FROM FIRST TO LAST WITHOUT CHANGING A PREDICTION: OBSERVABILITY IS NOT SPECIFICATION IN AGENT DIAGNOSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Agent diagnoses increasingly guide repairs and incident investigations, yet a complete trajectory can leave open what a correct diagnosis should identify. Should it name the final answer to correct or the upstream source to inspect? This choice can reverse a system comparison: on 38 shared cases, the leading method becomes last among four methods when the diagnostic question changes, without changing a single prediction. We make the rules that define acceptable diagnoses a controlled experimental variable: change the rules with the model fixed, then change the executor with the rules fixed. On 720 frozen trajectories, supplying explicit rules raises fully correct diagnoses from 233 to 716. The repairs concern the meaning of an identified event as well as its location and whether a location is justified at all. Targeted role swaps produce the predicted changes among cases originally answered correctly under the rules. On 217 external trajectories, explicit rules improve acceptable answers and reduce unsupported claims of uniqueness; a model and a program agree on all three main scoring outcomes for 213 cases. These tests make diagnostic-object identification operational: they check which rules govern which answers, and whether their execution is reproducible. **Before comparing diagnostic systems, specify and test what counts as a correct diagnosis.**

1 INTRODUCTION

An agent is asked when a stock first crossed a threshold in an unadjusted price series. It reads adjusted prices instead, then submits a year. The source mismatch and the final answer are both visible. One diagnostic system points to the submitted answer; another points to the source used to produce it. Which system is correct? The answer depends on what the evaluation asks: which final claim should be corrected, or which upstream evidence should be inspected. A complete record of the failure does not choose between these questions.

That choice can turn a leader into the last-ranked method. Figure 1 previews a comparison on 38 cases shared by two defensible diagnostic questions. The trajectories, predictions, and methods stay fixed; only the acceptable targets change. **The first-ranked method becomes last without making a new prediction.** The consequence is practical: a reported ranking needs the diagnostic question that gives it meaning.

Why does this matter now? Diagnosis is becoming part of how agents are repaired and investigated. LongRCA Bench contains 1,140 failed trajectories with a median of 145 steps, and scores responsible roles and root steps separately (Zhang et al., 2026). In August 2026, LangChain reported that LangSmith Engine had scanned over 60 million traces and found more than 20,000 issues; its workflow connects diagnoses to proposed fixes, evaluation examples, and monitoring (Saltzgaber & Tannyhill, 2026). In a security investigation involving roughly 1,300 long transcripts, METR reported heavy reliance on "often-unreliable AI agents" (METR, 2026). As diagnostic outputs guide action, evaluating the wrong target can favor a system that answers a different question from the one its users need.

Outcome checks have known limits. In τ-bench, even $r = 1$ may not establish a successful episode: a return can reach the correct state without required user confirmation (Yao et al., 2025). Agent-as-a-Judge evaluates intermediate requirements (Zhuge et al., 2025). Our question concerns

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

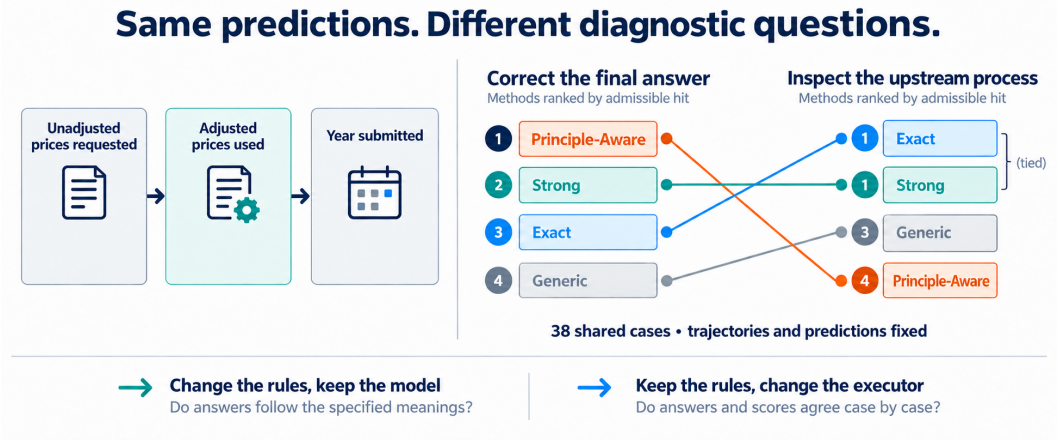


Figure 1: A diagnostic comparison needs a diagnostic question. Left: the opening example distinguishes the requested data, the source used, and the submitted answer. Right: changing the question reverses the leading method's position on 38 shared cases with frozen predictions; Exact and Strong tie under the process question. The two tests below examine how declared rules govern answers and whether another executor reproduces their behavior.

what the diagnosis itself must name. Even at one visible event, a submission, claim, source, or transformation may be relevant; alternatives or abstention may be appropriate. Process access and detailed requirements make evaluation richer, while the diagnostic specification determines which answers count.

We call these rules a *diagnostic specification*, or *contract*. The contract defines the object being scored and when alternatives or abstention are appropriate. Establishing that object is *diagnostic-object identification*. Our contribution is to make this choice experimentally testable. First, we change the rules given to a fixed model and predict which parts of its answers should change. Second, we hold the rules fixed and ask whether a model and a program reproduce the same mapping and scores case by case. These are the specification-sensitivity and implementation-stability tests, respectively.

Evidence-Path Contamination Evaluation (EPCE) is our executable test case. Its rules define acceptable diagnoses from observable events and source links, without adding observations or privileged task information. Many repairs concern events already found; semantic swaps isolate the components governed by each rule. External trajectories and a second executor test the reach and reproducibility of this mapping.

2 Same Predictions, Different Diagnostic Questions

2.1 A Leader Becomes Last When the Question Changes

The stock-price example admits two useful questions. The *commitment view* asks which accepted final answer or task claim should be repaired. The *process view* asks which upstream source, output, calculation, or state change should be inspected or repaired. Each question makes a different part of the same observed failure relevant to the diagnosis.

We apply these questions to the already-frozen candidate sets of 217 external trajectories. The commitment view retains a certified candidate containing an adopted terminal SUBMISSION or TASKCLAIM; the process view retains one containing an intermediate TOOLOUTPUT, SOURCE-PROVENANCE, CALCULATION, or STATEMUTATION. Here and throughout, "certified" means accepted by the reference-construction protocol. Filtering changes the acceptable set on 113/217 cases. The commitment view has nonempty sets on 88 cases and the process view on 66.

To compare the same cases, we use only the 38 trajectories with nonempty sets under both views. We rescore four frozen methods: *Generic SOL*, a model given a task-aware generic diagnosis instruction; *Principle-Aware SOL*, the model given diagnostic advice without complete decision rules; *Exact SOL*,

the model given the full contract; and *Strong*, a program executing that contract. Section 3 describes the controlled comparisons that produced these outputs.

Table 1: Frozen predictions, two diagnostic questions. Carrier-decomposition admissible hits on the same 38 cases: a hit means the named evidence carrier is accepted under the selected view. No predictor is rerun.

Frozen method	Correct final commitment	Inspect upstream process
Generic SOL	21/38	2/38
Principle-Aware SOL	**26/38**	0/38
Exact SOL	22/38	**3/38**
Strong	23/38	**3/38**

Principle-Aware leads under the commitment question and comes last under the process question (Table 1). Under commitment, the order is Principle-Aware $>$ Strong $>$ Exact $>$ Generic; under process, it is Exact $=$ Strong $>$ Generic $>$ Principle-Aware. The models have not become more or less capable between these two columns. The evaluator has changed what a prediction is asked to identify.

The low process scores concern predictions frozen before these two views were applied. In particular, the full external contract used by Exact and Strong prioritizes explicit submissions when selecting a terminal candidate, then applies completion and evidence rules. Rescoring those outputs for upstream inspection does not test predictors newly instructed for that purpose.

This is an exploratory existence result within the frozen candidate set, not an estimate of how frequently rankings reverse across benchmarks. It also does not identify a universally preferable question. It shows why that question must accompany the comparison. Even within the 38 shared cases, the two acceptable sets are disjoint on 18, overlap but differ on 17, and coincide on only 3. The change reaches the objects accepted as answers, not just the names attached to the metric.

2.2 Even the Same Event Can Name Different Answers

Choosing the diagnostic question is necessary, but a location alone may still be too coarse. In the released stock-price example, the final event e0129 submits a year. The reference allows both the adopted submission and the adopted task claim as semantic objects at this event (Figure 2). An output can name an acceptable object and still be wrong to insist that it is the only answer. Conversely, an output can preserve uncertainty without identifying any acceptable object.

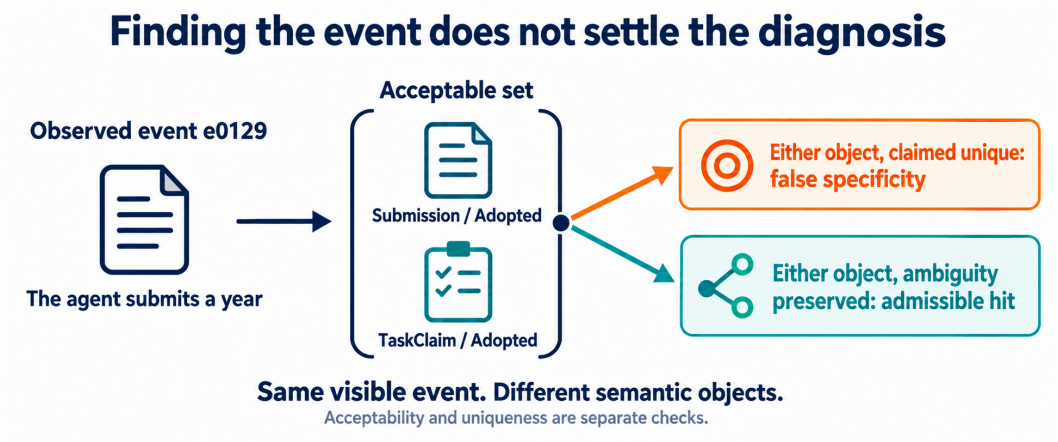


Figure 2: Locating an event and identifying a diagnosis are different decisions. This frozen worked example permits two semantic objects at event e0129. Either acceptable object can earn an admissible hit; claiming uniqueness without acknowledging ambiguity incurs false specificity. The two checks can therefore have different outcomes for the same output.

These distinctions require explicit rules. They determine which source, field, and transformation an answer names; which version of a revised claim counts; which claim was adopted; and when alternatives, nonlocalization, or abstention are warranted. More complete logging can reveal a source or a revision, but the rules determine how that observation affects the answer.

Let $G = (V, E)$ represent the observable trajectory, with events as nodes and their recorded relationships as edges, retaining source information and marking missing evidence. Let q be the task.

The contract C maps the record and task to permitted outputs:

$$C : (G, q) \mapsto \mathcal{H}_C(G, q) \cup \{\text{NoProp}, \text{Abstain}\}. \tag{1}$$

Here $\mathcal{H}_C(G, q)$ is the set of acceptable diagnoses; NoProp denotes a nonlocalizing output, and Abstain withholds a diagnosis under the contract's uncertainty rules. "Legal" and "admissible" mean permitted by this contract. The analysis uses observable records, not private chain of thought. Appendices A and D specify the formal record, taxonomy, completeness result, and scoring levels.

The ranking comparison changes this acceptable set while keeping predictions fixed. We now ask a different question: *when the rules are supplied to a predictor, do its answers actually follow them?* That requires changing the predictor's instruction and examining newly produced answers under controlled conditions.

3 Making Diagnostic Rules Testable

Two tests examine whether the contract governs behavior (Table 2). **Specification sensitivity (Axis S)** changes rules with the executor fixed and checks the affected answer components. **Implementation stability (Axis I)** changes the executor under fixed rules and checks whether answers and scores reproduce case by case.

Table 2: Controlled tests and the questions they answer. The two Axis S comparisons produce new answers under changed instructions. Axis I changes the executor under the same rules. The opening comparison instead rescores already-frozen predictions under two diagnostic questions.

Test	What changes; what stays fixed	Question answered
S1: internal	Rules change; model and 720 trajectories stay fixed	Which answer components follow which rules?
S2: external	Rules change; model and 217 external trajectories stay fixed	Do the distinctions matter on external records?
I: execution	Model becomes program; rules, 217 trajectories, reference, and scorer stay fixed	Can another executor reproduce the mapping and scores?

A contract available to both predictor and evaluator. Our test case, EPCE, uses recorded events and source links to define what the diagnosis names and what status it has. Generic SOL and Exact SOL are one-turn runs of the same frontier model with the same observations, task information, tools, and output format. Exact SOL additionally receives the complete contract C. Neither can see reference answers. One output is frozen per condition; the comparison concerns these matched runs.

The internal population crosses 240 tasks with three episode generators, yielding 720 trajectories and including 120 repeated scenario clusters. Under the contract, 350 cases permit a diagnosis at a location and 370 do not. The trajectories are not 720 independently authored templates. Appendix C.1 gives composition, truth-status balance, and uncertainty intervals clustered by task and scenario. A deterministic executor of C is retained as a supporting internal comparison.

Controls that predict a particular change. A detailed instruction could improve a score for several reasons. We therefore preserve the contract's structure and exchange specific semantic roles in three conditions, called *semantic placebos*. The swaps alter the rule-to-qualifier assignments in the supplied instruction; the input ZIP and output schema stay unchanged. P1 exchanges citation-set and transform roles; P2 exchanges citation-source and observation-source roles; P3 applies both swaps. The frozen swap map predicts the affected cases and answer components before inference. A behavioral

test should examine that response pattern, rather than only whether the total score falls (Ribeiro et al., 2020). These deliberate role swaps test rule execution; the opening commitment/process views ask two defensible diagnostic questions. For example, a missing required citation produces `citation-set` under the original rule, but `transform-obligation` under P1; the event and missing-citation condition are unchanged.

External trajectories and a second executor. The external population contains 73 AgentRx trajectories, all reproducibly retrievable trajectories in its frozen release, and 144 TELBench trajectories, a stratified confirmation sample selected without reference labels. S2 compares Generic SOL and Exact SOL on these 217 records with the model, reasoning effort, case order, observations, output format, reference, and scorer fixed. Principle-Aware SOL supplies diagnostic guidance without the complete decision rules, testing whether general advice recovers the effect.

Axis I compares Exact SOL with Strong under a shared semantic kernel K: fixed normalization, schema, identification rules, and guards. Exact receives these rules as a declarative text procedure; Strong executes their frozen code with its own input adapter. Strong's independently developed conversion and prediction logic translates observable benchmark fields into K's interface. This logic cannot read reference answers or scores, use case-specific exceptions, or modify K. The test changes how the rules are executed while retaining their meaning. Appendix D details this separation and the supporting Mini comparator; Appendix D.5 records the guidance prompt and the scope of its single frozen run.

What counts as success, and on which cases? Internal *joint exact* counts fully correct outputs. The external evaluation separates three questions. *Semantic exactness* checks the 97 cases with one certified semantic identity. *Admissible hit* checks the 194 cases with a nonempty certified set. *False specificity* counts unsupported claims of a unique answer on 113 eligible cases: 97 with multiple acceptable answers and 16 unresolved cases. These score-specific populations overlap and must not be added together.

On those 113 cases, an error requires a single answer without registering ambiguity, unresolvedness, or abstention. Acknowledging uncertainty avoids this error but need not earn an admissible hit. Seven cases with no admissible projection follow a separate uncertainty rule: abstain or explicitly state that no answer is admissible, without attaching a semantic or factual answer. They do not enter the 113-case score. We also compare the three main scoring outcomes jointly for each trajectory, the *semantic core-metric vector*. Prediction agreement and scoring agreement are reported separately.

A common reference for the comparisons. The external reference is model-assisted and protocol-certified, constructed without seeing predictions. Outputs are structurally checked and frozen before a single join to the common reference and scoring by the same frozen scorer. Appendix D documents construction and freeze order. The Reproducibility Statement points to public row-level recalculation, program-output reconstruction, and release scope.

4 How Rules Change Diagnoses

4.1 What Does the Large Improvement Repair?

Supplying the contract raises fully correct outputs from 233/720 to 716/720, a net gain of $67.08\,pp$. There are 485 repaired cases and two regressions, so repairs and net improvement are different counts. With model, inputs, tools, and output format unchanged, this comparison establishes a large response to the instruction change. To understand that response, we examine what the original answer got wrong.

The model can find the right event and still name the wrong object. Of the 485 repairs, 219 occur where Generic SOL already names the right parent event family once its semantic qualifiers are removed. Exact SOL corrects those qualifiers. Another 128 repairs recover an event family Generic SOL missed, and 138 correctly stop forcing a location (Figure 3, left). Thus 347 repairs concern localizable cases, with the remainder addressing whether a location should be supplied. The identity repairs mostly involve source, field, and transformation; 210/219 involve several of these dimensions together. Appendix C gives paired counts and the cluster interval.

As in the opening stock-price case, finding an event does not specify which evidence or claim the diagnosis names. The repairs concern both that identity and whether a location should be supplied.

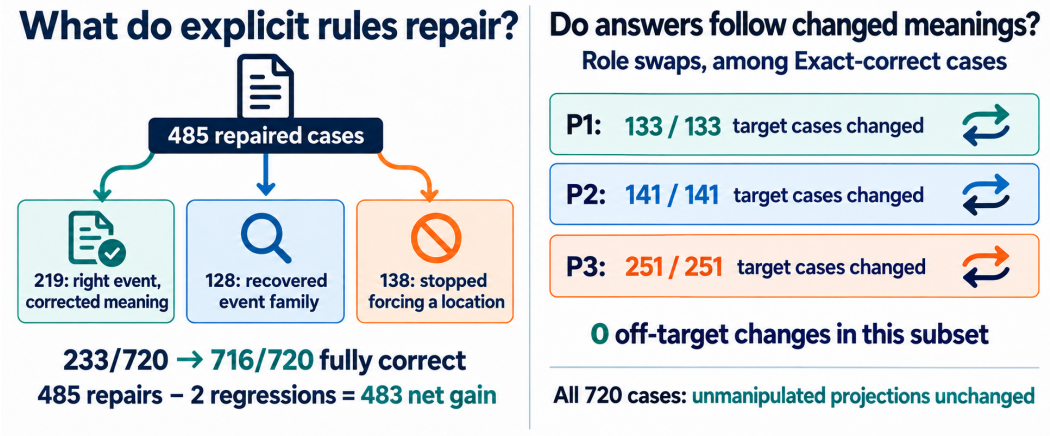


Figure 3: From improvement to a test of its explanation. Left: the 485 repairs separate corrected meaning at an already-located event family, recovered event families, and correctly withheld locations. Right: semantic role swaps change exactly the predicted targets among Exact-SOL-correct cases, with no off-target changes in that subset. The bottom-right statement is a separate all-720-case check of unmanipulated projections. Blocks show counts, not proportional areas.

4.2 Do the Answers Follow the Meanings We Change?

The gain alone does not show which aspects of a detailed instruction matter. The role swaps provide a more specific test. P1 swaps which roles denote the citation set and the transformation; P2 swaps which source is cited and which source was observed; P3 combines the two. If answers follow these meanings, they should change on the predicted components while retaining the others.

That pattern appears in the frozen runs (Figure 3, right). Among cases Exact SOL originally answers correctly, P1 changes its 133 predicted cases, P2 its 141, and P3 their 251-case union ($133 + 141 - 23 = 251$). No other cases in this correct subset change. Joint exact falls to 583/720, 575/720, and 465/720, respectively. Across all 720 cases, a separate check finds the parent event family, derived status, and every unmanipulated projection unchanged.

The outputs track the specific meanings that were exchanged. A longer prompt alone does not require this response pattern. Target matching is conditional on Exact-SOL-correct cases; preservation of unmanipulated projections is checked on all cases.

Do different rules support different decisions? The programmatic clause ablations provide a complementary test. Removing revision semantics loses 29 fully correct outputs. Removing the task-specification join, which connects trajectory evidence to task requirements, reduces coverage of localizable cases from 350/350 to 36/350. Without task definitions, the ablated executor cannot form submission-bound obligation qualifiers: an omitted citation is detectable as an error only against the required set. It retains trace-only failure rules (Appendix C). Removing integrity abstention retains that coverage but increases incorrectly forced locations from 2/370 to 13/370 (Table 3). These failures concern different decisions: the relevant version, whether evidence meets the task, and when a location is justified. Appendix C gives the full scoring profile and ablation design.

4.3 Do the Same Distinctions Matter on External Trajectories?

The internal construction shows where the rules act. The 217 external trajectories ask whether those distinctions also matter on records from other sources. Exact SOL improves both acceptable answers and uncertainty discipline relative to Generic SOL (Table 4). All 54 changed point-exact outcomes are improvements. Admissible hit gains 85 cases and loses one. False specificity removes 110 errors

Table 3: Programmatic contract ablations. Coverage is over 350 localizable cases; nonlocal false forcing (FF) counts failures over 370 cases. Joint exact counts fully correct outputs over all 720.

Contract condition	Joint exact	Coverage	Nonlocal FF ↓
Full programmatic C	**717/720**	**350/350**	**2/370**
− Revision semantics	688/720	347/350	4/370
− Task-specification join	402/720	36/350	4/370
− Integrity abstention	706/720	350/350	13/370

and introduces none. The same model, given the explicit rules, changes both what it identifies and when it claims a unique answer.

Table 4: External comparisons under one fixed contract and reference. The first two metrics count successes; false specificity counts unsupported claims of uniqueness (lower is better). General advice is compared with complete decision rules, and a program is included for the execution test in Section 5. Equal totals do not establish casewise agreement.

Metric	Generic	Principle-Aware	Exact	Strong
Semantic exact	0/97	7/97	**54/97**	**54/97**
Admissible hit	1/194	11/194	**85/194**	**85/194**
False specificity ↓	113/113	89/113	**3/113**	**3/113**

Would better general advice be enough? Principle-Aware SOL improves all three totals, but does not recover the effect of the complete rules. It removes 24 false-specificity errors; Exact SOL retains those improvements and removes 86 more. The point-exact successes, however, do not form the same progression: Principle-Aware's seven successes do not overlap Exact SOL's 54. Only one of its 11 admissible hits overlaps Exact SOL's hits. General advice improves caution in this frozen comparison, while explicit rules recover a different set of correct diagnoses.

This result separates general diagnostic considerations from an executable definition of the answer. Principle-Aware is a single frozen stochastic realization, so the casewise non-overlap does not establish distinct latent cognitive mechanisms or stability across runs. It does show that this stronger generic prompt fails to reproduce the contract's response pattern. Appendix D.5 reports the full transitions.

The external result also sets a useful limit: the contract does not make every diagnosis correct. Exact SOL obtains 54/97 semantic-exact answers and 85/194 admissible hits. Of the 109 hit failures, 40 still hit the accepted carrier but miss its semantic identity; other failures arise at earlier scoring layers (Appendix D.6). These external metrics and populations differ from internal joint exactness, so the two percentages are not a direct generalization-gap estimate.

5 Can Another Executor Reproduce the Diagnosis?

So far, changing the instruction has changed the answers in specific ways. The remaining question is whether those results depend on this model's reading of the rules. We now fix the contract, observations, reference, and scorer, and replace the model with Strong, the program described in Section 3.

Table 4 shows equal totals for Exact SOL and Strong on the three core metrics. Totals alone cannot establish stable execution: different successes and failures could cancel. We therefore compare both semantic predictions and scoring outcomes for each trajectory.

The returned semantic objects match after projection on 206/217 cases. All three main scoring outcomes match jointly on 213/217 cases (Figure 4). The second comparison allows different predictions to earn the same scores; it does not claim identical predictions. All four cases with different score vectors come from TELBench: two favor the model and two favor the program. These opposing differences leave the aggregate totals unchanged. All four concern object completion: one

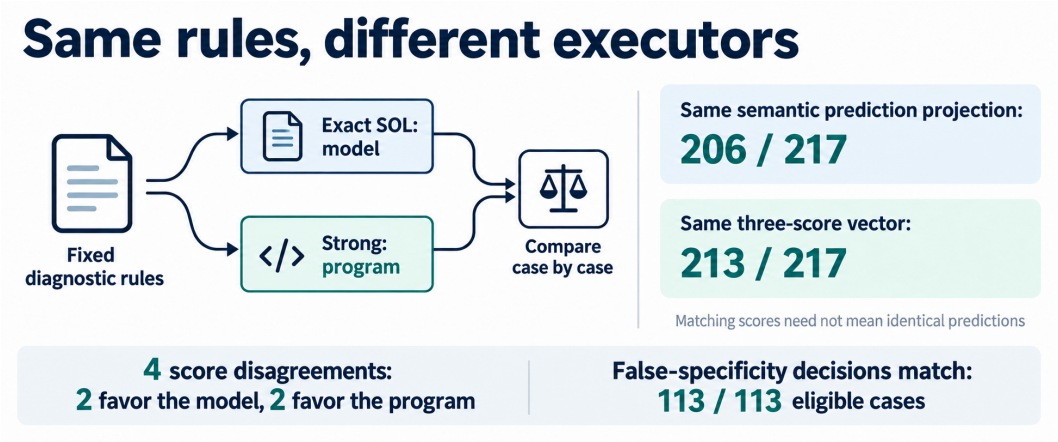


Figure 4: Cross-implementation agreement under the same semantic kernel, trajectories, reference, and scorer. Prediction projection compares returned semantic objects; the three-score vector compares their scoring outcomes. The four score disagreements balance in direction, explaining equal totals without identical casewise behavior. False specificity uses its separate 113-case eligible population.

executor abstains while the other names a submission. In each case, the frozen reference accepts the abstention (Appendix D.6).

For false specificity, the agreement is complete on the eligible set. Both executors avoid the same 110 errors and retain the same three; their decisions match on 113/113 cases. Appendix D.7 examines four levels of scoring resolution and reports each corpus separately.

The declared rules largely survive a change of executor in this population. The comparison shares a fixed semantic kernel. Strong's independent conversion and prediction logic is truth-blind, but it did not independently invent or reconstruct that kernel. The evidence supports reproducible execution of the specified mapping within this design.

Two executors can agree under one question while another defensible question changes the ranking of frozen methods. Reproducible execution does not select which scientific question to ask.

6 What a Diagnostic Comparison Establishes

The experiments connect the diagnostic question, its acceptable objects, and execution of its rules. The resulting order is $G \xrightarrow{C} \mathcal{H}_C(G, q) \rightarrow$ predictor comparison: start from the record, define acceptable diagnoses, then compare predictions.

6.1 Design the Evaluation Around the Question

Make the diagnostic target available before scoring. State the intended purpose, then define how events, sources, transformations, revisions, and adopted claims determine acceptable answers. Fix these rules before scoring and make them available to predictor and scorer. Distinguish missing evidence, multiple acceptable answers, and failure to find an answer identified by the rules.

Report answer quality and uncertainty separately. An acceptable answer can wrongly claim uniqueness, and an uncertain answer can miss every acceptable object. Report point exactness, admissible hit, and false specificity on their own eligible sets, with abstention and empty-set rules. Reduced false specificity does not imply recovered identity.

Test the rules before relying on the comparison. For a changed clause, predict which cases and answer components should respond, then check both affected and unaffected components. A lower total alone does not identify what the model followed. Give unchanged rules to another executor and compare answers and scores case by case: balanced disagreements can leave matching totals.

Carry the question into every reported ranking. A diagnostic result should state its question, available observations, identity and adoption rules, reference construction, permitted outputs, and score denominators. When the question changes, report changes in acceptable sets and eligible populations. Compare shared cases to isolate the question's effect from a change in which trajectories are scored.

Check diagnoses before they guide downstream action. AI-assisted incident investigations already use diagnoses to interpret recorded events (Hugging Face, 2026; METR, 2026). These reports establish operational stakes, not validation of our contract. Tie the scored diagnosis to its intended use, and test its answer and uncertainty rules before using the score to select a system.

6.2 Relation to Existing Evaluation Ideas

The distinction between a specification and its implementation is established in formal methods (Wing, 1990; Liskov & Wing, 1994; Meyer, 1992). Construct validity asks whether a measurement captures its intended object (Jacobs & Wallach, 2021), and evaluation frameworks make scenarios, metrics, artifacts, and semantic judge contracts explicit (Liang et al., 2023; Biderman et al., 2024; Looram et al., 2026).

Recent diagnostic work makes the measurement problem concrete. TelemetrySuffBench reports restricted views retaining 99.5–100% detection F1 but at most 0.5% origin-step accuracy, and separately tests when the evidence supports a unique origin (Zhu & Pu, 2026). He et al. (2026) distinguish action, task, and step evaluation; exposing downstream steps shifts judges' blame later in a controlled 70-episode experiment, which they interpret as collider bias. MP-Bench challenges a single deterministic attribution and reports stronger LLM attribution under a multi-perspective protocol (In et al., 2026). Ambig-DS shows that executable artifacts can silently commit to unintended prediction targets or evaluation objectives (Stoisser et al., 2026). These studies examine evidence sufficiency, measurement level, admissible perspectives, and task framing. Here we make the diagnostic object an explicit experimental variable: change its rules on fixed observations, then change the executor under fixed rules.

Progress, failure analysis, author-developed rubrics, and long-trajectory evaluation motivate explicit criteria (Ma et al., 2024; Liu et al., 2024; Starace et al., 2025; Wang et al., 2026b); other benchmarks expose steps, dependencies, and harmful spans (He et al., 2025; Barke et al., 2026; Wang et al., 2026a). Long-trace difficulty alone does not establish target ambiguity. Partial identification and set-valued prediction preserve compatible answers (Tamer, 2010; Sadinle et al., 2019; Angelopoulos & Bates, 2023); contrast sets and metamorphic testing examine controlled semantic changes (Gardner et al., 2020; Ribeiro et al., 2020; Chen et al., 2018).

Benchmark choice can change rankings (Dehghani et al., 2021), and multiverse analysis examines dependence on defensible analytical choices (Steegen et al., 2016). Pipeline underspecification studies different models under a shared training and evaluation target (D'Amour et al., 2022). Our opening comparison holds trajectories and predictions fixed and changes what the diagnosis is asked to name. Appendix F expands these distinctions.

Scope of the evidence. The conclusions concern frozen runs, populations, and views. The reversal establishes an occurrence, not prevalence or a stable winner. The reference is model-assisted and protocol-certified; implementation stability retains a shared semantic kernel. The appendix documents construction, blinding, freeze order, and shared-reference limitations.

7 Conclusion

A complete record does not complete an evaluation. On the same trajectories, two diagnostic questions can reverse the ranking of unchanged predictions. Explicit rules make the source of that dependence inspectable: targeted interventions change the governed answer components, and a second executor largely reproduces scoring under the same rules. Before asking which system diagnoses better, specify what it should diagnose and test whether the rules that define that object are being followed.

Reproducibility Statement

The anonymous V5 review artifact supports row-level checks of the reported comparisons and reconstruction of released program outputs from observable inputs. Appendix E maps claims to files and verification steps: readers can recalculate frozen external scores and paired transitions without importing the formal scorer, rerun Mini and Strong on all 217 external input packets without reference-answer access, and regenerate predictions for the 122-row internal public core and failure annex. Start with `V5_README.md` and `scripts/verify_v5_release.py` in the supplementary archive; `PAPER_SUPPORT_MAP.md` connects the reported analyses to their supporting records. Section 3 describes the controlled comparisons; Appendix D details prompts, reference construction, and scoring; Appendix A supplies formal assumptions and proofs. Appendix G specifies the release boundary: the artifact supports frozen-result recalculation and program reruns, while the full private 720-row internal dataset and fresh provider-model responses are outside its scope.

References

Anastasios N. Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023. doi: 10.1561/2200000101.

Shraddha Barke, Arnav Goyal, Alind Khare, Avaljot Singh, Suman Nath, and Chetan Bansal. AgentRx: Diagnosing AI agent failures from execution trajectories, 2026. URL https://arxiv.org/abs/2602.02475.

Stella Biderman et al. Lessons from the trenches on reproducible evaluation of language models, 2024.

Tsong Yueh Chen, Fei-Ching Kuo, Huai Liu, Pak-Lok Poon, Dave Towey, T. H. Tse, and Zhi Quan Zhou. Metamorphic testing: A review of challenges and opportunities. *ACM Computing Surveys*, 51(1):1–27, 2018. doi: 10.1145/3143561. URL https://doi.org/10.1145/3143561.

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D. Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *Journal of Machine Learning Research*, 23(226):1–61, 2022. URL https://jmlr.org/papers/v23/20-1335.html.

Mostafa Dehghani, Yi Tay, Alexey A. Gritsenko, Zhe Zhao, Neil Houlsby, Fernando Diaz, Donald Metzler, and Oriol Vinyals. The benchmark lottery, 2021.

Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1307–1323. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.findings-emnlp.117. URL https://aclanthology.org/2020.findings-emnlp.117/.

Jiawei He, Mengyu Shi, Jie Jia, Xikai Yang, and Dong Sun. What process evaluation of coding agents actually measures: Action, task, and step are three different levels, 2026. URL https://arxiv.org/abs/2608.22960v1.

Pengfei He, Zhenwei Dai, Bing He, Hui Liu, Xianfeng Tang, Hanqing Lu, Juanhui Li, Jiayuan Ding, Subhabrata Mukherjee, Suhang Wang, Yue Xing, Jiliang Tang, and Benoit Dumoulin. TRAJECT-Bench: A trajectory-aware benchmark for evaluating agentic tool use, 2025. URL https://arxiv.org/abs/2510.04550.

Hugging Face. Security incident disclosure—july 2026. Hugging Face blog, July 16, July 2026. URL https://huggingface.co/blog/security-incident-july-2026.

Yeonjun In, Mehrab Tanjim, Jayakumar Subramanian, Sungchul Kim, Uttaran Bhattacharya, Wonjoong Kim, Sangwu Park, Somdeb Sarkhel, and Chanyoung Park. Rethinking failure attribution in multi-agent systems: A multi-perspective benchmark and evaluation, 2026. URL https://arxiv.org/abs/2603.25001.

Abigail Z. Jacobs and Hanna Wallach. Measurement and fairness. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 375–385. ACM, 2021. doi: 10.1145/3442188.3445901.

Percy Liang et al. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023.

Barbara H. Liskov and Jeannette M. Wing. A behavioral notion of subtyping. *ACM Transactions on Programming Languages and Systems*, 16(6):1811–1841, 1994. doi: 10.1145/197320.197383.

Xiao Liu et al. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/e9df36b21ff4ee211a8b71ee8b7e9f57-Abstract-Conference.html.

Tadhg Looram, Lucas Nuzzi, Kyle Waters, and Steven Dillmann. AsymmetryZero: A framework for operationalizing human expert preferences as semantic evals, 2026. URL https://arxiv.org/abs/2605.04083.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. AgentBoard: An analytical evaluation board of multi-turn LLM agents. In *Advances in Neural Information Processing Systems*, volume 37, pp. 74325–74362, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/877b40688e330a0e2a3fc24084208dfa-Abstract-Datasets_and_Benchmarks_Track.html.

METR. Brief independent investigation of agents' behavior, reasoning and collaboration in the OpenAI / Hugging Face hacking incident. METR report, August 26, August 2026. URL https://metr.org/blog/2026-08-26-openai-hugging-face-incident-investigation/.

Bertrand Meyer. Applying design by contract. *Computer*, 25(10):40–51, 1992. doi: 10.1109/2.161279. URL https://doi.org/10.1109/2.161279.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4902–4912. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.442. URL https://aclanthology.org/2020.acl-main.442/.

Mauricio Sadinle, Jing Lei, and Larry Wasserman. Least ambiguous set-valued classifiers with bounded error levels. *Journal of the American Statistical Association*, 114(525):223–234, 2019. doi: 10.1080/01621459.2017.1395341.

Trammell Saltzgaber and Ben Tannyhill. New in LangSmith Engine: >2x better issue detection. LangChain blog, August 25, August 2026. URL https://www.langchain.com/blog/new-in-langsmith-engine-2x-better-issue-detection.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. PaperBench: Evaluating AI's ability to replicate AI research. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 56843–56873. PMLR, 2025. URL https://proceedings.mlr.press/v267/starace25a.html.

Sara Steegen, Francis Tuerlinckx, Andrew Gelman, and Wolf Vanpaemel. Increasing transparency through a multiverse analysis. *Perspectives on Psychological Science*, 11(5):702–712, 2016. doi: 10.1177/1745691616658637.

Josefa Lia Stoisser, Marc Boubnovski Martell, Sidsel Boldsen, Kaspar Märtens, and Robert Kitchen. Ambig-DS: A benchmark for task-framing ambiguity in data-science agents, 2026. URL https://arxiv.org/abs/2605.09698.

Elie Tamer. Partial identification in econometrics. *Annual Review of Economics*, 2(1):167–195, 2010. doi: 10.1146/annurev.economics.050708.143401.

Jiaming Wang, Ziteng Feng, Jiangtao Wu, Ruihao Li, Qianqian Xie, Yuxiang Ren, He Zhu, Xueming Han, Fanyu Meng, Junlan Feng, and Jiaheng Liu. Where do deep-research agents go wrong? span-level error localization in agent trajectories, 2026a. URL https://arxiv.org/abs/2606.02060.

Jiaxuan Wang, Yulan Hu, Wenjin Yang, Zheng Pan, Xin Li, and Lan-Zhe Guo. Aligning agents via planning: A benchmark for trajectory-level reward modeling. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23174–23200. Association for Computational Linguistics, 2026b. doi: 10.18653/v1/2026.acl-long.1062. URL https://aclanthology.org/2026.acl-long.1062/.

Jeannette M. Wing. A specifier's introduction to formal methods. *Computer*, 23(9):8–24, 1990. doi: 10.1109/2.58215.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations*, 2025. doi: 10.48550/arXiv.2406.12045. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1b126cc38b8638e07bef37e7b2bb72bf-Abstract-Conference.html.

Yunfei Zhang, Boyu Feng, Changhua Pei, Zexin Wang, Zhihuang Peng, Xinlong Liu, Hengyue Jiang, Difeng Ma, Jiayi Zhang, Yongzhou Yao, Yanan Zhao, Fei Sun, Yintong Huo, Zhaoyang Liu, Jingjing Li, Gaogang Xie, and Dan Pei. LongRCA Bench: Diagnosing responsible roles and root causes in long-horizon agent failures, 2026. URL https://arxiv.org/abs/2608.15242v3. Version 3, August 21.

Yuxuan Zhu and Peng Pu. TelemetrySuffBench: Is agent telemetry sufficient for failure-origin diagnosis?, 2026. URL https://arxiv.org/abs/2608.07899.

Mingchen Zhuge, Changsheng Zhao, Dylan R. Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, Yangyang Shi, Vikas Chandra, and Jürgen Schmidhuber. Agent-as-a-judge: Evaluate agents with agents. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 80569–80611. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhuge25a.html.

A Formal Details

This appendix specifies the objects used in the main text. It first records what the evaluator can observe, then defines how a contract turns those observations into a diagnostic answer. The proposition states what follows under those definitions and assumptions.

A.1 Observable record and terminal projection

Let $G = (V, E)$ be a predeclared, time-unrolled directed acyclic graph of observable events. Repeated tool calls create new event instances, so execution loops do not create graph cycles. For episode key k, task specification q, condition c, observation O_v, receipt metadata R_v, terminal record Y_{raw}, missingness M, and integrity disposition I, define

$$Z = (k, q, c, G, \{(O_v, R_v)\}_{v \in V}, Y_{\text{raw}}, M, I)\,. \tag{2}$$

Let J denote the fixed reference information and alignment used by the original evaluator, which remains available as $Y = g(Y_{\text{raw}}, J)$. For a controlled pair $(Z^{(0)}, Z^{(1)})$, the minimal changed frontier is

$$L = \{v : D_v = 1,\ \nexists u \prec v \text{ with } D_u = 1\}, \tag{3}$$

where D_v marks an observable difference at event v, and $u \prec v$ means that u is a strict ancestor of v in G.

Proposition 1 (Terminal projection)**.** Every eligible terminal decision is reproducible from (Z, J). If two admissible records share the same terminal projection but a specified diagnostic rule assigns them different answers, no rule using only that terminal projection can reproduce the diagnostic rule on both records.

Proof. Equation 2 retains Y_{raw}, so define $f(Z, J) = g(Y_{\text{raw}}, J)$. Any terminal-only decision rule therefore composes with f and is reproducible from (Z, J). If distinct records Z_1 and Z_2 share $f(Z_1, J) = f(Z_2, J)$, a rule defined only on the shared terminal value must return the same answer for both. By assumption, the specified diagnostic rule d satisfies $d(Z_1, J) \neq d(Z_2, J)$. It therefore cannot be reproduced on both records from the terminal projection alone. □

A.2 Completeness relative to a declared schema

Proposition 2 (Declared-schema completeness)**.** If each unit has a provenance-bound observation or explicit missingness for every $v \in V$, the empirical joint distribution over declared observations, its marginals, and every conditional transition with nonzero denominator are identified within that schema. For a fixed diagnostic rule d on admissible records, an event-removing projection π preserves the information needed to recover d exactly if and only if

$$\pi(z_1) = \pi(z_2) \implies d(z_1) = d(z_2) \quad \text{for all admissible } z_1, z_2.$$

The rule may return a set of diagnoses; equality then means equality of sets.

Proof. The record contains an observed value or an explicit missingness symbol for every declared coordinate. Empirical joint and marginal counts are therefore defined directly, and conditional transitions are ratios of joint counts when their denominator is nonzero. If two records share a projection but have different diagnostic outputs, any rule on the projection fails on at least one. Conversely, under the displayed condition, define $h(\pi(z)) = d(z)$ on the projection's image. This is well-defined because every record with the same projection has the same output, and $d = h \circ \pi$. Different paths may thus be collapsed when the target diagnosis is unchanged. □

This is observational completeness relative to the declared schema. It does not include hidden reasoning, undeclared latent state, or a universal causal model.

B SUPPORTING IDENTIFICATION EVIDENCE

B.1 POSITIVE EVIDENCE AND LICENSED CONCLUSIONS

The main argument uses several experiments, each answering a different question. Table 5 links each frozen result to the conclusion it supports and states where that conclusion stops. These results are not combined into a single score.

Table 5: Positive evidence ledger. Every signal is already frozen; the last column prevents a local success from being promoted into a broader claim.

Evidence role	Positive signal	Licensed conclusion	Boundary
Internal sensitivity	Joint exact rises 233/720 → 716/720 with model, trajectories, tools, and output schema fixed	Declared semantics has large operational force on the decision function	Not by itself evidence of one natural ontology or cross-implementation stability
Clause localization	Three role swaps change 133/133, 141/141, and 251/251 predeclared target rows with zero off-target changes on Exact-correct support	The manipulated clauses control their intended semantic roles	Support is conditional on the predeclared Exact-correct rows, not a full factorial audit
External sensitivity	Semantic exactness 0/97 → 54/97, admissible hit 1/194 → 85/194, and false specificity 113/113 → 3/113	The specification treatment transports to two external corpora and improves both recovery and commitment discipline	A bounded external witness, not a prevalence estimate
Informed-generic control	Principle-Aware reaches 7/97 exact, 11/194 hit, and rescues 24 false-specificity failures with no new one	Ordinary diagnostic competence has a real positive effect on caution	Single-realization inference; one-run/no-revision guards against adaptive selection
Implementation stability	Exact SOL and Strong match on 213/217 semantic core-metric vectors and all 113 eligible false-specificity decisions	One declared object is reproducibly realizable through heterogeneous operational paths	Two-realization scope; further paths require predeclared adaptation and freeze rules
Resolution profile	Agreement is 208/217 at disposition, 210/217 at localization, 210/217 at carrier decomposition, and 213/217 at identity	The cross-realization signal persists across progressively finer scoring layers	A robustness profile, not four independent replications
Evaluation consequence	Coherent commitment and process views differ on 113/217 admissible sets and reorder four frozen methods on 38 common-support rows	The scientific question can change the legal repair target and method ranking with observations and predictions fixed	Post-hoc filtered existence witness, not a frequency estimate
Recalculation and custody	Structural intake passes 217/217; scorer-free recount reproduces row metrics; V4 binds 442 files and 14 verifier checks	The reported evidence is independently inspectable and byte-bound	Structural closure does not prove scientific truth or universal validity

B.2 MECHANISM-BEARING 120-ROW PUBLIC CORE

The released V4 package contains a compact public core selected before this manuscript revision: 60 localizable rows, 30 no-error rows, and 30 other nonlocalizable rows. Its purpose is diagnostic rather than population-level. Table 6 shows that the full executable contract is nearly exact on this core, while targeted removals damage the endpoint predicted by the omitted clause. The same package records a 60/60 local-exact advantage over the task-aware minimal-hitting-set comparator; no significance claim is attached to this descriptive subset.

Table 6: Frozen 120-row public core from V4. False forcing is a failure count (lower is better); the ablations alter one contract component at a time.

Method or contract variant	Joint exact /120	Local family exact /60	Nonlocal false forcing /60	No-error false forcing /30
Full executable contract	**119**	**60**	**1**	**0**
− revision semantics	114	55	1	0
− task–specification join	66	7	1	0
− integrity abstention	118	60	2	0
AgentDebugX core heuristic	30	1	31	30
Task-aware earliest anomaly	32	0	28	27
Task-aware minimal hitting set	32	0	28	27

The pattern is clause-specific. Removing the task–specification join preserves the nonlocalizable guard but collapses local family exactness from 60/60 to 7/60. Removing revision semantics produces a smaller, localized 60/60 to 55/60 loss. Removing integrity abstention leaves local exactness intact while doubling the already small nonlocal false-forcing count from one to two. These are mechanism-bearing positive controls for the executable witness, not a claim that the 120-row core is an independent benchmark.

B.3 Complete semantic endpoint profile in the released audit

The main text selects point exactness, admissible hit, and false specificity because they are sufficient for the identification argument. V4 also retains the two row-based secondary semantic endpoints in Table 7. Certified-family coverage gives partial credit for the fraction of a legal family recovered; uncertainty discipline checks the frozen status-to-commitment mapping on all 217 rows.

Table 7: Secondary semantic-identity endpoints from the V4 four-method audit. Parentheses are frozen 95% corpus-and-task-cluster-stratified row-bootstrap intervals over 10,000 draws.

Method	Certified-family coverage	Uncertainty discipline
Mini	29.73% [24.32, 35.44]	48.85% [42.86, 54.84]
Generic SOL	0.17% [0, 0.53]	44.70% [38.71, 50.69]
Exact SOL	**34.71%** [28.98, 40.58]	**89.40%** [85.25, 93.09]
Strong	**34.71%** [29.02, 40.65]	88.94% [84.79, 93.09]

The package therefore contains a positive signal beyond exact matches: the two fixed-contract realizations recover the same mean fraction of certified legal families and follow the declared uncertainty boundary on about 89% of all rows. The audit forbids significance claims for this comparison. It also reports binding-level unsupported rates—39/146 for Mini, 108/217 for Generic, 34/169 for Exact, and 0/5 for Strong—but those method-specific denominators are not row-comparable and are retained only as descriptive diagnostics.

B.4 RAG-DAG detail

The controlled Hotpot acquisition uses 24 hard tasks, six onset stages, two regular repeats per stage, and three identity shams per task. Every regular onset is singleton-localized (288/288), while identity shams return no forced stage (0/72). Among 240 regular pairs with strict terminal signatures, 227 belong to terminal-equivalence groups containing multiple path profiles.

R8B, the frozen cross-dataset retrieval study, contains 2,880 locked pairs and 720 identity shams across QASPER, NQ-Open, BM25 and dense retrieval, and three generator families. Its 60 ranker no-ops remain in the denominator. R9, the separate fresh-task ranker study, uses fresh ranker-eligible tasks: both QASPER and NQ-Open obtain 358/360 strict joint successes, 360/360 eligibility, and 360/360 exact earliest-divergence localization. R8B and R9 remain separate studies.

B.5 STATEFUL-AGENT DETAIL

The frozen retail sandbox schedules 240 paired forks: 192 regular injected state interventions and 48 live-twin shams. It retains 239 complete pairs and one incomplete MiMo pair. The locator identifies an ordered first-observable chain in 167/192 regular pairs and forces none in 48 live twins or 192 derived zero-call identity shams. Visible exposure occurs in 180/192 regular pairs, with safe recovery in 175/180. Same-terminal collisions occur in 48/192 pairs; 36/48 still contain a located ordered chain.

The prespecified pooled regular-minus-sham contrast is 0.870 with a familywise simultaneous 95% interval $[0.786, 0.948]$ from 20,000 scenario-cluster bootstrap draws. The endpoint-block results are 42/64, 64/64, and 61/64 regular located chains for the DeepSeek, Codex-labelled, and MiMo blocks, respectively. Independent recomputation rebuilds all commitments, joins, and bootstrap replicates exactly.

B.6 BROADER DIAGNOSTIC SETTINGS

The present evidence directly supports state revision, evidence transformation, and general tool-use diagnosis. Revision is an explicit contract clause whose removal loses 29 joint successes in the programmatic ablation. Source, field, and transform are the dominant dimensions in the 219 compositional misses, and the same semantic intervention transports to external research-agent trajectories. These results show why an event location can be visible while its qualified identity or commitment status remains unspecified.

The same distinction is likely to become testable in persistent memory, multi-agent delegation, human-agent approval flows, and runtime safety monitoring. A stored claim may be overwritten and reread; a relay may summarize another agent; a proposal may or may not become an adopted action; observed unsafe content may or may not become an executed commitment. These are not empirical claims of the present study. They are settings in which the gap between observable events and declared diagnostic semantics yields concrete, falsifiable evaluation questions.

C MATCHED-TREATMENT DETAIL

This appendix explains where the 720 internal trajectories come from, what information each condition receives, and how their answers are compared. The population and freeze details matter because the experiment changes the instruction on the same cases; it does not compare separately sampled runs.

C.1 INTERNAL POPULATION AND CONSTRUCTION

The internal population crosses 240 frozen task definitions with one archived episode from each of three generator models (DeepSeek Flash, GPT-5.6 Luna-Codex, and MiMo-v2.5), yielding exactly $240 \times 3 = 720$ trajectories. The task definitions cover four mechanism families rather than one repeated surface form (Table 8). They also span 60 tier-3, 120 tier-4, and 60 tier-5 definitions. Task identities and answer content are removed from the predictor-facing package, but the observable trace retains event order, tool status, source and field bindings, revisions, typed transform receipts, task obligations, submission state, and integrity flags.

Table 8: Composition of the internal 720-row population. Each task contributes one episode from each of three generators.

Task family	Tasks	Rows	Primary construction target
Entity alias / cross-record join	72	216	Alias collisions, source binding, and multi-record joins
Stateful revision / stale memory	72	216	Revision, supersession, adoption, and stale-state reuse
Temporal / jurisdictional supersession	48	144	Validity windows, authority handoffs, and time-zone joins
Ambiguity / abstention	48	144	Set-valued answers, missing support, and legal nonlocalization
Total	240	720	Complete frozen crossing

The 350 localizable rows are 198 point-identifiable plus 152 observationally ambiguous cases. The 370 legally nonlocalizable rows are 184 no-observable-error, 182 unjudgeable, and 4 not-identifiable cases. These are the complete frozen episode outcomes, not a post-result downsample to an exact 50/50 balance. The construction intentionally reuses scenario semantics: 120 scenario titles each induce two task definitions at different scope/tier settings, and each definition has three model episodes. There are therefore 120 six-row scenario templates, not 720 independent templates.

The reported paired bootstrap samples the 240 task ids; each sampled cluster carries all three model episodes. With 20,000 draws and seed 20260811 it gives $[64.17, 70.00]$ *pp*. A new conservative sensitivity clusters the two related task definitions and all six episodes by shared scenario title; its 120-cluster interval is $[64.44, 69.58]$ *pp*, leaving the conclusion unchanged.

Chronology and non-circularity boundary. The task definitions, original episodes, and row truth were copied byte-for-byte into the frozen internal package on 10 August 2026. The deidentified common SOL input was built on 11 August; Generic and Exact outputs were then frozen before their one-time join to truth. Hence neither condition nor the scorer was revised against the A/B row outcomes. This internal experiment is a matched specification intervention, not a claim that the EPCE ontology was invented before every underlying task or trajectory existed. Independent ontology construction is therefore not inferred from these 720 rows; the external prediction-blind reference and the exploratory competing-view analysis address that distinct concern.

C.2 Matched information boundary

Table 9: Experiment 1 holds the model and observable information fixed. The explicit semantic contract is the treatment.

Input or capability	Generic SOL	Exact SOL	Programmatic C
Observable event graph	Yes	Yes	Yes
Task specification and qualifiers	Yes	Yes	Yes
Integrity / missingness state	Yes	Yes	Yes
Outcome truth	No	No	No
Explicit propagation contract	No	Yes	Yes
Deterministic execution	No	No	Yes

C.3 Custody

The common input digest is `defff46c100ae558df8c694bd1af42837182a9b49ba53d503b97ef1616532ab1`; the pre-run bilingual prompt archive digest is `68ff1d4626844f3b2fe0b5b66bfdf353bfdb9f75ce734e3aa28a44e14961467d`. The separate Generic SOL and Exact SOL conversations record one completed turn each, `gpt-5.6-sol`, xhigh reasoning, and identical available-tool name sets. The preserved return ZIP digests are `08d0862ba5f492f0c8bc19978860221201cbe8818f23d8a86f6872b5c0ac79c7` and `bca536fc83b488f74a62e3582e96e722d5e03225e00ac7139df1d887d3d0bd70`.

C.4 JOINT SCORER AND RESCUE DECOMPOSITION

The paired Generic–Exact matrix contains 231 both-correct rows, 485 improvements, two regressions, and two both-wrong rows. The net contract effect is $483/720 = 67.08\,pp$, with a 20,000-draw task-cluster bootstrap interval of $[64.17, 70.00]\,pp$.

Table 10: Full matched joint-scorer profile. FF denotes false forcing.

Condition	Joint exact	Local exact	Coverage	No-error FF	Nonlocal FF
Generic SOL	233/720	1/350	1/350	133/184	138/370
Exact SOL	**716/720**	**348/350**	**350/350**	**0/184**	**2/370**
Deterministic EPCE	717/720	349/350	350/350	0/184	2/370

Figure 3 reports the mechanism decomposition in the main results. Among the 219 compositional misses, source differs on 154 rows, field on 160, transform on 105, revision on 24, and bare-event collapse occurs on 89. These identity-domain counts overlap by construction: 210 rows differ on at least two dimensions and 131 misbind source and field together. The 138 nonlocalizable rescues comprise 133 NO-OBSERVABLE-ERROR and five UNJUDGEABLE rows.

C.5 SEMANTIC-PLACEBO AUDIT

Among rows correct under Exact SOL, the P1, P2, and P3 observed error supports equal their frozen predicted supports with zero symmetric difference. P3 is exactly the union of the P1 and P2 supports, including their 23-row intersection. Parent-event family, outer/inner family shape, derived status, and all nonintervened identity projections remain invariant on every row. This audit checks semantic sensitivity against a pre-inference permutation map; it is not a separately randomized factorial endpoint.

Let S_j be the Exact-SOL-correct rows changed under placebo j, and let P_j be the frozen support predicted by its semantic-role permutation. The complete support audit is

$$\begin{gathered} S_j = P_j \quad (j \in \{1,2,3\}), \\ (|S_1|, |S_2|, |S_3|) = (133, 141, 251), \qquad S_3 = S_1 \cup S_2, \qquad |S_1 \cap S_2| = 23. \end{gathered} \tag{4}$$

C.6 SUPPORTING COMPARATORS AND ABLATIONS

The task-specification-join ablation removes access to task-definition members before its predictions are frozen. The program returns an empty family at a successful submission because its obligation qualifiers require those definitions. For example, the trace can list the sources actually cited, but cannot identify an omitted required citation without the required source set. Without a successful submission it retains observable error-code rules, including failed transforms and invalid note references. This explains the asymmetric loss of coverage: the retained trace-only branch supplies the remaining 36 localizable hits. This intervention removes the task join's information as well as its associated rules; it does not isolate wording alone. The internal program is labeled *Programmatic C* or *Deterministic EPCE* in the supporting tables. *Strong* denotes the external programmatic realization with its benchmark-specific adapter.

Table 11: Supporting internal comparators and programmatic contract ablations. TA-MHS and TA-EA denote task-aware minimal hitting set and earliest anomaly; their aggregate scores coincide on this frozen population.

Method	Joint	Local	Coverage	No-error FF	Nonlocal FF
TA-MHS	211/720	0/350	0/350	156/184	159/370
TA-EA	211/720	0/350	0/350	156/184	159/370
AgentDebugX core	184/720	6/350	6/350	184/184	192/370
Ablations of the full executable contract					
− Revision semantics	688/720	322/350	347/350	0/184	4/370
− Task-specification join	402/720	36/350	36/350	0/184	4/370
− Integrity abstention	706/720	349/350	350/350	0/184	13/370

D External-Transport Detail

The external study asks whether the contract affects answers on trajectories from other sources and whether a program reproduces the model's scores. The sections below specify which trajectories enter, how their reference answers are constructed, what each executor may access, and how each score is computed. These details distinguish external source data from the new reference constructed under our diagnostic rules.

D.1 Frozen population, scoring layers, and custody

At the frozen snapshot, the AgentRx artifact exposes 29 τ-retail and 44 Magentic-One trajectories with reproducibly retrievable raw traces and native labels. All 73 enter the transport denominator. The 42 Flash trajectories reported by the upstream paper are absent from that public snapshot. TELBench contributes a gold-blind, stratified 144-row confirmation sample; a disjoint 12-row development set is used only to develop and freeze the semantic projection guide before the confirmation sample is scored. Gold error spans, annotations, metadata, and judge outputs are not model inputs.

D.2 Reference construction and certification

The external reference is model-assisted and protocol-certified; it is not a conventional panel of credentialed human domain annotators. Human authors designed, authorized, and reviewed the protocol. The semantic work was then instantiated in clean isolated Codex tasks locked to `gpt-5.6-sol/xhigh`, with no task-specific fine-tuning or claim of professional domain credentials. Each instance received only its frozen role prompt, rules, schema, and authorized blind packet. This distinction is material and is reported explicitly rather than relabeling model roles as human experts.

For each of 55 outcome-independent batches (target four rows), two fact extractors were reconciled by one fact reconciler; two independent candidate generators were followed by one adversarial challenger; three admissibility reviewers then voted on a provenance-blind, independently permuted candidate union. At least two ADMISSIBLE votes accepted a candidate and at least two INADMISSIBLE votes rejected it. All other patterns triggered a minimal frozen resolver/recheck packet. Reviews froze independently before cross-review. Finally, two identification auditors independently canonicalized the aggregate at disposition, localization, carrier-decomposition, and semantic-identity layers; exact agreement was required before reference freeze.

Table 12: Reference-certification protocol and realized quality statistics. Votes are ordered admissible/inadmissible/unresolved.

Stage	Frozen aggregation rule	Realized audit
Fact and candidate search	2 extractors $\rightarrow$ reconciler; 2 generators + adversarial challenger	Checklist-stopped search; completion is process-relative, not exhaustive natural truth
Admissibility	3 blind independent reviewers; ≥ 2 votes decide, otherwise resolver/recheck	576 candidates: 380 unanimous, 196 nonunanimous; 106/217 rows contain a split candidate
Resolver	Minimal conflict packet only	27 rows contain an unresolved vote; resolver changes the final admissible set on 9 rows
Identification	2 independent auditors; exact canonical match required	217/217 row agreement; 0 audit mismatches
Final status	Point, multiple, no projection, or unresolved	97 / 97 / 7 / 16 rows

D.3 External-reference coverage and challenger-removal robustness

The two-generator union froze at 490 candidates before challenger generation; the final union contains 576. The V5 derivative joins stages by (`row_id`, `semantic_identity_sha256`), not candidate ID. All 490 initial objects have a field-for-field semantically identical final object and zero mismatch. Only 400 retain the same candidate ID; 90 are renumbered by final sorting.

Table 13: Post-hoc source decomposition of the frozen final candidate union.

Source class	Candidates	Initial member
G1 only	228	Yes
G2 only	173	Yes
Both generators	89	Yes
Challenger only	86	No
Total	**576**	490 initial

The 86 challenger-only candidates occur on 83 rows. Adjudication yields 68 admissible, 16 inadmissible, and 2 unresolved candidates; 66 rows gain at least one admissible challenger-only candidate. Removing the full class and reconstructing status and denominators from the initial union preserves the Generic-to-Exact direction: semantic exactness is 0/109 versus 58/109, hit is 1/200 versus 87/200, and false specificity is 98/98 versus 3/98. Strong equals Exact on all three restricted-reference totals. This is a post-hoc sensitivity analysis over frozen scientific bytes, not a second experiment or evidence that the initial or final search is exhaustive.

Table 14: Hash-bound public phase ledger. Sequence preserves order without publishing wall-clock times that could create anonymous linkage.

Seq.	Phase	Prediction-blind	Truth opened	Public binding
1	Initial generation	Yes	No	Private receipt/output commitments
2	Initial union freeze	Yes	No	Eight initial unions and commitments
3	Challenger generation	Yes	No	Private receipt/output commitments
4	Final union freeze	Yes	No	Eight released final unions and commitments
5	Adjudication and resolution	Yes	No	Eight resolved aggregates
6	Reference freeze	Yes	No	Freeze receipt and independent validation
7	Prediction freeze	Yes	No	Pre-run custody/freeze receipts
8	Prediction truth join	No	Yes	Truth-join receipts and frozen score bundles

The public provenance derivative contains row ID, semantic-identity hash, initial/final candidate IDs, initial membership, a four-class source label, and aggregate adjudication/reference bindings. It excludes thread and semantic-turn IDs, provider response/account IDs, machine paths, local candidate IDs, and orchestration details. Original supervisor provenance and freeze receipts stay private but are SHA-256 committed. The labels *supervisor-only* and `never_dispatch_to_semantic_roles` were construction-time custody rules, not permanent publication bans. This minimized disclosure was generated only after reference construction, adjudication, prediction freeze, and truth join; it therefore cannot retroactively defeat blinding.

Circular adjustment is blocked at three boundaries. Candidate authors are hidden from reviewers; native labels, A/B outputs, old predictions, placebo outputs, and all score/probe artifacts are forbidden before reference freeze; and the prediction custodian cannot participate in reference construction. The final freeze validation confirms 217 distinct rows, no active technical quarantine, and no prediction or probe artifact read. The certified family is therefore relative to the declared search and adjudication process. It is not claimed to enumerate every naturally defensible diagnosis, which is why the secondary contract-view consequence check is kept explicitly exploratory.

D.4 Frozen semantic kernel and programmatic realization boundary

What is held fixed. Let K denote the frozen EPCE semantic kernel: the observable-trace and closure normalizers, canonical prediction schema, contract vocabulary, core identification semantics, and fail-closed scientific guards. For programmatic release $r \in \{\text{Mini}, \text{Strong}\}$, L_r is a release-local, deterministic conversion and prediction layer. For observable packet o_i, the operational path is

$$o_i \xrightarrow{L_r} e_{ir} \in \mathcal{E}_K \xrightarrow{K} \hat{y}_{ir}, \qquad f_{K,L_r}(o_i) = K(L_r(o_i)), \tag{5}$$

where L_r translates a benchmark-specific schema into a K-valid evidence object and K supplies the diagnostic meaning. L_r may parse source-specific fields, bind observable carriers, and project candidate objects. It may not read truth, reference status, labels, scores, or another release's outputs; branch on row identity; encode row exceptions; tune on scored outcomes; or modify K. During Strong's external development, only L_{Strong} was editable; the core EPCE identification code within K remained frozen and byte-identical. Each programmatic release embeds its own frozen copy of K and executes without runtime import or filesystem dependency on the other release.

What "independently developed" means. Experiment 3 does not claim that two realizations independently invented the same semantics. Exact SOL is instructed by the declared K, whereas Strong uses independently developed truth-blind release-local conversion/adaptation and prediction logic around the same K. The controlled comparison is

$$K \text{ fixed}, \qquad I_{\text{Exact}} \to I_{\text{Strong}}, \qquad f_{K,I_{\text{Exact}}}(x) \approx f_{K,I_{\text{Strong}}}(x). \tag{6}$$

Thus independence applies to the operational implementation boundary, not to a clean-room rewrite of the semantic object. This shared kernel is the necessary control for the question: replacing it with a separately reconstructed K' would compare f_{K,I_1} with f_{K',I_2}, so disagreement could reflect specification drift rather than implementation choice. Byte-identical kernel components document this controlled freeze; they are not a cross-release runtime dependency.

Confirmatory freeze and local adaptation. The EPCE semantic kernel is immutable within a confirmatory evaluation once frozen. In particular, confirmation gold, error spans, row-level error patterns, aggregate scores, and judge outputs cannot be used to revise the current K. Scientific revision remains possible between studies through a new, explicitly versioned kernel; it is not permitted retroactively on the population used to confirm the current version. What may adapt locally is the truth-blind interface layer L_r needed to parse a new benchmark or harness, and that layer must itself be frozen before confirmatory scoring.

TELBench makes the timing concrete. A disjoint 12-row development set was used to develop and freeze the semantic projection guide for the new observable schema. The guide and release-local interface logic were frozen before the 144-row confirmation sample was scored; confirmation gold annotations, error spans, metadata, and judge outputs were never prediction inputs. The later Exact SOL follow-up was designed after the Generic SOL and Strong aggregate returns, but it reused K unchanged and froze its prompt, output, and predictions before its one-time truth join. The design principle is: *freeze the specification globally within confirmation; adapt only truth-blind interface realizations locally.*

Candidate identities are projected through four promoted maps. A row is point-identified when one equivalence class remains at a layer, multiply admissible when several remain, and unresolved when certification fails. Point exactness uses only point-identified rows; admissible hit asks whether the prediction belongs to any certified class. Denominators consequently change with semantic resolution and are never pooled across layers.

Let $\mathcal{A}_{i\ell}$ be the certified admissible set for row i at layer ℓ, and let $\hat{a}_{i\ell}$ be the corresponding prediction. We score

$$\begin{aligned} \mathcal{I}_\ell &= \{i : |\mathcal{A}_{i\ell}| = 1\}, \qquad \mathcal{E}_\ell = \{i : |\mathcal{A}_{i\ell}| \geq 1\}, \\ \text{Exact}_\ell &= \frac{1}{|\mathcal{I}_\ell|} \sum_{i \in \mathcal{I}_\ell} \mathbf{1}\{\hat{a}_{i\ell} \in \mathcal{A}_{i\ell}\}, \\ \text{Hit}_\ell &= \frac{1}{|\mathcal{E}_\ell|} \sum_{i \in \mathcal{E}_\ell} \mathbf{1}\{\hat{a}_{i\ell} \in \mathcal{A}_{i\ell}\}. \end{aligned} \tag{7}$$

Semantic false specificity is evaluated on 97 multiply-admissible and 16 unresolved rows. It equals one only when the predictor emits exactly one interpretation and registers none of ambiguity, unresolvedness, or abstention. These uncertainty-preserving outputs score zero on false specificity; their treatment by other endpoints is status-specific. Cross-implementation analysis additionally compares frozen status, normalized prediction projection, and the complete row-level metric vector at every layer.

All four 217-row candidate sets are immutable score bundles with verified SHA-256 manifests. Exact SOL's returned ZIP digest is `5b82eb8f485bc8bef8f25359acc9b07d71e1490113f7b0311276da09523354fe`, and its frozen prediction digest is `fb292bc6dd36882b52af83ff725df0bb394422b60f21aaa2dd50994f118d9eb0`. Structural intake passes on 217/217 rows with zero technical failures. The truth join is activated once after prediction freeze; Mini, Generic SOL, Exact SOL, and Strong are then scored by the same frozen scorer. An independent scorer-free recount reproduces every row metric and aggregate.

D.5 Principle-Aware robustness control

Role and timing. This post-primary control was added to examine whether Generic SOL's behavior could reflect the absence of ordinary agent-diagnosis advice. It is therefore reported as a reviewer-motivated robustness analysis rather than as a preregistered fourth main experiment. Before the condition was run, we froze the complete prompt, model (`gpt-5.6-sol`), reasoning effort (`xhigh`), exact 217-row input, generic prediction specification, output schema, three endpoints, paired transition analysis, and a one-run/no-prompt-revision rule. The byte-identical Generic input

ZIP was supplied as the sole authorized archive through a frozen local path binding rather than a new UI upload; no other input bytes changed. The returned ZIP was preserved before one truth join to the same frozen reference and scorer.

Design boundary and inference license. The control evaluates one prospectively frozen stochastic realization, $f_{\mathrm{PA}}(x)$. A repeat-run protocol would instead characterize a distribution $P(\hat{y} \mid x, \mathrm{PA})$ or a declared aggregation operator $A(\hat{y}^{(1)}, \ldots, \hat{y}^{(R)})$. Such a protocol would require advance choices about run count, row- versus run-level aggregation, run selection, and whether prompt adaptation is permitted. The single realization therefore does not estimate expected performance, run-to-run variance, or stochastic repeatability. At the same time, the one-run/no-revision rule keeps observed outcomes from determining additional runs or prompt changes, protecting the paired treatment contrast from adaptive selection. The resulting claim is correspondingly local to the frozen realization and its prespecified row-level transitions.

Rule-incomplete treatment. The added instruction asks the model to inspect six dimensions across each complete packet: source provenance, field binding, evidence transformation, revision, ambiguity/evidence closure, and nonlocalization/abstention. It explicitly states that these are considerations rather than a resolution algorithm. It does not define semantic-identity equivalence, which object remains active after revision, when a singleton is licensed, any precedence or resolution hierarchy, or a canonical semantic mapping. It contains no row exception, reference, score, prior prediction, or truth.

Table 15: Complete frozen Principle-Aware endpoints and paired row structure. For exactness and hit, $+/-$ are $0 \to 1/1 \to 0$. For false specificity, "rescued/new" are failure→success and new failure.

Endpoint	Generic	Principle-Aware	Exact	Generic→PA	PA→Exact
Semantic exact	0/97	7/97	54/97	$+7/-0$	$+54/-7$; shared 0
Admissible hit	1/194	11/194	85/194	$+11/-1$	$+84/-10$; shared 1
False specificity ↓	113/113	89/113	3/113	24 rescued/0 new	86 rescued/0 new

The two row structures differ. Commitment discipline moves monotonically: the 24 Principle-Aware false-specificity rescues are a subset of the 110 Exact SOL rescues. Semantic identification does not show the same nesting: the seven Principle-Aware point-exact rows and 54 Exact-SOL point-exact rows have no shared success, while admissible hit has one shared success. We report this as a failure of success-set convergence, not as proof of a distinct latent process. It is inconsistent with the simple explanation that sufficiently strong generic advice gradually recovers the same mapping.

Custody and verification. The prompt SHA-256 is `be54584c345924d39ae50bc674ec1fe38415f9605372d35da64532f8e27214ad`. The returned ZIP SHA-256 is `d06f10144b516eefc7625fb8de55b306d3a0de65c435c52217a7e0de1bb09e40`, and its `PREDICTIONS.jsonl` payload SHA-256 is `df738a5e299c1c4a5f0f8bdee02b24981d5060ae90aec7710882688b4f27e3c1`. All 217 rows pass schema, order, uniqueness, packet-hash, closed-reference, binding, canonicalization, and prediction-hash checks with zero technical failures and `truth_join_executed=false` before scoring. The unique truth join produces a frozen row-score file; a separate scorer-free recount that reads no truth or reference reproduces all three counts and every paired transition. The anonymous V4 artifact releases the immutable prediction ZIP, score bundle, design and analysis freeze, custody receipts, primary report, and independent recount. Its public prompt changes only the machine-local input locator and binds the original frozen prompt hash in a redaction receipt. No numerical success threshold or significance claim was defined, and no prompt revision or repeat run followed output inspection.

D.6 Residual errors and the four executor disagreements

This retrospective description reads the frozen predictions and row scores; it neither reruns the models nor changes their reference or scoring. Internal joint exactness and external semantic exactness use different constructions, output spaces, and eligible populations. Their percentages therefore do not measure an internal-to-external accuracy drop on a shared endpoint.

Among the 194 external rows with a nonempty certified set, Exact misses 109. We assign each miss to its first failed admissible-hit layer, checking disposition, localization, carrier decomposition, then semantic identity. The resulting disjoint groups contain 35, 21, 13, and 40 rows, respectively. Thus 40 misses already recover an accepted carrier but not its fully qualified identity; the remaining misses arise earlier. Separately, 89 of the 109 outputs propose one hypothesis, 19 abstain on presence uncertainty, and one returns INPUTUNJUDGEABLE. The latter is a scientific output status, not a technical intake failure. These counts describe where matching fails; they do not establish long horizons or missing telemetry as its cause.

The four Exact–Strong core-vector disagreements have a common operational form (Table 16). One executor emits an adopted SUBMISSION, while the other returns no object with the reason `PROJECTION_TERMINAL_OBJECT_NOT_CLOSED`. The accepted projection in each case is abstention, so both point exactness and admissible hit favor the abstaining executor. None of these four rows enters false specificity. These are disagreements about whether an object can be completed from the record, rather than evidence that either executor is uniformly more cautious. The outputs identify that decision boundary; they do not reveal the model's internal reasoning or prove a particular parser defect.

Table 16: All four semantic core-vector disagreements. Row suffixes refer to `TEL-CONFIRMATORY-`; event identifiers are observable packet references. In every row, the abstaining executor receives both point-exact and hit credit.

Row suffix	Exact SOL	Strong	Accepted output
0573	Abstain	Submission at s006	Abstain
0695	Submission at s008	Abstain	Abstain
0740	Submission at s016	Abstain	Abstain
0897	Abstain	Submission at s010	Abstain

The descriptive grouping is recoverable from the released Exact `ROW_SCORES.jsonl`; the four row identifiers and metric directions are listed in `audits/SOL_EXACT_VS_STRONG_ROW_ALIGNMENT_AUDIT_V2.json`. Strong's released predictions and the frozen Exact return supply the output statuses, reasons, and event references shown above.

D.7 RESOLUTION AND CORPUS DETAIL

Only after the three main experiments do we expand the analysis across the full four-layer resolution hierarchy. Table 17 shows that the semantic-layer result is not an isolated endpoint.

Table 17: Supporting resolution-depth analysis for Exact SOL versus Strong. Reference status audits denominator invariance; the remaining columns measure cross-implementation agreement.

Resolution layer	Ref. status	Prediction projection	Core-metric vector
Disposition	217/217	208/217	208/217
Localization	217/217	208/217	210/217
Carrier decomposition	217/217	208/217	210/217
Semantic identity	217/217	206/217	**213/217**

The common reference fixes identical eligibility partitions at every layer. Conditional on these shared partitions, cross-implementation core-metric-vector alignment is 208/217 at disposition, 210/217 at localization, 210/217 at carrier decomposition, and 213/217 at semantic identity. Deeper semantic resolution therefore does not erode agreement between the two realizations. The changing denominators and complete four-method score table follow below.

The effect also appears in both external corpora. On AgentRx, Exact SOL and Strong both obtain 4/12 semantic point exactness, 24/65 admissible hit, and 3/59 false-specificity failures; on TELBench both obtain 50/85, 61/129, and 0/54, respectively. Generic SOL is 0/12, 1/65, 59/59 on AgentRx and 0/85, 0/129, 54/54 on TELBench. Thus transport is not created by pooling corpora: the contract changes semantic behavior in each source, while the heterogeneous realizations converge within each source.

Absolute identification remains resolution dependent; the denominators reflect the frozen admissible sets and must not be interpreted as a single accuracy curve. Earlier path-preservation studies, the Mini profile, and traditional deterministic baselines remain useful comparative context, but they support rather than define the central specification–implementation result (Appendices B and D). All four methods have zero technical failures, and independent static recounts report no row-metric or aggregate mismatch.

Table 18: Complete four-method external score table. This supporting table retains Mini and the three coarser resolution layers removed from the main mechanism table. Exactness and hit are successes; false specificity is a failure count (lower is better).

Metric	Mini	Generic SOL	Exact SOL	Strong
Disposition point exact	154/186	113/186	154/186	154/186
Localization point exact	95/134	43/134	96/134	97/134
Carrier point exact	78/130	26/130	85/130	86/130
Semantic point exact	51/97	0/97	**54/97**	**54/97**
Semantic admissible hit	67/194	1/194	**85/194**	**85/194**
Semantic false specificity ↓	105/113	113/113	**3/113**	**3/113**

The four TELBench semantic-vector differences are `TEL-CONFIRMATORY-0573`, `-0897`, `-0695`, and `-0740`. The first two favor Exact SOL and the last two favor Strong on point exactness and admissible hit. This two-to-two exchange yields identical corpus aggregates. Semantic false specificity agrees for all 73 AgentRx and all 144 TELBench rows, including every eligible row.

D.8 Supporting path and baseline comparisons

Earlier path-preservation experiments establish that terminal equivalence does not erase diagnostic structure (Appendix B). The previously frozen Mini realization also provides a descriptive external reference profile: point exactness is 154/186 at disposition, 95/134 at localization, 78/130 at carrier decomposition, and 51/97 at semantic identity; semantic admissible hit is 67/194. Figure 5 reproduces that accepted profile unchanged and places the historical extended-disposition comparison in its supporting role.

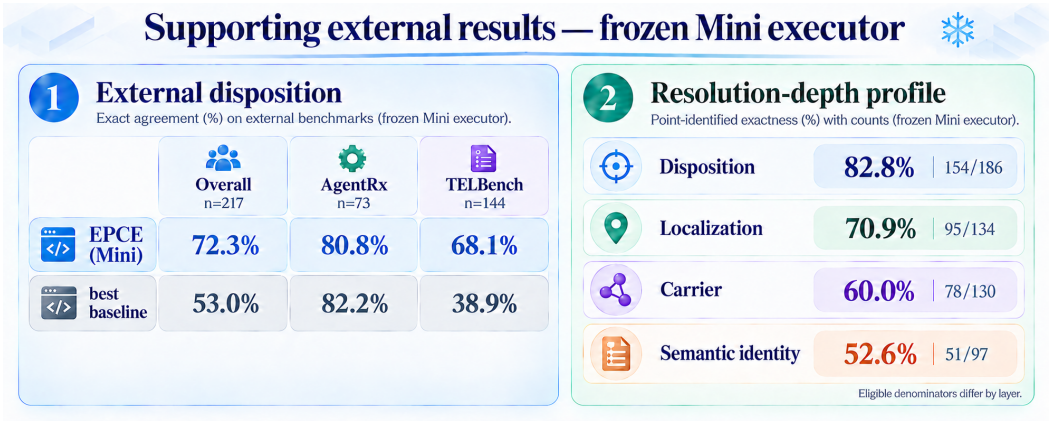


Figure 5: Supporting external profile for the frozen Mini realization. Left: Extended-disposition agreement versus the strongest precommitted deterministic comparator in each column. Right: point exactness across resolution layers; eligible denominators are printed beside each estimate.

Extended disposition is a complementary exact-agreement endpoint over LOCATE, ABSTAIN, and DONOTLOCATE. Against the strongest overall baseline, AgentDebugX event, the paired cluster-bootstrap difference interval is $[12.44, 26.27]$ *pp* and is reported descriptively. No new p-value family was registered for this endpoint. All 217 predictions, including technical and integrity dispositions, remain in the score.

Table 19: Supporting extended-disposition exact agreement for Mini and six precommitted deterministic comparators. This endpoint is not the mechanism isolation result.

Method	Overall	AgentRx	TELBench
Mini	**157/217 (72.35)**	59/73 (80.82)	**98/144 (68.06)**
AgentDebugX event	115/217 (53.00)	**60/73 (82.19)**	55/144 (38.19)
Task-aware PageRank	113/217 (52.07)	**60/73 (82.19)**	53/144 (36.81)
Terminal event only	113/217 (52.07)	**60/73 (82.19)**	53/144 (36.81)
BM25 task event	111/217 (51.15)	**60/73 (82.19)**	51/144 (35.42)
Task-aware MHS	111/217 (51.15)	**60/73 (82.19)**	51/144 (35.42)
Earliest anomaly	109/217 (50.23)	53/73 (72.60)	56/144 (38.89)

The Mini formal rerun is byte-identical. Independent validation finds zero formal row-metric mismatches, zero aggregate mismatches, and zero baseline comparison mismatches. The result packages bind the population, frozen candidates, promoted layer maps, scoring contract, and comparison summaries by manifest and hash.

E Reproduction Artifact Audit Trail

The review artifact is not merely a pointer to unreleased analysis. V5 contains 471 files: `MANIFEST.json` binds 469 payload files by path, byte count, and SHA-256, while `SHA256SUMS.txt` supplies the second root index. This section exposes the artifact's main audit structure in the paper so that a reviewer can assess the released work before opening the archive. The archive remains authoritative; the groupings below are a deterministic display summary of manifest paths.

E.1 Manifest inventory and claim-to-evidence map

Table 20: Logical inventory of all 469 manifest-bound files. Counts are disjoint and exhaustive.

Evidence layer	Files	Released contents and role
Frozen external input, contract, prompt, reference, and licensed-source records	300	Common 217-row observable packets; declarative contract and prompts; prediction-blind reference construction; upstream source and license records.
Frozen external outputs and custody	65	Five immutable score bundles, prediction returns, freeze and truth-join receipts, and the Principle-Aware prospective control record.
Executable closures, scorer, and root verifiers	50	Self-contained Mini and Strong releases, the shared frozen scorer, reproduction entry points, tests, and fail-closed binding checks.
Authorized internal public reconstruction	31	The 122-row public core and failure annex, canonicalizer, predictors, scripts, tests, and internal manifest.
Audits, reproduced summaries, and worked examples	12	Four-method and scorer-free recounts, Exact–Strong row alignment, code-independence and validation reports, plus a readable frozen example.
Coverage provenance and phase commitments	7	V4 predecessor binding, private-source commitments, minimized coverage summary, challenger-removal replay, phase ledger, and build validation.
Entry-point and release-boundary documents	4	V4 and V5 reviewer navigation, paper-support map, and an explicit inclusion/exclusion boundary.
Total	**469**	Every payload file named by `MANIFEST.json`.

The package retains historical internal directory keys where needed for byte custody, but the paper-facing method names are only Mini and Strong. Those keys do not encode an ordinal algorithm comparison.

Table 21: Short paper-to-artifact support ledger. Paths are relative to the V5 archive root.

Paper-facing object	Primary released support
122-row internal public recalculation	`internal_public_core/selection_manifest.json`, `internal_public_core/FAILURE_ANNEX.md`, and release-local scripts and tests.
Matched Generic–Exact treatment	`external_common_input/`, `external_prompts/`, `external_contract/`, and `external_custody/`.
Four frozen 217-row outcomes	`external_score_bundles/`, `audits/FOUR_METHOD_COMPARISON.json`, and `audits/INDEPENDENT_STATIC_RECOUNT_FOUR_METHODS.json`.
Principle-Aware control	`external_principle_aware/`: pre-run freeze, public prompt and redaction receipt, immutable return, primary transitions, and independent recount.
Reference-coverage robustness	`external_reference/truth_construction/candidate_unions_initial/`, `public_provenance_derivative/`, and `v5_metadata/CHALLENGER_REMOVAL_ROBUSTNESS.json`.
Implementation-stability contrast	`code/mini/`, `code/strong/`, `audits/CODE_INDEPENDENCE_REPORT.json`, and `audits/SOL_EXACT_VS_STRONG_ROW_ALIGNMENT_AUDIT_V2.json`.
Row-level readable trace	`examples/WORKED_EXAMPLE.json`, including reference candidates, method projections, row metrics, and projection hashes.
Whole-release integrity	`MANIFEST.json`, `SHA256SUMS.txt`, `PAPER_SUPPORT_MAP.md`, and `scripts/verify_release.py`.

E.2 Freeze-to-recount chain

The public record separates prediction from scoring. In particular, a successful structural reproduction is not treated as a performance result. Table 22 makes the chronology and information boundary explicit.

Table 22: Frozen custody chain for the external comparison. The private truth join is not rerun by the public artifact.

Stage	Bound object	Released evidence	Truth/reference access
1	Observable population and schema	One shared, ordered 217-row packet set and its hashes.	None.
2	Condition-specific instruction	Generic, Exact, and Principle-Aware prompt/configuration custody; Strong's release-local interface is frozen.	None.
3	Returned prediction artifact	Immutable returns, prediction payload digests, order/uniqueness and packet-hash checks.	None.
4	Structural intake	Closed-schema, binding, canonicalization, and projection-hash validation; the receipt records `truth_join_executed=false`.	None.
5	One-time score join	Frozen reference and common scorer are applied only after prediction freeze; receipts bind scorer source and outputs.	Activated once after freeze.
6	Immutable score bundles	Row scores, summaries, manifests, and four-method comparison are released for recalculation.	No new prediction access.
7	Independent static recount	Direct aggregation reproduces all reported overall and corpus metrics without importing the formal scorer.	No private truth or reference input.

The two programmatic releases make the fixed-specification boundary executable. The code audit finds 8 byte-identical frozen-kernel files, zero cross-root imports, zero cross-root runtime filesystem reads, zero forbidden imports, and zero entry-point `sys.path` mutations. Mini and Strong each reproduce 217/217 and 217/217 truth-blind rows, respectively, with zero mismatches. Neither release reproduction reads truth, reference, score, or label inputs; row-identity branching and scored-outcome tuning are not authorized. These checks establish a shared frozen semantic substrate plus release-local operational execution. They do *not* claim source-byte disjointness or clean-room invention of the semantic kernel.

The resulting stability claim is correspondingly narrow and auditable: Exact SOL and Strong agree on 213/217 semantic core-metric vectors and 113/113 eligible false-specificity decisions. The four semantic-vector disagreements are retained row by row in the alignment audit rather than hidden by equal aggregate scores.

E.3 FAIL-CLOSED VERIFICATION LEDGER

The entry point `scripts/verify_v5_release.py` runs `scripts/verify_release.py` (the 14 inherited V4 checks below) and `scripts/verify_v5_coverage.py` (12 V5 coverage, join, timing, anonymity, and robustness checks), for 26 gates in total. Running it against a fresh extraction of the exact V5 archive used for this candidate returned `PASS` throughout:

Table 23: The fourteen checks executed by `scripts/verify_release.py`.

No.	Gate	Fail-closed target	V5
1	Manifest	Payload paths, byte counts, SHA-256 values, and release inventory.	PASS
2	Scorer-source binding	Released scorer-source bytes against all frozen truth-join/source receipts.	PASS
3	Count reproduction	Four-method table, Generic–Exact transitions, and Exact–Strong row alignment.	PASS
4	Principle-Aware audit	Pre-run freeze, custody, endpoints, paired transitions, and scorer-free recount.	PASS
5	Mini reproduction	Truth-blind Mini reproduction on all 217 observable packets.	PASS
6	Mini tests	Mini release-local unit suite.	PASS
7	Mini score binding	217/217 projection-hash agreement at each of four resolution layers.	PASS
8	Strong reproduction	Truth-blind Strong reproduction on all 217 observable packets.	PASS
9	Strong tests	Strong unit and metamorphic suites.	PASS
10	Scorer tests	Frozen shared scorer unit suite.	PASS
11	Internal manifest	Internal public-core selection, files, and hashes.	PASS
12	Internal EPCE	All 122 released deterministic EPCE prediction records.	PASS
13	Internal TA	All 122 released TA-MHS and TA-EA prediction records.	PASS
14	Internal tests	Internal canonicalizer, predictors, scorers, and guards.	PASS

This ledger proves artifact integrity, executable reconstruction, and stated cross-file bindings. It does not prove that a semantic contract is uniquely correct, regenerate excluded private provider responses, or authorize a new scientific claim beyond the frozen analyses.

E.4 WORKED ROW: VISIBLE TERMINAL EVENT, MULTIPLY ADMISSIBLE SEMANTICS

The frozen worked example is row `ARX-MAG-08cae58d-4084-4616-b6dd-dd6534e4825b`. Its task asks for the first year Apple stock exceeded $50 on an unadjusted series, while the observable receipts repeatedly expose adjusted data. The terminal event `e0129` nevertheless emits "FINAL ANSWER: 2007." The trace is paraphrased because raw trajectory content is outside the release boundary, but the candidate and score records below are exact.

Table 24: Frozen reference candidates for the worked row. Votes are admissible/inadmissible/unresolved.

Candidate	Semantic domain	Adoption state	Terminal role	Votes
C001	Submission	Adopted	Submission and terminal	3/0/0
C002	Task claim	Adopted	Terminal only	2/1/0

The reference is therefore `CERTIFIED_MULTIPLY_ADMISSIBLE`. Merely finding the visible terminal event does not determine whether a unique semantic claim is licensed.

Exact SOL and Strong use different uncertainty labels but bind the same frozen projection hash, `76e204d58c21045da2956a99626f8d9d96edfa337bd5ce498b33005e9e037d34`. The example instantiates the paper's identification claim at row level: observation fixes what happened, whereas the declared contract determines which semantic object and ambiguity set are legal.

Table 25: Method-level projections on the same worked row. False specificity is a failure indicator.

Method	Predicted adoption	Registered uncertainty	Admissible hit	False specificity
Generic SOL	Not observable	Unique	0	1
Exact SOL	Adopted	Set or ambiguity	1	0
Strong	Adopted	Multiple	1	0

F Extended Discussion and Literature Context

The main text argues that diagnostic rules belong in the evaluation design. This appendix places that argument beside related work and separates what the present tests establish from questions that would require another study.

F.1 Expanded related-work positioning

Contracts and realizations. Formal methods separate specification from realization (Wing, 1990; Liskov & Wing, 1994; Meyer, 1992), while evaluation work makes scenarios, metrics, and artifacts explicit (Liang et al., 2023; Biderman et al., 2024). Most directly, AsymmetryZero encodes expert requirements as stable semantic-evaluation contracts and compares judge realizations under a fixed contract (Looram et al., 2026). EPCE takes the next step for diagnosis: how does a contract identify the scored object?

Underspecification and measurement. Underspecification in ML pipelines shows that one training-and-validation specification can admit predictors with materially different deployment behavior (D'Amour et al., 2022). Measurement modeling and construct validity ask whether operationalizations preserve the intended construct rather than merely produce a repeatable number (Jacobs & Wallach, 2021). These concerns are complementary to ours: we hold the realized trajectory fixed and vary neither a trained model realization nor the observable record. The missing variable is the semantic identity of the diagnostic object that the scorer treats as measured.

The immediate empirical neighborhood. MP-Bench admits multiple plausible multi-agent failure attributions and scores multi-perspective answers (In et al., 2026). Ambig-DS pairs specified and ambiguous data-science tasks, exposing silent commitment to unintended targets or objectives (Stoisser et al., 2026). TelemetrySuffBench separates detection, origin localization, and safe abstention, showing that telemetry can reveal a failure without licensing its origin (Zhu & Pu, 2026). These works establish attribution plurality, task-framing underspecification, and evidence insufficiency. They leave open whether the scored diagnostic object is identified by the visible record or by a declared specification. EPCE isolates that axis through changed-contract/fixed-implementation and fixed-contract/changed-implementation contrasts.

Trajectory benchmarks, ambiguity, and interventions. Agent trajectory benchmarks evaluate behavior, interaction, steps, dependencies, spans, and claims (Liu et al., 2024; Ma et al., 2024; Yao et al., 2025; He et al., 2025; Barke et al., 2026; Wang et al., 2026a). Partial identification and set-valued prediction retain compatible sets rather than force a singleton (Tamer, 2010; Sadinle et al., 2019; Angelopoulos & Bates, 2023); contrast sets, CheckList, and metamorphic testing probe semantics under controlled changes (Gardner et al., 2020; Ribeiro et al., 2020; Chen et al., 2018). EPCE combines these concerns: it preserves contract-licensed ambiguity and uses frozen semantic-role swaps as a mechanism check without inspecting private reasoning tokens.

Benchmark and analytic-choice sensitivity. The Benchmark Lottery shows that changing benchmark tasks can change method rankings (Dehghani et al., 2021); multiverse analysis makes the dependence of conclusions on defensible analytic choices explicit (Steegen et al., 2016). Our secondary contract-view analysis is a small diagnostic analogue, not a full multiverse: on the 38-row intersection where both views have nonempty admissible sets, it changes only which visible semantic object is admissible and obtains a different method ordering. Prior work primarily varies model realization, pipeline, task, metric, benchmark, or analysis path; EPCE isolates specification dependence of the scored diagnostic object under a fixed fully visible trace.

F.2 EXPANDED IMPLICATIONS AND LIMITATIONS

The rules define what the diagnosis names. The experiments explain more than an overall gain from a detailed prompt. Providing C changes the same model's answers on internal and external trajectories. Exchanging specific roles changes the predicted parts of the answer. Under the unchanged C, a program and a model largely agree on individual scoring outcomes. Together, these tests separate the diagnostic rules from their execution. Principle-Aware adds a further distinction: advice about what to consider can improve caution while leaving most of the exact contract's gains unrecovered.

EPCE is one declared object, not a claim to the uniquely natural ontology. Section 2 makes the consequence concrete rather than leaving it implicit: an outcome-centered and a process-centered question change admissible families on 113 rows and reorder methods on their common support. Other defensible contracts therefore strengthen the requirement to specify the estimand before systems are compared.

Three questions for separately designed studies. *Can another team reconstruct the rules from their description?* Our implementation comparison keeps the semantic kernel K identical and changes its executor (Equation 6). A team reconstructing K' from prose would test a different property: whether the description communicates the same rules. Comparing f_{K,I_1} with f_{K',I_2} changes both the reconstructed rules and their execution. This would complement the present fixed-rule comparison.

Would independent experts choose these diagnostic objects? Experts working without the EPCE canonicalization could propose their own candidates. Such a study would need to state who the experts are, what they are asked, when two answers count as equivalent, and how answers are combined. Overlap with $\mathcal{H}_C(G, q)$ could then measure agreement with the chosen objects. The model reviewers in Table 12 instead judge admissibility under an already-frozen contract. These are different forms of validation.

The reference roles and the evaluated SOL conditions use the same model family. Prediction blinding prevents direct adaptation to submitted answers, but does not remove correlated omissions or interpretive preferences across those roles. Strong's agreement supports reproducible execution of the fixed rules; it does not independently establish that the model-assisted reference is free of such shared bias.

How often do different diagnostic questions change rankings? The current analysis changes acceptable sets on 113 of the 217 frozen cases and changes the method order on 38 shared cases (Section 2). It establishes an occurrence. A preregistered study on fresh cases could estimate how often this happens and whether the result extends to other populations.

How the two tests support each other. On external cases, changing the rules given to the same model repairs 54 point-exact outcomes and 110 false-specificity errors without regressions on those two scores. Changing the executor under fixed rules produces agreement on all three scoring outcomes for 213/217 cases. The internal role swaps locate the response more precisely: among Exact-correct cases, exactly the predicted cases change, while a separate all-case check preserves the unmanipulated projections. These results support the operational role of the declared semantics. Principle-Aware tests whether general advice recovers the same effect; in the frozen comparison, it does not recover the same set of point-exact successes.

The commitment boundary matters. False specificity asks whether a unique answer is justified, separately from ordinary accuracy. Exact SOL and Strong avoid the same 110 errors and make the same three. Principle-Aware repairs 24 errors without new failures; Exact SOL preserves those repairs and adds 86. The rules treat ambiguity as an allowed answer. This connects diagnosis to partial identification and selective prediction, with a distinct target: semantic admissibility rather than statistical confidence.

Three principles for evaluation design. Define the target before ranking systems; preserve acceptable alternatives; compare executors of the same rules. A benchmark can implement these principles by publishing its observable schema, identity rules, allowed outputs, and scorer. Our 213/217 agreement on scoring vectors and 113/113 on eligible false-specificity decisions demonstrate the execution check without requiring private chain of thought.

Scope and limitations. Test I compares two executors under one frozen contract on 217 trajectories from two sources. It does not prove agreement for arbitrary future implementations. Eleven semantic prediction projections differ, and four cases have different core scoring vectors. Further executors would need their allowed input adaptation, access to other implementations, and freeze point stated in advance. The role swaps test specified semantic changes; they are not a randomized factorial experiment on every contract clause. EPCE evaluates declared observations and task requirements, not hidden reasoning or an unobserved causal root.

The additional questions above require their own populations and protocols. Principle-Aware SOL is one prospectively frozen stochastic run, so it cannot establish variation across runs or expected performance. Freezing the run without later revision prevents selection of prompts or runs after seeing the outcomes. Its zero overlap with Exact SOL's point-exact successes is a fact about these outputs, not evidence of different hidden reasoning mechanisms.

G Public Recalculation and Release Boundary

The internal public core is selected by a deterministic post-result hash rule over truth-status strata: 60 localizable, 30 no-error, and 30 other nonlocalizable rows. All three deterministic EPCE mismatches are included in a separate annex; one overlaps the core, yielding 122 unique rows. Starting from the released truth-free projection, the public canonicalizer and predictor match all 122 frozen EPCE, TA-MHS, and TA-EA prediction records.

The anonymous package excludes the full private 720-row truth, complete internal SOL-A/SOL-B returns, raw provider responses and reasoning traces, credentials, endpoint secrets, HMAC material, and protected task documents. External sources are identified through their official releases and frozen acquisition receipts; redistribution follows their upstream licenses and venue policy.

H Required Statements

AI use. Generative AI was used both as an experimental system and in supporting research workflows. AI-generated predictions constitute research data and were generated only under the frozen protocols described in the manuscript and review artifact. Separately, generative AI assisted protocol development, implementation and testing, exploratory analysis, figure preparation, literature organization, and drafting. All research decisions, experimental protocols, reported analyses, citations, and claims were reviewed and validated by the authors. Generative AI was not used to modify frozen outputs after inspection or to fabricate or substitute verification of references. The authors take responsibility for the complete contents of the work; no AI system is an author.

Ethics. Controlled modifications remain in local corpora and sandboxes and do not alter public sites, indexes, or services. Unsanitized truth, credentials, protected data, and operational attack strings are excluded from the review artifact.

Reproducibility. The internal artifact regenerates 122 released prediction records from truth-free sanitized projections. SOL materials bind exact prompts, common input, returned ZIP digests, prediction freeze, and the shared scorer. The external artifact retains four immutable 217-row primary score bundles and the Principle-Aware robustness-control bundle, their SHA-256 manifests, structural and prediction freezes, one-time truth-join receipts, and the common frozen scorer. It also releases the frozen Principle-Aware prediction ZIP, prompt and redaction receipt, prospective analysis plan, paired-transition report, and scorer-free recount. The Mini formal rerun is byte-identical; independent static recount and Exact–Strong row alignment checks report zero mismatches. V5 preserves 439 V4 payload files byte-for-byte and separately binds three documentation derivatives with normalized publication prose; scientific content and original freeze receipts are unchanged. It retains V4's scorer-source repair and adds the minimized reference-provenance derivative, challenger-removal replay, phase ledger, and verification. Its combined verifier passes 14 inherited and 12 V5 checks on 469 manifest-bound files; the ZIP SHA-256 is `77f3f0536f694ffafa746cb6af0bae2052c51b48021ce71fafe35f38a6fbae5b`.

Data and code availability. The review artifact includes the authorized internal public core, failure annex, prompts, custody receipt, canonicalizer, deterministic predictor, scorers, and manifest checks. AgentRx and TELBench acquisition is documented against their official releases. External releases include or bind the frozen initial and final candidate projections, minimized source/adjudication provenance, phase commitments, five score-bundle manifests, scorer, audit receipts, and the Principle-Aware robustness-control materials needed for static recount. Private internal trajectories, complete provider outputs, and protected mappings remain outside the anonymous package.