Proceedings of the 23rd International Conference on Principles of Knowledge Representation and Reasoning
KR meets Machine Learning and Explanation

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

Serafim Batzoglou
Independent Researcher
serafim.batzoglou@gmail.com

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) 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. A closer companion benchmark is INDUCTION, which studies first-order concept synthesis over finite relational worlds with labeled target predicates (Batzoglou 2026).

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.

(a) Dataset Statistics

Scenario	N	Prompt	Holdout	$\|D\|$	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	$P(x) \rightarrow \forall y(R(x,y) \rightarrow Q(y))$

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.

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

	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 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.

	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

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.

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 FULL/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

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

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.

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, ≥ 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

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

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.

(+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.

	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	—

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.

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.

AI Declaration

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.

Acknowledgements

The author thanks the anonymous reviewers for their constructive feedback.

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.

Batzoglou, S. 2026. INDUCTION: Finite-structure concept synthesis in first-order logic. In *Proceedings of the 43rd International Conference on Machine Learning*. To appear. arXiv:2602.18956.

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)*.