	92.2%	6.0%	1.8%	0.2%	**0.0%**
Gemini-3.1	98.0%	2.0%	**0.0%**	0.0%	**0.0%**
Grok4.1f	95.2%	3.9%	0.9%	0.7%	**0.0%**
DSR	95.9%	3.9%	**0.2%**	1.6%	**0.0%**
GPT-5.4	85.2%	14.0%	0.8%	6.7%	**0.0%**
Grok4	89.4%	6.2%	4.4%	0.2%	**0.0%**
Gemini-3	76.9%	21.7%	1.4%	0.0%	**0.0%**
Kimi-K2t	71.5%	12.2%	16.3%	6.0%	**0.0%**
Hermes4	2.3%	89.8%	8.0%	0.0%	**0.0%**
GPT-4o	19.8%	77.7%	2.5%	0.0%	**0.0%**

Table H.2: Failure-mode counts across all 6,367 model predictions.

Failure Mode	ABD-Full	ABD-Partial	ABD-Skeptical
Auto-repaired parse	47	26	18
Parse error	87	43	86
All invalid (prompt)	208	343	218
Partial invalid (prompt)	288	270	54
Brittle (holdout)	357	583	571
of which catastrophic	121	174	332
Parsimony inflation	292	183	73

Observations. Auto-repaired parses are not rare: there are 93 across 6,571 predictions, compared with 217 unrepaired parse errors. Syntax failures therefore remain a minority, but they are not negligible. The dominant failure mode in both ABD-Partial and ABD-Skeptical is *brittleness* on holdout (603 and 571 cases, respectively), rather than prompt-time invalidity. Catastrophic brittleness is concentrated in ABD-Skeptical (332/571 = 58% of brittle predictions fail on at least half of holdout worlds). Parsimony inflation moves in the opposite direction, peaking in ABD-Full (305 cases) and becoming much rarer in ABD-Skeptical (73): under universal completion, models more often fail validity outright than remain valid with inflated cost.

Brittle formula patterns. Table H.3 groups brittle predictions by syntactic pattern.

Table H.3: Brittle formula syntactic patterns. "Exists-only" = existential quantifiers without universals.

Pattern	ABD-Full	ABD-Partial	ABD-Skeptical
Exists-only	126	215	225
Mixed quantifiers	79	32	47
Propositional (and)	0	9	19
Forall-only	0	3	9
Other	1	10	2

Exists-only formulas dominate across all regimes: they assert the existence of witnesses with specific properties that hold incidentally on prompt worlds (e.g. a unique P-witness, a self-loop, a specific neighborhood count). These formulas tend to be brittle because existential witnesses are sensitive to the particular world draw. The *mixed quantifiers* pattern is second most common in ABD-Full, where models attempt precise cardinality constraints ("exactly two R-successors") that break on holdout worlds with different domain structure.

I Software and Data

Code and data are available at https://github.com/SerafimBatzoglou/concept-synth/tree/main/benchmarks/abduction. The repository includes the frozen benchmark files, prompt templates, cached model outputs, evaluation scripts, table-generation scripts, and instructions for reproducing the reported tables and figures. The arXiv source additionally includes the appendix tables and prompt listings used for the extended version of the paper.

References

Alur, R.; Bodík, R.; Juniwal, G.; Martin, M. M. K.; Raghothaman, M.; Seshia, S. A.; Singh, R.; Solar-Lezama, A.; Torlak, E.; and Udupa, A. 2013. Syntax-guided synthesis. In *2013 Formal Methods in Computer-Aided Design*, 1–8. IEEE.

Arkoudas, K., and Batzoglou, S. 2025. Stress-testing the reasoning competence of language models with formal proofs. In Christodoulopoulos, C.; Chakraborty, T.; Rose, C.; and Peng, V., eds., *Findings of the Association for Computational Linguistics: EMNLP 2025*, 12376–12394. Suzhou, China: Association for Computational Linguistics.

Azerbayev, Z.; Piotrowski, B.; Schoelkopf, H.; Ayers, E. W.; Radev, D.; and Avigad, J. 2023. ProofNet: Autoformalizing and formally proving undergraduate-level mathematics. *arXiv preprint arXiv:2302.12433*.

Bansal, A.; Loos, S.; Rabe, M.; Szegedy, C.; and Wilcox, S. 2019. HOList: An environment for machine learning of higher order logic theorem proving. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, 454–463.

Baral, C. 2003. *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge, UK: Cambridge University Press.

Bhagavatula, C.; Bras, R. L.; Malaviya, C.; Sakaguchi, K.; Holtzman, A.; Rashkin, H.; Downey, D.; Yih, W.; and Choi, Y. 2020. Abductive commonsense reasoning. In *International Conference on Learning Representations (ICLR)*.

Brewka, G.; Eiter, T.; and Truszczyński, M. 2011. Answer set programming at a glance. *Communications of the ACM* 54(12):92–103.

Clark, P.; Tafjord, Ø.; and Richardson, K. 2020. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, 3882–3890.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Cropper, A., and Muggleton, S. 2020. Learning higher-order logic programs. *Machine Learning* 109:1289–1322.

Dai, W.-Z.; Xu, Q.; Yu, Y.; and Zhou, Z.-H. 2019. Bridging machine learning and logical reasoning by abductive learning. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2811–2822.

d'Avila Garcez, A., and Lamb, L. C. 2023. Neurosymbolic AI: The 3rd wave. *Artificial Intelligence Review* 56:12387–12406.

de Kleer, J., and Williams, B. C. 1987. Diagnosing multiple faults. *Artificial Intelligence* 32(1):97–130.

de Moura, L., and Bjørner, N. 2008. Z3: An efficient smt solver. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*.

Denecker, M., and Kakas, A. C. 2002. Abduction in logic programming. In Kakas, A. C., and Sadri, F., eds., *Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski, Part I*, volume 2407 of *Lecture Notes in Computer Science*. Springer. 402–436.

Dziri, N.; Lu, X.; Sclar, M.; Li, X. L.; Jiang, L.; Lin, B. Y.; Welleck, S.; West, P.; Bhagavatula, C.; Bras, R. L.; Hwang, J.; Sanyal, S.; Ren, X.; Ettinger, A.; Harchaoui, Z.; and Choi, Y. 2023. Faith and fate: Limits of transformers on compositionality. In *Advances in Neural Information Processing Systems*, volume 36, 70293–70332. Curran Associates, Inc.

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

Gandarela, J. P.; Carvalho, D. S.; and Freitas, A. 2025. Inductive learning of logical theories with LLMs: An expressivity-graded analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 23752–23759.

Gebser, M.; Kaminski, R.; Kaufmann, B.; and Schaub, T. 2012. *Answer Set Solving in Practice*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

Gelfond, M., and Lifschitz, V. 1988. The stable model semantics for logic programming. In *Proceedings of the 5th International Conference and Symposium on Logic Programming (ICLP)*, 1070–1080.

Hamscher, W.; Console, L.; and de Kleer, J. 1992. *Readings in Model-Based Diagnosis*. Morgan Kaufmann.

Han, S.; Schoelkopf, H.; Zhao, Y.; Qi, Z.; Riddell, M.; Zhou, W.; Coady, J.; Peng, D.; Qiao, Y.; Benson, L.; Sun, L.; Wardle-Solano, A.; Szabó, H.; Zubova, E.; Burtell, M.; Fan, J.; Liu, Y.; Wong, B.; Sailor, M.; Ni, A.; Nan, L.; Kasai, J.; Yu, T.; Zhang, R.; Fabbri, A.; Kryscinski, W. M.; Yavuz, S.; Liu, Y.; Lin, X. V.; Joty, S.; Zhou, Y.; Xiong, C.; Ying, R.; Cohan, A.; and Radev, D. 2024. FOLIO: Natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 22017–22031. Association for Computational Linguistics.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021. Measuring mathematical problem solving with the MATH dataset. *arXiv preprint arXiv:2103.03874*.

Hobbs, J. R.; Stickel, M. E.; Appelt, D. E.; and Martin, P. 1993. Interpretation as abduction. *Artificial Intelligence* 63(1–2):69–142.

Kakas, A. C.; Kowalski, R. A.; and Toni, F. 1992. Abductive logic programming. *Journal of Logic and Computation* 2(6):719–770.

Law, M.; Russo, A.; and Broda, K. 2014. Inductive learning of answer set programs. In *Logics in Artificial Intelligence: 14th European Conference, JELIA 2014, Funchal, Madeira, Portugal, September 24-26, 2014. Proceedings 14*, 311–325. Springer.

McCarthy, J. 1980. Circumscription—a form of non-monotonic reasoning. *Artificial Intelligence* 13(1–2):27–39.

Mostafazadeh, N.; Chambers, N.; He, X.; Parikh, D.; Batra, D.; Vanderwende, L.; Kohli, P.; and Allen, J. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 839–849. San Diego, California: Association for Computational Linguistics.

Muggleton, S., and Raedt, L. D. 1994. Inductive logic programming: Theory and methods. *Journal of Logic Programming* 19–20:629–679.

Muggleton, S. 1991. Inductive logic programming. *New Generation Computing* 8(4):295–318.

Muggleton, S. 1995. Inverse entailment and Progol. *New Generation Computing* 13(3-4):245–286.

Pei, Y.; Du, Y.; and Jin, X. 2025. FoVer: First-order logic verification for natural language reasoning. *Transactions of the Association for Computational Linguistics* 13:1340–1359.

Polu, S., and Sutskever, I. 2020. Generative language modeling for automated theorem proving. *arXiv preprint arXiv:2009.03393*.

Poole, D. 1989. Explanation and prediction: An architecture for default and abductive reasoning. *Computational Intelligence* 5(2):97–110.

Quinlan, J. R. 1990. Learning logical definitions from relations. *Machine Learning* 5(3):239–266.

Raedt, L. D. 2008. *Logical and Relational Learning*. Springer.

Reiter, R. 1980. A logic for default reasoning. *Artificial Intelligence* 13(1–2):81–132.

Reiter, R. 1987. A theory of diagnosis from first principles. *Artificial Intelligence* 32(1):57–95.

Ren, L.; Xiao, G.; Qi, G.; Geng, Y.; and Xue, H. 2025. Can LLMs solve ASP problems? insights from a benchmarking study. In *Proceedings of the 22nd International Conference on Principles of Knowledge Representation and Reasoning*, 621–631.

Roemmele, M.; Bejan, C. A.; and Gordon, A. S. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *Logical Formalizations of Commonsense Reasoning, Papers from the 2011 AAAI Spring Symposium, Technical Report SS-11-06, Stanford, California, USA, March 21–23, 2011*. AAAI.

Solar-Lezama, A.; Tancau, L.; Bodík, R.; Seshia, S.; and Saraswat, V. 2006. Combinatorial sketching for finite programs. In *Proceedings of the 12th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 404–415.

Solar-Lezama, A. 2008. *Program Synthesis by Sketching*. Ph.D. Dissertation, University of California, Berkeley.

Srinivasan, A. 2001. The ALEPH manual. Technical report, Computing Laboratory, Oxford University.

Sun, Y., and Saparov, A. 2025. Language models do not follow occam's razor: A benchmark for inductive and abductive reasoning. *CoRR* abs/2509.03345.

Tafjord, Ø.; Dalvi, B.; and Clark, P. 2021. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 3621–3634.

Torlak, E., and Bodík, R. 2014. A lightweight symbolic virtual machine for solver-aided host languages. In *Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, 530–541.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; and Zhou, D. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 24824–24837.

Yang, K., and Deng, J. 2019. Learning to prove theorems via interacting with proof assistants. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 6984–6994. PMLR.

Zheng, K.; Han, J. M.; and Polu, S. 2022. miniF2F: A cross-system benchmark for formal olympiad-level mathematics. In *International Conference on Learning Representations (ICLR)*.