WHEN CAN AN UNDERSPECIFIED LANGUAGE AGENT ACT? CERTAIN-ACTION VERIFICATION VIA OUTCOME INVARIANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Tool-using language agents must decide whether to act, clarify, or declare a request non-actionable under underspecification. Existing confidence and clarification methods focus on recovering a likely intent, but action safety depends on whether feasible interpretations induce different policy-relevant outcomes. We introduce **CAVI**, a contract-relative framework that represents interface-feasible task completions and certifies action only when all represented completions collapse to one outcome class. Otherwise, it returns outcome-distinguishing counterexamples or a machine-checkable non-actionability certificate. Under conservative completion coverage, faithful outcome abstraction, exact verification, and transaction-stable execution, CAVI certificates are sound for atomic or transaction-scoped actions. At the verifier level, across 120,000 requests in 12 domain-distribution cells, CAVI reaches the matched zero-false-action frontier while certifying 36.42% of requests; independent enumerative and symbolic implementations agree on all labels, and 1.2 million metamorphic checks produce no label failure. A frozen 2,160-call cross-domain parser confirmation yields 0 false actions and 0 false certificates among 533 certificates while certifying 24.68% of calls, and exposes high completion under-approximation and low clarification recall as the remaining parser boundary. We then test the same decision principle end-to-end: in a blind executable validation on 105 held-out tasks in local state-mutating application sandboxes, CAVI operates as a consequential-action gate inside an LLM agent loop, reducing the false action rate to 1/105 versus 33/105 for direct execution while cutting unnecessary clarification to 0.019 versus 0.571 under conservative always-clarify interaction. Conservative completion coverage remains an important boundary; a mechanical semantic audit found one proposal–task mismatch among 83 CERTAIN_ACTION certificates, exposing proposal faithfulness as an additional agent-integration boundary. CAVI thus converts model-proposed task hypotheses into conditional, auditable action guarantees rather than relying on confidence alone.

1 INTRODUCTION

A language agent that answers ambiguously may be annoying; a language agent that acts ambiguously may be dangerous. Acting too early is unsafe; clarifying too often is brittle; refusing too broadly is unhelpful. Consider "cancel my cheapest order": with a single cancellable cheapest order, asking only adds friction; with two tied orders differing in refund channel or shipment consequence, acting without clarification is unsafe; if the order is invisible through the interface or policy forbids cancellation, the right response is non-actionability. Ambiguity and safety are therefore not the same axis: the central issue is whether ambiguity can change the side-effecting outcome.

Most existing approaches treat underspecification as uncertainty over a latent intent and decide whether confidence warrants acting or further interaction. For a side-effecting agent, however, the relevant object is not uncertainty over token sequences—or even semantic interpretations—but the set of environment outcomes those interpretations induce. A low-confidence request can be safe when this set is a singleton, whereas a high-confidence interpretation can be unsafe when a fea-

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.

sible alternative changes refund, notification, authorization, or reversibility. CAVI therefore turns ambiguity resolution into instance-level outcome verification.

We introduce **CAVI**, a contract-relative certification layer for atomic or transaction-scoped language-agent actions. Given a request x, snapshot e, interface policy π, and outcome abstraction Ω, CAVI constructs a conservative completion space $\widehat{Z}(x, e, \pi)$ and certifies action only when all represented completions collapse to one Ω-equivalence class:

$$|\mathcal{O}_\Omega(\widehat{Z})| = 1,$$

where

$$\mathcal{O}_\Omega(\widehat{Z}) \;=\; \big\{\, [o]_\Omega \;:\; z \in \widehat{Z}(x, e, \pi),\; o \in F(z, e, \pi) \,\big\}.$$

Otherwise it returns clarification counterexamples or a non-actionable-under-interface certificate. The guarantee is explicitly contract-relative: conservative completion coverage, faithful outcome abstraction, exact verification, and transaction-stable execution are part of the certificate boundary.

Viewed as selective action, CAVI abstains not because the model is uncertain, but because the verified outcome set is non-singleton or cannot be conservatively represented.

Crucially, CAVI compares outcomes rather than surface actions. Two tool calls may differ syntactically while producing the same policy-relevant state change, refund channel, notification, and reversibility; conversely, similar-looking calls can differ in downstream consequences. The main design challenge is therefore not only to enumerate possible completions, but to decide which execution differences matter. We capture this policy-authored comparison lens as the outcome abstraction Ω, and make its version, assumptions, and transaction preconditions part of the certificate.

Our contributions are fourfold. First, we formulate certain-action verification as outcome invariance over conservative interface-feasible completions and prove conditional soundness for atomic or transaction-scoped actions. Second, we introduce a verifier-backed architecture with policy-authored outcome abstractions, deterministic interface grounding, and machine-checkable action, clarification, and non-actionability certificates. Third, we introduce Certain-Action Arena and evaluate CAVI through matched-risk baselines, exhaustive dual-implementation and metamorphic integrity audits, and a frozen cross-domain parser confirmation. Fourth, we validate the decision principle end-to-end by deploying CAVI as a consequential-action gate inside an LLM agent loop over local state-mutating application sandboxes under a frozen, blind, held-out executable validation protocol. Throughout, we are explicit that end-to-end failures remain possible whenever the representation supplied to the verifier fails to cover the true completions; the executable evaluation is designed to expose, not hide, that boundary.

2 Related Work

Uncertainty, abstention, and clarification. Semantic-uncertainty methods group surface generations by meaning to estimate whether a model knows an answer, while selective systems abstain or seek interaction when uncertainty is high Kuhn et al. (2023); clarification and mixed-initiative methods ask when additional user information is worth its cost Horvitz (1999); Yao et al. (2019); Kuhn et al. (2025). CAVI addresses a different, decision-level object: whether all interface-feasible interpretations induce the same policy-relevant state transition; model confidence or semantic agreement is therefore neither necessary nor sufficient for certification.

Tool-using agents and safety evaluation. AgentBench, MINT, ToolLLM, WebArena, and τ-bench evaluate interactive tool use, while ToolEmu, AgentDojo, and AgentHarm expose accidental, injected, or adversarial harms Liu et al. (2024); Wang et al. (2024); Qin et al. (2024); Zhou et al. (2024); Yao et al. (2025); Ruan et al. (2024); Debenedetti et al. (2024); Andriushchenko et al. (2025). Certain-Action Arena instead isolates when underspecification is harmless for a state-changing action and compares systems on a matched risk–coverage frontier.

Verification under incomplete information. Certain-answer semantics accepts results invariant across completions of incomplete data Imieliński & Lipski (1984); Libkin (2011); conformant planning and runtime enforcement reason about goal achievement

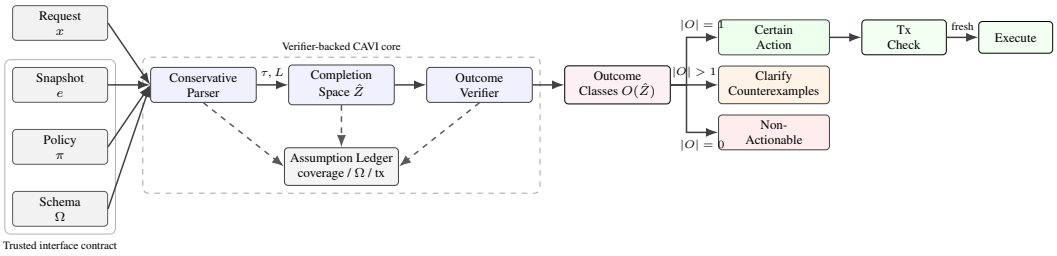

Figure 1: Contract-relative certain-action certification pipeline. The model proposes a conservative task specification, but only the verifier certifies whether all represented completions collapse to one Ω-outcome class. Coverage or parser failure routes to clarification; non-actionability is issued only after the verifier establishes zero executable represented completions. A certain-action certificate authorizes execution only after transaction freshness checks.

or policy compliance under uncertainty Palacios & Geffner (2009); Wang et al. (2025). CAVI applies this invariance principle to side effects under an explicit interface contract; parser coverage, proposal faithfulness, and outcome-schema faithfulness remain explicit assumptions.

Formal policy enforcement and verified tool use. Recent systems bring formal constraints into agent execution: AgentSpec supplies a runtime-enforcement DSL; Agent-C enforces temporal/state constraints over tool calls; solver-aided checking translates tool-use policies to SMT and checks calls with Z3; and STPA-based work derives tool-sequence and data-flow constraints Wang et al. (2025); Kamath et al. (2025); Winston et al. (2026); Doshi et al. (2026). These approaches determine whether a proposed action or execution trace satisfies a prescribed specification. CAVI addresses a complementary problem: whether all interface-feasible completions of an underspecified request induce the same policy-relevant outcome. Contextual-security formalisms characterize task/action alignment, source authorization, and data isolation Siu et al. (2026); task/action alignment is closely related to CAVI's Proposal Faithfulness boundary. CAVI treats request-to-proposal alignment and policy enforcement as trusted-boundary components; conditional on the represented specification, its core verifier asks whether residual request underspecification can change the reachable policy-relevant outcome class. Where runtime-enforcement methods verify compliance of a proposed action or trace, CAVI verifies outcome invariance across feasible request completions.

3 METHOD

CAVI separates parsing from certification: the parser must conservatively cover outcome-relevant latent slots, while only the verifier certifies under the explicit interface contract. Coverage failures route to clarification; non-actionability is issued only when the verifier establishes that no represented interface-feasible completion is executable (Figure 1).

Procedure. CAVI parses x into τ, L, builds $\hat{Z}$, executes each feasible $z \in \hat{Z}$ through F, projects outcomes through Ω, and stops early when two distinct classes provide clarification counterexamples. Zero executable completions yield a non-actionable certificate; one class yields a certain-action certificate plus transaction preconditions.

We define the *projected outcome class set* as

$$\mathcal{O}_\Omega(\widehat{Z}) \;=\; \big\{\, [o]_\Omega \;:\; z \in \widehat{Z}(x, e, \pi),\; o \in F(z, e, \pi) \,\big\}.$$

Certification requires $|\mathcal{O}_\Omega(\widehat{Z})| = 1$. In the executable agent integration (Section 6.1), the verifier receives a model-proposed consequential action specification; proposal faithfulness—the consistency of that specification with the request-derived task representation—is therefore part of the integration boundary (Table 1).

Problem Setup. Let x be a request, e an environment snapshot, and π an interface-policy specification defining visible objects, actions, preconditions, and policy rules. CAVI does not model

the open world; it reasons over latent task completions z induced by x under e, π—in the running order example, filling missing variables such as the intended order and tie-breaking convention. The feasible completion space is

$$Z(x, e, \pi) = \{z : z \text{ consistent with } x, e, \pi\}.$$

Executing a completion induces a structured outcome $F(z, e, \pi)$, which is *set-valued*: it contains every outcome reachable when completion z is executed (API failures, races, policy nondeterminism), so $F(z, e, \pi) \subseteq \mathcal{O}$. A policy-authored outcome abstraction Ω defines equivalence: $o_1 \equiv_\Omega o_2$ iff the two outcomes agree on all fields in Ω. For the order example, Ω may retain `target_order_id`, `order_state`, `refund_channel`, `shipment_stopped`, `notification`, and `irreversible_effect`. Object identity is part of the default outcome abstraction for side-effecting operations unless the interface policy explicitly declares the affected objects interchangeable.

We define the projected outcome class set $\mathcal{O}_\Omega(Z)$ as

$$\mathcal{O}_\Omega(Z) \;=\; \big\{\, [o]_\Omega \;:\; z \in Z,\; o \in F(z, e, \pi) \,\big\}.$$

CAVI certifies a request iff $|\mathcal{O}_\Omega(Z)| = 1$, i.e., all represented completions collapse to one outcome class. This property is interface-relative: changing the snapshot, policy version, or outcome abstraction can change the decision.

Conservative Completion Spaces. The parser maps x into a task DSL τ and assumption ledger L, including intent, latent variables, constraints, preconditions, and interface boundaries. It is not trusted to certify safety. Its contract is conservative coverage:

$$Z_{\text{true}}(x, e, \pi) \subseteq \widehat{Z}(x, e, \pi).$$

The completion builder converts τ into symbolic constraints over task-relevant variables such as order identifiers, refund channels, calendar events, target slots, table filters, or tie-breaking rules. A domain-authored latent-variable registry (e.g., `order_id`, `price_tie_rule`, `refund_channel` for orders; `event_id`, `time_window`, recurrence scope for calendars) prevents the parser from silently dropping outcome-relevant slots. If a request mentions an uncertain or out-of-registry slot, CAVI expands $\hat{Z}$ to include all interface-feasible values when the slot is representable. If user information could make an action executable or unique, it fails closed to clarification; if no represented completion is executable under the interface policy, it returns non-actionable; if the parser cannot represent an outcome-relevant latent variable, it clarifies rather than certifies. Parser under-approximation is therefore a safety violation, not merely a parsing error; the verifier can certify only the represented space it receives.

Designing the Outcome Abstraction. The outcome abstraction Ω is a policy-authored equivalence relation over concrete outcomes—not a convenience field list, but part of the safety boundary. Practitioners author Ω by enumerating side-effect fields (object identity by default, monetary effects, notifications, reversibility, authorization, audit-visible mutations, downstream preconditions) and running harmful-pair mutation tests; before comparison and hashing, Ω canonically serializes semantically unordered collections using policy-defined identity keys. In our domains, Ω retains order identity, refund channel, shipment stop, attendee notification, recurrence scope, row identity, and audit-visible mutations; field removal requires a policy waiver audited by schema mutation tests. A schema is faithful when $o_1 \equiv_\Omega o_2 \Rightarrow o_1 \equiv_{\text{safe}} o_2$; over-refinement reduces coverage, while under-refinement can create false certificates. Appendix A gives the full schema protocol.

Verification and Certificates. The verifier supports deterministic simulators as a special case of nondeterministic interface semantics. In the general form, execution maps a completion to a set of possible outcomes,

$$F(z, e, \pi) \subseteq \mathcal{O},$$

where the set may represent API failures, races, policy uncertainty, or external nondeterminism. A completion is certain-actionable only if all possible outcomes for all represented completions collapse to one Ω class:

$$\begin{aligned} &\forall z_1, z_2 \in \widehat{Z}, \\ &\forall o_1 \in F(z_1, e, \pi),\ \forall o_2 \in F(z_2, e, \pi) : \\ &\qquad o_1 \equiv_\Omega o_2. \end{aligned}$$

Table 1: Trusted boundary for contract-relative certification. CAVI verifies outcomes under these assumptions and audits the failure modes; it does not claim open-world deployment safety when they are absent.

Assumption	Meaning	If Violated	Audit
Closed interface	Only visible objects/actions are in scope	Hidden feasible actions	unverifiable cases
Conservative parser	$Z_{\text{true}} \subseteq \widehat{Z}$	false certificate	omission/fail-closed
Faithful Ω	no harmful collapse	unsafe equivalence	field-drop pairs
Transaction stability	snapshot/policy/schema unchanged	stale certificate	stepwise recheck
Modeled nondeterminism	all outcome branches represented	ignored branch	set-valued F check
Proposal faithfulness†	proposal matches request-derived spec	task-inconsistent action certified	semantic action replay

Proposal Faithfulness (†). Agent-integration assumption connecting the proposal-derived implementation to the request-derived theorem statement. The proposed consequential action specification must implement an operation/target/constraints semantically consistent with the request-derived task specification.
This condition concerns alignment of the agent's proposal with the user intent expressed in x; it is not an algebraic premise of Theorem 1 but is required for end-to-end correctness when deploying CAVI inside an agent loop (see Section 6.1). Violation produces task-inconsistent actions rather than false certificates per Proposition 1.

The deterministic simulator used for controlled evaluation is the singleton case.

The explicit verifier computes projected outcome classes

$$O(\widehat{Z}) = \{[o]_\Omega : z \in \widehat{Z},\ o \in F(z, e, \pi)\}.$$

If $|O(\widehat{Z})| = 0$, no feasible completion is executable and CAVI returns a non-actionable-under-interface certificate. If $|O(\widehat{Z})| = 1$, all represented completions collapse to one outcome class and CAVI returns a certain-action certificate. If $|O(\widehat{Z})| > 1$, CAVI returns counterexample completions from distinct outcome classes and asks about the outcome-relevant difference. Certificates are machine-checkable objects binding schema versions, canonical outcome hashes, snapshot and policy hashes, completion-space hash, registry version, transaction preconditions, and the assumption ledger (Appendix A); natural language is only a rendering. Certain-action certificates authorize execution only after a transaction check confirms that the bound snapshot, policy, schema, and action preconditions remain valid; otherwise CAVI aborts or re-verifies.

Trusted boundary. Table 1 summarizes the assumptions under which certificates are valid: if parser coverage, schema faithfulness, transaction stability, or modeled nondeterminism cannot be established, CAVI clarifies rather than certifying action; non-actionability requires a valid represented space with zero executable completions.

4 Theoretical Guarantees

Theorem 1 (Certain-action soundness). Fix a request x, snapshot e, interface policy π, and outcome abstraction Ω. The theorem is stated over the request-derived completion space $Z_{\text{true}}(x, e, \pi)$ and represented space $\widehat{Z}(x, e, \pi)$. Suppose that: (i) *conservative completion coverage* holds, $Z_{\text{true}}(x, e, \pi) \subseteq \widehat{Z}(x, e, \pi)$; (ii) Ω is faithful to the domain safety relation, so $o_1 \equiv_\Omega o_2 \Rightarrow o_1 \equiv_{\text{safe}} o_2$; (iii) the verifier exactly computes the projected outcome classes

$$O(\widehat{Z}) = \{[o]_\Omega : z \in \widehat{Z}(x, e, \pi),\ o \in F(z, e, \pi)\};$$

and (iv) the snapshot hash, policy version, schema version, and action preconditions remain stable until execution. If CAVI returns a certain-action certificate, i.e., $|O(\widehat{Z})| = 1$, then every true feasible completion $z \in Z_{\text{true}}(x, e, \pi)$ and every reachable outcome $o \in F(z, e, \pi)$ belongs to the certified Ω-class. Consequently, all true feasible executions are equivalent under the domain safety relation.

(The theorem quantifies over request-derived spaces; applying its guarantee to the proposal-derived executable agent pipeline additionally requires proposal faithfulness as described in Table 1.)

Proof sketch. By conservative coverage, every true feasible completion lies in the represented space checked by the verifier; by exact verification and $|O(\widehat{Z})| = 1$, all reachable outcomes project to the same Ω-class, and faithfulness lifts this to the domain safety relation. Transaction stability ensures the checked snapshot, policy, schema, and preconditions still hold at execution, so executing under

the certificate cannot select a true completion whose policy-relevant side effect differs from the certified class. The full formal statement and proof appear in Appendix A.2.

Proposition 1 (Trusted-boundary necessity). The assumptions in Theorem 1 are not merely implementation details. If conservative coverage fails, there may exist a true feasible completion $z^\star \in Z_{\mathrm{true}} \setminus \widehat{Z}$ whose outcome differs from the certified class, yielding a false certificate. If Ω is unfaithful, two safety-distinct outcomes may collapse into the same projected class, again allowing an unsafe certificate. Therefore, parser coverage and schema faithfulness are part of the certificate contract; when they cannot be established, CAVI must clarify rather than certify. Non-actionability is reserved for a valid represented space with no executable completion. This proposition is central to our experiments: parser, schema, transaction, and scaling audits test the boundary conditions under which certificates are meaningful. Full constructive proofs appear in Appendix 10 and Appendix 11.

Complexity. Under representation assumptions—polynomial-size encodings for completions and outcomes; polynomial-time decidability of feasibility $C(z)$ and membership $o \in F(z)$; polynomial computability and comparability of projections $P_\Omega(o)$—CERTAIN-ACTION is in coNP (Appendix A). It is coNP-hard for succinct DSLs capable of encoding Boolean assignments as completions and polynomial-time Boolean predicates into the execution semantics; this is a worst-case statement over expressive DSLs and does not assert hardness for every CAVI domain language.

5 CERTAIN-ACTION ARENA

Certain-Action Arena is generated from explicit interfaces rather than surface ambiguity alone. Each instance contains a request, snapshot, interface-policy specification, oracle completion space, outcome schema, and gold decision. Labels are computed from outcome classes over Z_{true}: one class yields certain-actionable, multiple classes yield clarification-needed, and no executable completion yields non-actionable-under-interface. The benchmark covers order/refund, calendar, and restricted table-action domains (side-effect inventory: Table 5 in Appendix A.6) crossed with balanced, realistic, hard, and boundary distributions, yielding 12 cells with 10,000 labeled instances each. Examples include tied cheapest orders with different refund channels, ambiguous recurring-meeting cancellations with different notification consequences, and table updates whose filters select audit-distinct rows.

Generation proceeds from the interface contract: for each instance we sample a snapshot and policy variant, choose a latent completion family, render a request with selected slots omitted or constrained, enumerate Z_{true}, execute every feasible completion, and project outcomes through the full schema; the four distributions respectively weight canonical categories evenly, favor fully specified and harmlessly underspecified requests, increase outcome-relevant ties and conflicts, and stress non-actionability and contract failures (Appendix A.6).

6 EXPERIMENTS

Protocol and Evidence Levels. We separate five evidence levels and do not pool them: (1) *controlled verifier evidence*, where oracle completion spaces test outcome-invariance decisions on Certain-Action Arena; (2) *integrity evidence*, where independent label implementations and metamorphic audits test implementation stability; (3) *frozen parser evidence*, where model-generated completion spaces test the conservative-coverage contract and expose the parser boundary; (4) *executable agent validation*, where CAVI gates consequential actions in an LLM agent loop over local state-mutating sandboxes, testing clarification, non-actionability, and transaction drift; and (5) *trusted-boundary stress*, where controlled violations of parser, schema, transaction, and scaling assumptions test where certificates stop being meaningful.

Domains and methods. We evaluate the three Arena domains—order/refund, calendar, and restricted table-action—each with a deterministic simulator, interface policy, latent-variable registry, and versioned outcome schema Ω; the main evaluation crosses three domains with four request distributions (12 cells, 10,000 instances each). CAVI is compared with safe clarification policies (AlwaysClarify is the safe but brittle reference), diagnostic oracle baselines (ExactEVPI, OutcomeRuleChecker), stricter certifiers (ActionEqualityCertifier, PolicyUniqueCertifier), executability and

Table 2: Matched-safety comparison across 12 domain-distribution cells. CAVI matches the strongest safe structured baselines in FAR and clarification cost, while adding certified action coverage. The Cert column reports mean ± population standard deviation across cells.

Method	Zero-FAR	FAR	Q	Act	Cert	Cert range
CAVI (oracle)	12/12	0.0000	0.2192	0.3642	0.3642 ± 0.1339	$[0.20, 0.62]$
AlwaysClarify	12/12	0.0000	1.0000	0.0000	0.0000 ± 0.0000	$[0.00, 0.00]$
ExactEVPI	12/12	0.0000	0.2192	0.3642	0.0000 ± 0.0000	$[0.00, 0.00]$
OutcomeRuleChecker	12/12	0.0000	0.2192	0.3642	0.0000 ± 0.0000	$[0.00, 0.00]$
ActionEqualityCert.	12/12	0.0000	0.2192	0.3642	0.0000 ± 0.0000	$[0.00, 0.00]$
PolicyUniqueCert.	12/12	0.0000	0.2192	0.3642	0.0000 ± 0.0000	$[0.00, 0.00]$
Self-consistency	0/12	0.0138	0.2053	0.3762	0.0000 ± 0.0000	$[0.00, 0.00]$

policy checks, conformant-goal planning, and self-consistency; full formal-baseline and local-model qualification tables are in Appendix A.6, under the Baseline Definitions and Parser-Backed Audits paragraphs.

Metrics and reporting. FAR is false actions over all evaluated calls; Cert.FPR is false certificates over issued certificates; Cert.Cov. is issued correct certificates over all calls; Q-cost is clarification decisions over all calls; in the parser confirmation, Act.Cov. is certified actions among 720 gold-actionable calls and NCR correctly routed clarifications among 720 gold-clarification calls. Zero-event results carry exact two-sided 95% Clopper–Pearson intervals and are never described as zero risk.

RQ1: Does Outcome Invariance Expand the Safe-Action Frontier? Table 2 reports the matched-safety frontier across 12 domain-distribution cells. CAVI reaches zero FAR in every cell and matches the strongest safe structured baselines in average FAR and Q-cost; its distinction is certified action coverage (36.42% on average, 0.3642 ± 0.1339, range $[0.20, 0.62]$). ExactEVPI and OutcomeRuleChecker match the frontier only with oracle or hand-written access; ActionEqualityCertifier and PolicyUniqueCertifier certify none of the additional outcome-equivalent cases, and self-consistency slightly lowers Q-cost but leaves the zero-FAR regime. Thus CAVI's gain is certification at the matched safe frontier, not merely asking fewer questions.

Figure 2 (Appendix A.6) confirms distribution-robust verifier behavior: all 12 cells retain observed FAR 0 and NCR 1, while realistic mixes yield higher Cert.Cov. and hard or boundary mixes require more clarification or non-actionability.

RQ2: Are Arena Labels Implementation-Stable? We independently recomputed all 120,000 Arena labels with enumerative and symbolic implementations that do not share final-label code; they agreed on every decision, and across 1.2 million metamorphic checks (object order, identifiers, serialization, irrelevant-object transformations) no transformed instance changed its gold decision. This supports label-level integrity rather than byte-identical outcome projections; Appendix A records the legacy Calendar serialization issue and certificate-artifact boundary.

RQ3: What Is the Parser Deployment Boundary? The frozen 2,160-call, three-model confirmation reached terminal status for every call: 1,721 outputs were schema-valid and 439 ended in terminal failure. Under intention-to-treat routing, CAVI produced no observed false action (0/2,160; exact 95% CI $[0, 0.001706]$) and no false certificate among 533 certificates (0/533; exact 95% CI $[0, 0.006897]$), certifying 24.68% of calls and 74.03% of gold-actionable calls, at substantial utility cost (Q-cost 71.94%, NCR 10.14%). The coverage-aware gate rejected all 806 known under-approximations, but the raw parser also produced no unsafe action on this population; the ablation demonstrates conservative rejection and its utility cost, not that the gate alone caused the zero-event result. Restricted-table three-way routing (NCR 0) was substantially less reliable than action certification; we treat this as a routing failure rather than evidence of safety. Per-domain breakdown, model-route variation, and earlier local qualification are in Appendix A.6.

6.1 Executable Agent Validation

Setup. Levels 1–3 test the verifier and its coverage contract in isolation; we now test whether the same decision principle operates inside an actual state-changing agent loop. The environment is a

Table 3: Integrity and frozen cross-domain parser confirmation. Zero-event bounds are upper endpoints of exact two-sided 95% Clopper–Pearson intervals. The parser study contains 720 calls per gold class.

A. Benchmark and verifier integrity		
Audit	Scale	Result
Independent label implementations	120,000	0 mismatches
Metamorphic transformations	1,200,000	0 label failures
B. Coverage-aware parser routing		
Metric	Count	Rate / bound
Schema-valid	1,721/2,160	79.68%
Terminal failure	439/2,160	20.32%
FAR	0/2,160	95% CI $[0, 0.001706]$
Cert.FPR	0/533	95% CI $[0, 0.006897]$
Cert.Cov.	533/2,160	24.68%
Act.Cov.	533/720	74.03%
Q-cost	1,554/2,160	71.94%
NCR	73/720	10.14%
Under-approximation	806/1,721	46.83%
Rejected under-approximations	806/806	100%
Three-way macro-F1	n/a	0.5560

Table 4: Executable agent validation ($n = 105$ tasks each; independent-audit canonical values). FAR is the False Action Rate, the task-level rate of executed consequential actions that violate the frozen task semantics; exact 95% Clopper–Pearson intervals and ablations in Appendix B. Clar.: clarification rate; Unnec. Q: unnecessary clarification; CERTAIN: CERTAIN_ACTION issuance rate (task-level certificate coverage is 105/105 and is reported separately).

Method	Success ↑	FAR ↓	Clar. ↓	Unnec. Q ↓	CERTAIN ↑
DirectAgent	52/105	33/105	0.000	0.000	–
ClarifyPrompt	80/105	10/105	0.514	0.162	–
AlwaysClarify	99/105	3/105	0.838	0.571	–
CAVI-PolicyOmega	97/105	1/105	0.305	0.019	83/105

local executable sandbox (simulated calendar, mail, todo, and payments applications) with persistent mutable state and state-based evaluation; actions actually mutate sandbox state, but no live external service is accessed (Section 7). The LLM agent reasons freely with read-only tools; only consequential (state-changing) actions pass through the CAVI gate (ACT / CLARIFY / NON-ACTIONABLE, with a transaction recheck before execution). All conditions share the same base model, tools, and task instances: **DirectAgent** (no gate), **ClarifyPrompt** (prompt-level clarification instruction), **AlwaysClarify** (confirmation before every consequential action), and **CAVI-PolicyOmega**; CAVI-FullState and CAVI-no-tx-recheck are diagnostic ablations (Appendix B). The frozen, blind validation split contains 105 held-out tasks across five categories—fully specified (FS, 30), harmfully ambiguous (HA, 30), harmlessly underspecified (HU, 15), non-actionable (NA, 15), and state-drift (SD, 15)—disjoint from all pilot tasks; all 630 trajectories completed sealed (Appendix B). This is a formal blind *validation*, not the final held-out test.

Main results. We operationalize executable safety as FAR. CAVI substantially reduced false actions relative to DirectAgent (1/105 vs. 33/105; paired false-action test $p = 4.1\text{e}{-9}$, $\Delta = -0.305$) and ClarifyPrompt (vs. 10/105; $p = 0.012$), while its observed FAR was statistically indistinguishable from the conservative AlwaysClarify baseline (vs. 3/105; $p = 0.625$); CAVI made no premature attempts on the 45 gold-non-act opportunities (0/45). The central thesis result is on interaction: with safety statistically tied to AlwaysClarify, CAVI required far less unnecessary clarification (0.019 vs. 0.571; paired $p = 6.9\text{e}{-18}$, $\Delta = -0.552$), and on the HU category it completed 15/15 tasks with 0/15 clarifications—ambiguity triggered interaction only when it could change the consequence of action. Task success was statistically tied with AlwaysClarify ($p = 0.774$) and higher than DirectAgent ($p = 2.0\text{e}{-11}$), at an action-coverage price of 44/60 gold-act tasks executed (Appendix B).

Transaction, coverage, and residual failure. Under state drift, transaction rechecking detected all 14/14 encountered stale snapshots, aborted and re-verified to a goal-consistent decision in every case, executed 0 unsafe stale actions, and preserved drift-task utility 14/15; without rechecking, 14 stale execution attempts occurred (all rejected by environment preconditions) and drift utility collapsed to 0/15 (paired $p = 0.023$). Conservative completion coverage remains imperfect: deterministic grounding improved coverage from 86/105 to 91/105, but 15 tasks remained outside the

theorem's premise. The validation contained one residual CAVI failure: the user requested an event deletion, but the agent proposed an event creation with the same identifier; CAVI correctly certified outcome invariance over the proposal-derived singleton space. Independent replay classifies this as a proposal–intent mismatch, not parser under-approximation. As the Semantic CERTAIN_ACTION Error Rate, this amounts to 1/83. Verifier-internal soundness is not, by itself, end-to-end task correctness.

Prospective H4-targeted OOD test. We ran a separately frozen, preregistered 80-task H4-targeted OOD evaluation pairing PolicyOmega with FullState. PolicyOmega reduced unnecessary clarification to 29/80 versus 78/80 for FullState ($\Delta = -0.6125$; paired discordance 49 FullState-only vs. 0 reverse), with no observed false actions. At the pre-clarification gate, 63/80 task pairs reached identical represented spaces; 42 exhibited the predicted direct abstraction effect (PolicyOmega CERTAIN vs. FullState CLARIFY). FullState usually certified after clarification collapsed the completion space (eventual CERTAIN: 80/80 vs. 79/80); the benefit is therefore reduced unnecessary interaction. These results support H4 within this frozen H4-targeted OOD suite, with the attribution qualification detailed in Appendix B.

Trusted-boundary audits. Table 1 summarizes the theorem assumptions. Controlled parser omission creates false actions; fail-closed routing restores zero errors. Dropping `refund_channel` or `attendees_notified` collapses all 40 targeted harmful pairs and gives Cert.FPR 0.3333. One-shot certificate reuse yields unsafe first actions in 42.86%, whereas stepwise re-verification eliminates this. Symbolic product checking stays below 0.0015 ms p95 at 65,536 states when outcome projections factor.

7 Scope and Trusted Boundary

Theorem 1 is conditional on conservative completion coverage, faithful Ω, exact verification, and transaction stability. End-to-end correctness additionally requires proposal faithfulness—the action specification supplied to the verifier being consistent with the request-derived task representation—an agent-integration assumption distinct from the theorem's algebraic premises (Table 1). The residual failure was classified as a proposal–intent mismatch.

CAVI's strongest evidence remains verifier-contract evidence. In the frozen parser confirmation, 20.32% of calls were not schema-valid and 46.83% of schema-valid parses were known under-approximations; coverage-aware routing rejected all of them. Additional artifact boundaries remain: legacy Calendar projections required canonicalization of unordered collections, legacy serialized certificates omitted canonical outcome hashes, field-drop audits cannot prove global schema completeness, and CERTAIN-ACTION remains worst-case coNP-hard for expressive DSLs.

Limitations. The executable environment is a local sandbox, not production services. CAVI incurs additional inference cost relative to direct execution (2.89 vs. 1.00 LLM calls/task). Conservative completion coverage remains imperfect (grounded coverage 91/105, with 15 tasks outside the theorem's premise). Proposal–request alignment is an additional trusted boundary (one semantic certificate error, 1/83, observed). We do not claim production-service safety.

8 Conclusion

CAVI reframes underspecified agent action as verification over completion-induced outcomes rather than confidence over interpretations: ambiguity should trigger interaction only when it can change the consequence of action. Controlled Arena experiments, integrity audits, and a frozen parser confirmation establish the verifier-level property and its parser boundary; a blind executable validation shows the same principle operating inside an actual state-changing agent loop—sharply reducing false actions relative to direct and prompt-only execution, near the AlwaysClarify safety regime with far less unnecessary clarification, and re-verifying under drift. The validation exposes both completion-coverage and proposal-faithfulness boundaries: CAVI verifies whether ambiguity within the supplied action specification can change consequence; it does not by itself prove that the agent proposed the correct specification. CAVI makes the conditions under which action is justified explicit, machine-checkable, and auditable.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools to revise and polish the manuscript and to implement experimental code. We also used large language models as experimental endpoints in the reported evaluations. We reviewed all AI-assisted writing and code, and inspected and tested the experimental code against the protocols and checks described in this paper. We take responsibility for the final content of this work, including all text, claims, code, and experimental artifacts produced with the aid of generative AI.

See the ICLR 2027 AI Policy for Authors for more details. This statement should not be more than 1 page.

REPRODUCIBILITY STATEMENT

(This section is **recommended** and does not count toward the page limit.)

All code and experimental artifacts will be uploaded with this manuscript and publicly released. The appendix provides the corresponding methodological, implementation, evaluation-protocol, and experimental details needed to reproduce the reported results.

REFERENCES

Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, Zico Kolter, Matt Fredrikson, Yarin Gal, and Xander Davies. AgentHarm: A benchmark for measuring harmfulness of LLM agents. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/c493d23af93118975cdbc32cbe7323f5-Abstract-Conference.html.

Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tram‘er. AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=m1YYAQjO3w.

Aarya Doshi, Yining Hong, Congying Xu, Eunsuk Kang, Alexandros Kapravelos, and Christian Kästner. Towards verifiably safe tool use for LLM agents. In *2026 IEEE/ACM 48th International Conference on Software Engineering, NIER Track (ICSE-NIER 2026)*, 2026. doi: 10.1145/3786582.3786839.

Eric Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pp. 159–166, New York, NY, USA, 1999. ACM Press. doi: 10.1145/302979.303030.

Tomasz Imieliński and Witold Lipski, Jr. Incomplete information in relational databases. *Journal of the ACM*, 31(4):761–791, 1984. doi: 10.1145/1634.1886.

Adharsh Kamath, Sishen Zhang, Calvin Xu, Shubham Ugare, Gagandeep Singh, and Sasa Misailovic. Enforcing temporal constraints for LLM agents, 2025.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=VD-AYtP0dve.

Lorenz Kuhn, Annika Aicher, Willa Potosnak, Zi Lin, Hamid Palangi, Aditya Kumar, Dragomir Radev, Asli Celikyilmaz, and Saleema Amershi. Clarify when necessary: Resolving ambiguity through interaction with lms. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 2025.

Leonid Libkin. Certain answers as objects and knowledge. *Artificial Intelligence*, 175(7–8):1149–1167, 2011.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as agents. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/e9df36b21ff4ee211a8b71ee8b7e9f57-Abstract-Conference.html.

Héctor Palacios and Héctor Geffner. Compiling uncertainty away in conformant planning problems with bounded width. *Journal of Artificial Intelligence Research*, 35:623–675, 2009. doi: 10.1613/jair.2708.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=dHng2O0Jjr.

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris Maddison, and Tatsunori Hashimoto. Identifying the risks of LM agents with an LM-emulated sandbox. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=GEcwtMk1uA.

Vincent Siu, Jingxuan He, Kyle Montgomery, Zhun Wang, Neil Gong, Chenguang Wang, and Dawn Song. A framework for formalizing LLM agent security, 2026.

Haoyu Wang, Christopher M. Poskitt, and Jun Sun. AgentSpec: Customizable runtime enforcement for safe and reliable LLM agents, 2025. URL https://arxiv.org/abs/2503.18666.

Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. MINT: Evaluating LLMs in multi-turn interaction with tools and language feedback. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jp3gWrMuIZ.

Cailin Winston, Claris Winston, and René Just. Solver-aided verification of policy compliance in tool-augmented LLM agents, 2026.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik R. Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=roNSXZpUDN.

Ziyu Yao, Yu Su, Huan Sun, and Wen-tau Yih. Model-based interactive semantic parsing: A unified framework and a text-to-SQL case study. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5447–5458, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1547. URL https://aclanthology.org/D19-1547/.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=oKn9c6ytLx.

A Technical Appendix

This appendix provides additional proofs, protocols, dataset details, and extended audits supporting the main paper.

Integrity and Legacy Artifact Boundaries. The independent enumerative and symbolic labelers agree on all 120,000 final Arena decisions, and 1.2 million metamorphic checks produce no label failure. This is a final-label claim, not a claim that legacy projected bytes were identical. The audit found 9,900 Calendar projection serialization mismatches caused solely by attendee ordering. They did not change gold decisions, but motivated the canonicalization rule in the outcome-schema specification: semantically unordered collections must be sorted by policy-defined identity keys before comparison or hashing.

The 120,000 legacy serialized certificates also omitted a canonical outcome hash and outcome-schema version. Accordingly, the main paper claims integrity of verifier decisions and defines the revised certificate schema, but does not claim replay-complete validation of the legacy certificate objects. A Certificate v2 replay and hash-validation audit is required before making that stronger artifact claim.

Full CAVI Certificate and DSL. The certificate specification requires a certificate-schema version, outcome-schema version, canonical outcome hash, snapshot and policy hashes, DSL/completion-space hash, latent-registry version, transaction preconditions, and assumption-ledger hash. Clarification certificates additionally require two counterexample hashes, their outcome hashes, and distinguishing fields. Non-actionability certificates require executable-completion count zero, a reason code, and a replayable proof or witness. These are schema requirements; as noted in Appendix A, the legacy artifacts did not serialize every field.

Outcome Schema Protocol. Outcome schemas are derived from effect inventories covering object identity, state changes, monetary or resource effects, notifications, authorization, reversibility, audit records, and downstream preconditions. Domain schemas include refund channel and order state for order/refund tasks, attendee notification and recurrence effects for calendar tasks, and row identity plus audit-visible mutations for table-action tasks. Semantically unordered collections are canonically serialized by policy-defined identity keys before comparison and hashing; order is retained only when safety-relevant. Schema version changes invalidate stale certificates, and harmful-pair field-drop audits test whether removing a field merges policy-different outcomes.

Theory Statements and Proofs. The main text gives the proof sketch of Theorem 1 and the argument for Proposition 1. Full formal statements and proofs follow.

A.1 Formal Preliminaries

Objects. Fix a request x, an environment snapshot e, an interface policy π (visible objects, actions, preconditions, policy rules), and a concrete outcome space $\mathcal{O}$ (structured records of state changes, monetary effects, notifications, reversibility, audit-visible mutations).

- A *latent completion* z assigns values to the latent slots of x (target identifiers, tie-breaking rules, scopes, amounts).
- The *true feasible completion space* $Z_{\text{true}}(x, e, \pi) = \{z : z \text{ consistent with } x, e, \pi\}$ is the set of completions genuinely compatible with the request, snapshot, and policy. It is a semantic object, generally not computable by the system.
- The *represented completion space* $\widehat{Z}(x, e, \pi)$ is the space constructed by the parser/grounding pipeline and consumed by the verifier.
- The *execution semantics* $F(z, e, \pi) \subseteq \mathcal{O}$ is *set-valued*: it contains every outcome reachable when completion z is executed under e, π (API failures, races, policy nondeterminism). Deterministic simulators are the special case $|F(z, e, \pi)| \leq 1$ for all z.
- The *outcome abstraction* Ω induces an equivalence relation $\equiv_\Omega$ on $\mathcal{O}$: $o_1 \equiv_\Omega o_2$ iff o_1, o_2 agree on all policy-retained fields (with canonical serialization of unordered collections). $[o]_\Omega$ denotes the Ω-equivalence class of o.

Three notions of equivalence. We distinguish:

1. *Syntactic action equality*: the action terms of two completions are identical. CAVI never certifies on this relation.

2. Ω-*equivalence* $\equiv_\Omega$: equality of projected policy-retained fields.
3. *Domain safety equivalence* $\equiv_{\text{safe}}$: the domain's actual safety relation on outcomes ("these two executions are interchangeable for policy purposes"). It is not formalized field-by-field here; Ω connects to it via faithfulness below.

Projection map notation. To avoid ambiguity, we may equivalently view Ω as inducing a canonical projection map $P_\Omega : \mathcal{O} \to \mathcal{Y}_\Omega$ whose fibers are the $\equiv_\Omega$-classes:

$$o_1 \equiv_\Omega o_2 \iff P_\Omega(o_1) = P_\Omega(o_2).$$

Theorem statements typically use $\equiv_\Omega$ for conciseness; complexity statements may use $P_\Omega(o)$ equality when emphasizing witness verification.

Assumption 1 (Faithful abstraction, A2)**.** *Ω is faithful to the domain safety relation on the outcomes under consideration:*

$$o_1 \equiv_\Omega o_2 \;\Rightarrow\; o_1 \equiv_{\text{safe}} o_2.$$

The converse is not *assumed: Ω may retain more distinctions than safety requires (over-refinement), which costs certification coverage but not soundness. Faithfulness is domain-specific; it is assumed, audited by harmful-pair mutation tests, and never derived from the verifier.*

Executability and projected outcome classes.

Definition 2 (Executable completion)**.** $\text{Exec}(z, e, \pi)$ *holds iff executing z under e, π can produce at least one outcome, i.e. $F(z, e, \pi) \neq \varnothing$. A completion that is consistent with x but blocked by interface preconditions is feasible yet non-executable.*

Definition 3 (Projected outcome classes)**.**

$$\mathcal{O}_\Omega(\widehat{Z}) \;=\; \big\{\, [o]_\Omega \;:\; z \in \widehat{Z}(x, e, \pi),\; o \in F(z, e, \pi) \,\big\}.$$

Non-executable completions contribute no outcomes and hence no classes.

Decision convention (anti-vacuity). The verifier decision is a function of $\mathcal{O}_\Omega(\widehat{Z})$:

$$|\mathcal{O}_\Omega(\widehat{Z})| = 0 \Rightarrow \textsc{Non-Actionable}, \quad |\mathcal{O}_\Omega(\widehat{Z})| = 1 \Rightarrow \textsc{Certain-Action}, \quad |\mathcal{O}_\Omega(\widehat{Z})| > 1 \Rightarrow \textsc{Clarify}. \tag{1}$$

The condition for CERTAIN-ACTION is $|\mathcal{O}_\Omega(\widehat{Z})| = 1$, *not* ≤ 1: when no represented completion is executable the statement "all reachable outcomes are equivalent" is vacuously true, and the correct decision is NON-ACTIONABLE, not CERTAIN-ACTION. Equivalently, $|\mathcal{O}_\Omega(\widehat{Z})| = 1$ already entails the existence of at least one executable represented completion. $\widehat{Z} = \varnothing$ yields $|\mathcal{O}_\Omega(\widehat{Z})| = 0$ and thus NON-ACTIONABLE as well.

Definition 4 (Certain-actionable)**.** *A represented space $\widehat{Z}$ is* certain-actionable *(under e, π, Ω) iff $|\mathcal{O}_\Omega(\widehat{Z})| = 1$, i.e.*

$$\exists z_0 \in \widehat{Z},\; o_0 \in F(z_0, e, \pi) \quad \textit{and} \quad \forall z_1, z_2 \in \widehat{Z},\; \forall o_1 \in F(z_1, e, \pi),\; \forall o_2 \in F(z_2, e, \pi) :\; o_1 \equiv_\Omega o_2.$$

Assumption 5 (Exact verification, A3)**.** *The verifier exactly determines the projected outcome classes induced by all reachable outcomes of represented completions: $\mathcal{O}_\Omega(\widehat{Z})$ is correctly accounted for (no class omitted or falsely introduced). Any procedure (enumerative, symbolic model checking, or otherwise) that correctly establishes $\mathcal{O}_\Omega(\widehat{Z})$ satisfies A3. Observing one outcome per completion does not satisfy A3 when F is genuinely set-valued. A3 is a statement about correct computation of a mathematical set; it does* not *assert soundness of the certificate, which additionally requires A1, A2, and A4.*

Assumption 6 (Conservative completion coverage, A1)**.** $Z_{\text{true}}(x, e, \pi) \subseteq \widehat{Z}(x, e, \pi)$.

Assumption 7 (Transaction stability, A4)**.** *Let $\theta = (e, \pi, \Omega\text{-version}, \text{action preconditions})$ be the contract tuple against which the verifier computed $\mathcal{O}_\Omega(\widehat{Z})$. A4 holds iff execution is governed by the same tuple θ. Operationally, the transaction gate compares snapshot hash, policy hash, schema hash, and a precondition dry-run immediately before execution and aborts/re-verifies on any mismatch.*

A.2 CERTAIN-ACTION SOUNDNESS

Theorem 8 (Certain-action soundness)**.** *Fix x, e, π, Ω. Assume A1 (coverage), A2 (faithful Ω), A3 (exact verification), and A4 (transaction stability). If the verifier issues a certain-action certificate with certified class $C^\star$, i.e. $|\mathcal{O}_\Omega(\widehat{Z})| = 1$ and $\mathcal{O}_\Omega(\widehat{Z}) = \{C^\star\}$, then:*

$$\forall z \in Z_{\mathrm{true}}(x, e, \pi),\ \forall o \in F(z, e, \pi) : \quad [o]_\Omega = C^\star,$$

and consequently any two outcomes reachable from true feasible completions are equivalent under the domain safety relation $\equiv_{\mathrm{safe}}$.

Proof. The argument is written for the general set-valued F; the deterministic case is the $|F| \leq 1$ specialization.

1. Let $z \in Z_{\mathrm{true}}(x, e, \pi)$ be arbitrary.
2. By A1 (coverage), $z \in \widehat{Z}(x, e, \pi)$.
3. Let $o \in F(z, e, \pi)$ be arbitrary. (If $F(z, e, \pi) = \varnothing$ there is nothing to show for this z.)
4. By A3 (exact verification), every class arising from a represented completion and one of its reachable outcomes belongs to the computed set: since $z \in \widehat{Z}$ and $o \in F(z, e, \pi)$, $[o]_\Omega \in \mathcal{O}_\Omega(\widehat{Z})$.
5. The certificate asserts $|\mathcal{O}_\Omega(\widehat{Z})| = 1$ with unique class $C^\star$, hence $[o]_\Omega = C^\star$.
6. Since z and o were arbitrary, all outcomes reachable from all true feasible completions lie in $C^\star$; in particular any two such outcomes o, o' satisfy $o \equiv_\Omega o'$. (This step uses only A3 and the certificate condition $|\mathcal{O}_\Omega(\widehat{Z})| = 1$.)
7. By A2 (faithfulness), the pair (o, o') satisfies $o \equiv_{\mathrm{safe}} o'$ directly (one application of A2 to the pair; no transitivity assumption needed).
8. By A4, the contract tuple governing execution is the tuple against which $\mathcal{O}_\Omega(\widehat{Z})$ was computed, so the statement above applies to the executed semantics themselves, not to a stale copy of them.

□

Corollary 9 (Nondeterministic interfaces)**.** *Theorem 8 holds unchanged when F is set-valued, because the proof quantifies over every $o \in F(z, e, \pi)$. Certification therefore requires*

$$\forall z_1, z_2 \in \widehat{Z},\ \forall o_1 \in F(z_1, e, \pi),\ \forall o_2 \in F(z_2, e, \pi) :\ o_1 \equiv_\Omega o_2,$$

and is not *established by comparing one sampled outcome per completion: a verifier that observes one outcome per completion computes a subset of $\mathcal{O}_\Omega(\widehat{Z})$ and violates A3 whenever an unobserved branch carries a different class.*

A.3 NECESSITY OF THE TRUSTED BOUNDARY

Each result below is constructive: it exhibits an instance satisfying all assumptions except the one under test in which a certificate is issued but the conclusion of Theorem 8 fails. They show the assumptions *can* fail to be dispensable (*permit* false certificates); they do not assert that every violation produces one.

Proposition 10 (Coverage necessity; main text Proposition 1, part 1)**.** *If A1 fails, a certain-action certificate can be unsound even when A2–A4 hold.*

Proof. Let there be one latent slot with two interface-feasible referents α, β, and let $Z_{\mathrm{true}} = \{z_\alpha, z_\beta\}$ while the parser represents $\widehat{Z} = \{z_\alpha\}$. Let $F(z_\alpha) = \{o_A\}$, $F(z_\beta) = \{o_B\}$ with $[o_A]_\Omega \neq [o_B]_\Omega$ and $o_A \not\equiv_{\mathrm{safe}} o_B$, and let Ω be faithful everywhere else. Then $\mathcal{O}_\Omega(\widehat{Z}) = \{[o_A]_\Omega\}$,

so the verifier legitimately issues a certain-action certificate for $C^\star = [o_A]_\Omega$; A2, A3, A4 all hold on this instance. Yet $z_\beta \in Z_{\text{true}}$ reaches $o_B \notin C^\star$, so the certificate's guarantee is false. Conversely, under-approximation does not *force* a false certificate: the same instance with $F(z_\beta) = \{o_A\}$ violates A1 but the certificate remains correct. Hence: under-approximation permits false certificates; it does not imply that every under-approximation produces one. □

Proposition 11 (Ω-faithfulness necessity; Proposition 1, part 2)**.** *If A2 fails, a certain-action certificate can be unsound even when A1, A3, A4 hold.*

Proof. Let $Z_{\text{true}} = \widehat{Z} = \{z_1, z_2\}$ (coverage holds), $F(z_1) = \{o_1\}$, $F(z_2) = \{o_2\}$, with $o_1 \equiv_\Omega o_2$ but $o_1 \not\equiv_{\text{safe}} o_2$ (e.g. Ω drops `refund_channel` and the two completions refund through different channels). Then $|\mathcal{O}_\Omega(\widehat{Z})| = 1$ and the verifier issues a certificate, so the Ω-class membership conclusion of Theorem 1's first part remains valid, but the safety-equivalence conclusion ($o \equiv_{\text{safe}} o'$) fails. Faithful Ω is therefore a necessary boundary, not a convenience. □

Proposition 12 (Transaction-stability necessity)**.** *If A4 fails, a certificate valid at verification time can be unsound at execution time even when A1–A3 hold at verification time.*

Proof. At snapshot e_0 let $Z_{\text{true}} = \widehat{Z} = \{z\}$, $F(z, e_0) = \{o\}$, so $|\mathcal{O}_\Omega(\widehat{Z})| = 1$ and a certificate for $[o]_\Omega$ is issued. Between verification and execution the environment changes to e_1 (a concurrent actor modifies state), and the same action now yields $F(z, e_1) = \{o'\}$ with $[o']_\Omega \neq [o]_\Omega$ (e.g. the target record no longer exists, so the action either fails or applies to a different record). Executing under the old certificate realizes an outcome outside the certified class. The defect is precisely the mismatch between the verified tuple $(e_0, \pi, \ldots)$ and the governing tuple $(e_1, \pi, \ldots)$: A4 is non-decorative.

Remark: operationally, the transaction gate also verifies that the executed action matches the verified proposal; however, in the request-derived theorem this is absorbed by quantification over $z \in Z_{\text{true}}$, and in the proposal- pipeline it holds by construction. □

Closed-interface as a scope-transfer condition. Theorem 8 is interface-relative: $Z_{\text{true}}(x, e, \pi)$ quantifies over completions consistent with the modeled interface contract π. Relevant actions or objects outside π lie outside the theorem's modeled universe.

Transferring the guarantee to a broader external/open-world semantics requires a separate closed-interface bridge assumption: if relevant completions exist outside π, then Z_{true} computed relative to π may omit them, and additional assumptions are needed to ensure completeness with respect to the external semantics. The closed-interface condition thus serves as a boundary assumption for cross-model transfer, not as part of the interface-relative theorem itself.

Proposition 13 (Modeled-nondeterminism necessity)**.** *If the execution relation presented to the verifier omits a reachable branch, A3 fails with respect to the true F and a false certificate can result even when A1, A2, A4 hold.*

Proof. Let the true semantics be $F(z) = \{o_{\text{ok}}, o_{\text{part}}\}$ where o_{part} is a partial-mutation/failure outcome with $[o_{\text{part}}]_\Omega \neq [o_{\text{ok}}]_\Omega$, and let the modeled relation used by the verifier be $\hat{F}(z) = \{o_{\text{ok}}\}$. The verifier computes $\{[o_{\text{ok}}]_\Omega\}$ and certifies, but the true reachable outcome set contains a second class. Exact verification must be exact about the *true* reachable set; unmodeled branches are indistinguishable from coverage failure at the outcome level. □

A.4 Monotonicity Results

Definition 14 (Schema refinement)**.** Ω_2 *is a* refinement *of* Ω_1 *(written* $\Omega_1 \preceq \Omega_2$*) iff* Ω_2 *retains at least the distinctions of* Ω_1*:*

$$o_1 \equiv_{\Omega_2} o_2 \;\Rightarrow\; o_1 \equiv_{\Omega_1} o_2,$$

i.e. $\equiv_{\Omega_2} \subseteq \equiv_{\Omega_1}$ *as relations on* $\mathcal{O}$*.*

Theorem 15 (Schema-refinement monotonicity)**.** *Let* $\Omega_1 \preceq \Omega_2$*. Then* $|\mathcal{O}_{\Omega_2}(\widehat{Z})| = 1$ *implies* $|\mathcal{O}_{\Omega_1}(\widehat{Z})| = 1$*: a completion space certain-actionable under the finer schema is certain-actionable under the coarser schema.*

Proof. Define $\varphi : \mathcal{O}_{\Omega_2}(\widehat{Z}) \to \mathcal{O}_{\Omega_1}(\widehat{Z})$ by $\varphi([o]_{\Omega_2}) = [o]_{\Omega_1}$. *Well-defined:* if $[o]_{\Omega_2} = [o']_{\Omega_2}$ then $o \equiv_{\Omega_2} o'$, hence $o \equiv_{\Omega_1} o'$ by refinement, hence $[o]_{\Omega_1} = [o']_{\Omega_1}$. *Surjective:* every class in $\mathcal{O}_{\Omega_1}(\widehat{Z})$ is $[o]_{\Omega_1}$ for some reachable o, the image of $[o]_{\Omega_2}$. A surjection from a singleton is a singleton, so $|\mathcal{O}_{\Omega_1}(\widehat{Z})| \leq 1$; and refinement does not change which completions are executable, so $|\mathcal{O}_{\Omega_2}(\widehat{Z})| = 1$ (which entails at least one executable completion) implies $|\mathcal{O}_{\Omega_1}(\widehat{Z})| = 1$, not 0. □

Proposition 16 (Converse fails; coverage/safety tradeoff)**.** *Certainty under Ω_1 does not imply certainty under Ω_2.*

Proof. Let two completions send the same notification-free state change, differing only in whether an attendee is notified: Ω_1 ignores notifications, Ω_2 retains them. Then $|\mathcal{O}_{\Omega_1}| = 1$ while $|\mathcal{O}_{\Omega_2}| = 2$. Coarser schemas therefore certify at least as much as finer ones ($|\mathcal{O}_{\Omega_1}| \leq |\mathcal{O}_{\Omega_2}|$ always), but coarsening can only be adopted while preserving faithfulness (A2): certification coverage and Ω-faithfulness trade off, and the tradeoff is resolved by policy authorship, not by the verifier. □

Theorem 17 (Completion-space expansion is anti-monotone for certificates)**.** *Let $\widehat{Z}_1 \subseteq \widehat{Z}_2$. Then $\mathcal{O}_\Omega(\widehat{Z}_1) \subseteq \mathcal{O}_\Omega(\widehat{Z}_2)$, hence $|\mathcal{O}_\Omega(\widehat{Z}_2)| = 1$ implies $|\mathcal{O}_\Omega(\widehat{Z}_1)| \leq 1$, and if $\widehat{Z}_1$ contains at least one executable completion then $|\mathcal{O}_\Omega(\widehat{Z}_1)| = 1$. Expanding the represented space can preserve or destroy a certificate; it can never create outcome invariance that was absent.*

Note: the inference "certainty under $\widehat{Z}_2 \Rightarrow$ certainty under $\widehat{Z}_1$" presupposes $\widehat{Z}_1$ has at least one executable completion; otherwise $|\mathcal{O}_\Omega(\widehat{Z}_1)| = 0$ and the correct decision is NON-ACTIONABLE.

Proof. Set inclusion is immediate from Definition 3: every completion of $\widehat{Z}_1$ is a completion of $\widehat{Z}_2$. A subset of a singleton has cardinality at most 1. If $\widehat{Z}_1$ has an executable completion then $\mathcal{O}_\Omega(\widehat{Z}_1) \neq \varnothing$, so its cardinality is exactly 1. For the edge case: if $\widehat{Z}_1$ has no executable completion, $|\mathcal{O}_\Omega(\widehat{Z}_1)| = 0$ and the correct decision is NON-ACTIONABLE, not CERTAIN-ACTION—the anti-vacuity convention of (1) is essential here. □

Proposition 18 (Converse fails)**.** *Certainty under $\widehat{Z}_1$ does not imply certainty under $\widehat{Z}_2$.*

Proof. Take $\widehat{Z}_1 = \{z_1\}$ with $F(z_1) = \{o\}$: necessarily $|\mathcal{O}_\Omega(\widehat{Z}_1)| = 1$. Expand to $\widehat{Z}_2 = \{z_1, z_2\}$ with $F(z_2) = \{o'\}$ and $[o']_\Omega \neq [o]_\Omega$; then $|\mathcal{O}_\Omega(\widehat{Z}_2)| = 2$ and the certificate disappears. □

A.5 NON-ACTIONABILITY AND CLARIFICATION DECISIONS

Proposition 19 (Non-actionability soundness)**.** *Assume A1. If no represented completion is executable, $\forall z \in \widehat{Z} : \neg\,\mathrm{Exec}(z, e, \pi)$, then no true feasible completion is executable: $\forall z \in Z_{\mathrm{true}} : \neg\,\mathrm{Exec}(z, e, \pi)$.*

Proof. Immediate: $z \in Z_{\mathrm{true}} \Rightarrow z \in \widehat{Z}$ by A1, and the premise applies to every element of $\widehat{Z}$. Equivalently, $|\mathcal{O}_\Omega(\widehat{Z})| = 0 \Rightarrow$ no true completion contributes any reachable outcome under e, π. Note the strict distinction between a *true latent completion* (consistent with the request) and an *interface-executable completion* ($F \neq \varnothing$): the certificate states that nothing the interface admits can execute the request, not that no consistent interpretation exists. Without A1 the statement fails exactly as in Proposition 10: an unrepresented true completion may be executable. □

Proposition 20 (Clarification counterexample soundness)**.** *If the verifier exhibits $z_1, z_2 \in \widehat{Z}$ and $o_1 \in F(z_1, e, \pi)$, $o_2 \in F(z_2, e, \pi)$ with $[o_1]_\Omega \neq [o_2]_\Omega$, then $\widehat{Z}$ is not certain-actionable.*

Proof. $[o_1]_\Omega, [o_2]_\Omega \in \mathcal{O}_\Omega(\widehat{Z})$ by Definition 3, so $|\mathcal{O}_\Omega(\widehat{Z})| \geq 2$. This is a *represented-space* counterexample: it proves that the represented space cannot be certified as certain-actionable. Under CAVI's fail-closed interaction policy, this condition routes to clarification.

It does *not* prove that both z_1, z_2 are true user intents; that additionally requires $z_1, z_2 \in Z_{\mathrm{true}}$ (A1 gives only $Z_{\mathrm{true}} \subseteq \widehat{Z}$, not the reverse). The lack of certification on the represented space justifies

routing to clarification under CAVI's interaction policy, regardless of which counterexample reflects actual intent. □

A.6 COMPLEXITY

Definition 21 (The decision problem CERTAIN-ACTION)**.** *An instance consists of a completion description (feasibility predicate $C(z)$), execution specification ($F(z)$ for each feasible z), outcome projection Ω, and designated reference (z_0, o_0). We define the language* CERTAIN-ACTION *over polynomial-size encodings as follows:*

Well-formed instances satisfy $C(z_0)$ and $o_0 \in F(z_0)$; malformed encodings receive classification NO. The instance is a YES-instance iff

$$\forall z\ \forall o: \quad C(z) \wedge o \in F(z) \ \Rightarrow\ P_\Omega(o) = P_\Omega(o_0),$$

i.e. all feasible completion/outcome pairs project to the reference class (equivalently, $|\mathcal{O}_\Omega| = 1$, since well-formedness guarantees at least one executable completion). Malformed encodings are classified NO.

The complement's violation form is

$$\exists z, o:\ C(z) \wedge o \in F(z) \wedge P_\Omega(o) \neq P_\Omega(o_0),$$

the counterexample-guided formulation used for symbolic audits. Well-formedness of instances (including $C(z_0)$ and $o_0 \in F(z_0)$) is polynomial-time decidable under R1–R2; malformed encodings are classified NO by convention or equivalently restricted to a recognizable well-formed sublanguage.

Assumption 22 (Representation assumptions for complexity statements)**.** *R0 completions z admit polynomial-size encodings in the instance size; R1 feasibility predicate $C(z)$ is decidable in time polynomial in the instance encoding; R2 outcome membership $o \in F(z)$ is polynomial-time decidable, and outcomes have polynomial-size representations (deterministic polynomial-time execution semantics are the canonical case, where R2 holds directly); R3 computing $P_\Omega(o)$ and comparing two projections is polynomial-time. These assumptions hold for the canonical deterministic simulation regime used by CAVI's primary verifier; for genuinely set-valued F, checkability of membership may require stronger computational guarantees.*

Theorem 23 (CERTAIN-ACTION $\in$ coNP)**.** *Under Assumption 22,* CERTAIN-ACTION $\in$ *coNP.*

Proof. We show the complement is in NP. First, well-formedness is polynomial-time decidable by R1–R2; if an encoding is malformed, this is recognizable in polynomial time without requiring a projected-difference witness, and such malformed instances belong to the complement. Otherwise the instance is well formed; a NO-instance then admits a witness (z, o) with $C(z)$, $o \in F(z)$, and $P_\Omega(o) \neq P_\Omega(o_0)$ (Definition 21). The witness has polynomial size (R0+R2), and verification computes $C(z)$ (R1), checks $o \in F(z)$ (R2), and computes and compares $P_\Omega(o)$ with $P_\Omega(o_0)$ (R3), all in polynomial time. Hence NO $\in$ NP and CERTAIN-ACTION $\in$ coNP. These checkability conditions are genuine premises: without them the witness is not polynomially verifiable. □

Theorem 24 (CERTAIN-ACTION is coNP-hard)**.** *For succinct DSLs capable of encoding Boolean assignments as completions and polynomial-time Boolean predicates into the execution semantics,* CERTAIN-ACTION *is coNP-hard under polynomial-time many-one reductions. This is a worst-case statement over expressive DSLs; it does not assert hardness for every CAVI domain language.*

Proof. Reduce from UNSAT, which is coNP-complete. Let $\phi(v_1, \ldots, v_n)$ be a Boolean formula. Construct a CERTAIN-ACTION instance as follows. *Completions:* pairs $z = (b, a)$ with branch bit $b \in \{0, 1\}$ and assignment $a \in \{0, 1\}^n$. *Feasibility:* $C(b, a)$ holds for all 2^{n+1} completions (polysize description; R1 holds). *Execution semantics* (deterministic, hence R2 holds):

$$F(b, a) \ =\ \begin{cases} \{o_1\} & \text{if } b = 1 \text{ and } \phi(a) = 1, \\ \{o_0\} & \text{otherwise,} \end{cases}$$

implemented by one evaluation of ϕ, so the construction is polynomial in $|\phi| + n$. *Projection:* $P_\Omega(o_0) \neq P_\Omega(o_1)$, both computable in constant TIME (R3). *Reference:* $z_0 = (0, 0^n)$ with $o_0 \in F(z_0)$; note the branch $b = 0$ maps to o_0 for *every* a, so o_0 is always reachable regardless of ϕ.

Forward direction ($\phi \in$ UNSAT $\Rightarrow$ instance is YES): if ϕ is unsatisfiable, the case $b = 1 \wedge \phi(a) = 1$ never arises, so $F(b, a) = \{o_0\}$ for all completions. Thus $\mathcal{O}_\Omega = \{[o_0]_\Omega\}$, a singleton, and the instance is YES.

Backward direction (instance YES $\Rightarrow \phi \in$ UNSAT): contrapositively, suppose ϕ is satisfiable with witness $a^\star$. Then $F(1, a^\star) = \{o_1\}$ while $F(0, 0^n) = \{o_0\}$, so both $[o_0]_\Omega$ and $[o_1]_\Omega$ belong to $\mathcal{O}_\Omega$; since $P_\Omega(o_0) \neq P_\Omega(o_1)$, $|\mathcal{O}_\Omega| \geq 2$ and the instance is NO.

Both directions hold, the reduction is polynomial, and UNSAT $\leq_p$ CERTAIN-ACTION. The fixed $b = 0$ reference branch is essential: without it a tautological ϕ would yield the singleton $\{[o_1]_\Omega\}$ and the equivalence "singleton iff unsatisfiable" would be false. □

Corollary 25 (Qualified completeness statement)**.** *For succinct DSLs capable of encoding Boolean assignments as completions and polynomial-time outcome predicates into the execution semantics, and under Assumption 22,* CERTAIN-ACTION *is coNP-complete. This is a worst-case statement over expressive DSLs; it does not assert hardness for every CAVI domain language. For a fixed bounded completion space, explicit enumeration decides the problem in time linear in the enumerated space, which is the regime used for auditability.*

Consistency with the counterexample-guided formulation. The NO-witness (z, o) of Theorem 23 is exactly a projected-difference witness against the reference (z_0, o_0), matching the main-text formula $\exists z, o : C(z) \wedge o \in F(z, e, \pi) \wedge P_\Omega(o) \neq P_\Omega(o_0)$. Under set-valued F the existential ranges over all reachable branches, so symbolic product-form checking must cover branch sets, not single sampled outcomes.

Agent Integration Condition / Interface Mapping P. Theorem 8 reasons about the *request-derived* spaces $Z_{\text{true}}(x, e, \pi)$ and $\widehat{Z}(x, e, \pi)$. In the executable agent system, the verifier instead receives an agent-*proposal*-derived specification: completions are enumerated for the proposed consequential action.

We define the Agent Integration Condition $P(x, a, e, \pi)$ as:

Definition 26 (Agent Integration Condition / Interface Mapping)**.** *The proposed consequential action specification a is semantically consistent with the request-derived task specification with respect to operation type, target class/object (when specified by the request), and explicit constraints. Formally we write $P(x, a, e, \pi)$ to denote this condition.*

This condition connects the executable representation to the request-derived theorem; it is *not* an algebraic premise of Theorem 8 and is not added to A1–A4.

Its failure mode is precisely a certified-but-task-inconsistent action: outcome invariance can hold over a proposal-derived singleton space while the proposal itself does not implement the request. Such failures are audited as semantic certificate errors, and the verifier algebra is correctly implicated in none of them. The main-text trusted-boundary table marks this condition with † accordingly.

Table 5: Arena domain inventory (moved from the main text). Each domain exposes side effects for which surface ambiguity may or may not change the projected outcome.

Domain	Side effects	Latent variables	Outcome fields
Order	refund, shipment, notice	order id, tie rule, eligibility	target, refund channel, stop, notice
Calendar	room, recurrence, notice	event id, time, scope	event, room release, attendees, scope
Table	row update, permission, audit	row id, filter, column	row delta, permission, audit mutation

Dataset Generation Details. Generation samples a snapshot and policy variant, samples a true latent completion set, renders a request with selected slots omitted or constrained, enumerates Z_{true}, executes every feasible completion through the domain simulator, and projects outcomes through the full schema. Categories include fully specified, missing irrelevant slot, missing relevant slot, conflicting constraints, and interface-unverifiable requests.

Baseline Definitions. Baselines cover always-clarify and safe-clarify policies, diagnostic oracle variants, stricter certifiers, precondition-only and policy-model-checking variants, conformant-goal

Table 6: Formal-baseline comparison moved from the main paper. Executability and policy checks do not establish outcome invariance.

Method	Zero-FAR	FAR	Q	Cert
CAVI	12/12	0.0000	0.2192	0.3642
ActionEqualityCert.	12/12	0.0000	0.2192	0.0000
PolicyUniqueCert.	12/12	0.0000	0.2192	0.0000
PolicyModelCheck	0/12	0.2192	0.0000	0.0000
PreconditionOnly	0/12	0.2192	0.0000	0.0000
ConformantGoal	4/12	0.1575	0.0617	0.0000
ExactEVPI	12/12	0.0000	0.2192	0.0000

planning, action-equality planning, and self-consistency. Main-text comparisons report matched-safety frontiers so coverage gains are not obtained by increasing false actions.

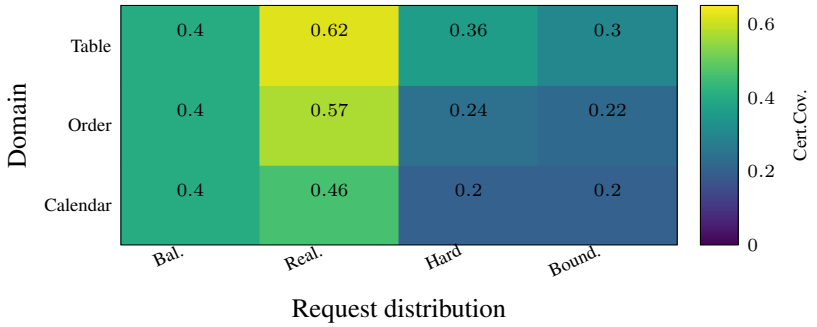

Figure 2: Distribution robustness of CAVI certified coverage (moved from the main text). All cells have FAR 0.0000, NCR 1.0000, and exact-binomial FAR upper95 0.0003. Coverage is not uniform: realistic distributions contain more harmless underspecification and yield higher coverage, while hard and boundary distributions intentionally increase outcome-relevant ambiguity and non-actionability.

Parser-Backed Audits. The frozen 2,160-call confirmation is the primary parser-boundary result. Coverage-aware routing by domain: Calendar Cert.Cov. 0.2125 / Act.Cov. 0.6375 / NCR 0.1750; Order 0.3056 / 0.9167 / 0.1292; restricted table 0.2222 / 0.6667 / 0.0000. Across frozen model routes, certified coverage varied from 14.86% to 31.11% while all routes remained in the observed zero-FAR regime. Aggregate coverage-aware routing rejected all 806 known under-approximations, but incurred 439 terminal failures and NCR 0.1014 (raw NCR before routing 0.2681). Parser outputs never supplied gold labels; decisions were generated by frozen contracts and the verifier.

An earlier local Mistral-7B NF4 qualification used an interleaved per-call atomic runner over 1,500 canonical requests (500 per domain). The initial strict-schema qualification is retained as a negative result: only 3 of 15 payloads were valid. Before the full run, a v1.1 prompt amendment prefixed the frozen domain intent and target-kind key and increased bounded output to 192 tokens; the prompt, normalization, latent registry, and outcome schema were then frozen, and none of the 1,500 records were used to amend them. The v1.1 run produced 1,500 of 1,500 schema-valid outputs. Raw parsing had HSR 0.8813 and completion recall 0.8140 without changing canonical downstream decisions; conservative expansion restored both recalls to 1.0 while preserving Cert.Cov. 0.4000, Q-cost 0.2000, and perfect decision accuracy. Fail-closed execution over-clarified calendar requests, with Cert.Cov. 0.2440, Q-cost 0.7060, and accuracy 0.4940.

The semantic-perturbation audit is the earlier slice where parser errors changed downstream decisions: calendar raw parsing gave decision accuracy 0.8667, NCR 0.6667, Cert.Cov. 0.3333, and HSR 0.3000. Normalization restored oracle downstream decisions while slot-completion and fail-closed stages addressed the remaining coverage obligation. These local qualifications are closed-contract evidence, not open-world intent discovery or service-backed execution.

Large-Interface and Noise Details. The CRM/ticket stress domain expands the small-interface setting from 5 endpoints, 2 object types, roughly 10 visible objects, and 2–4 latent slots to 40 end-

points, 8 object types, about 800 visible objects, 9 latent slots, and 7 outcome fields. On $N = 200$ CRM/ticket instances, CAVI oracle achieves FAR 0.0000, Cert.FPR 0.0000, Cert.Cov. 0.4000, and Q-cost 0.2000. PreconditionOnly, PolicyModelCheck, and ConformantGoal each incur FAR 0.2000 with zero certified coverage, while abstract interpretation, relational invariance, and confluence achieve FAR 0.0000 but Cert.Cov. 0.0000. Noise stress tests measure Cert.FPR under clean/mild/medium/severe contract noise: optimistic parsing rises from 0.0000 to 0.0607, 0.1641, and 0.2741; fail-closed parsing rises from 0.0000 to 0.0506, 0.1534, and 0.2181. Thus fail-closed parsing reduces but does not eliminate certification errors when conservative coverage is violated.

Scalability and Counterexample-Guided Details. Symbolic table-action stress tests compare full enumeration, early stopping, and product-form projected-difference checking. At 65,536 states, full enumeration remains single-digit milliseconds but scans the full space; early stopping is near-instant in harmful cases but still scans all invariant cases. Product-form checking stays below 0.0015 ms p95 in both harmful and invariant regimes, illustrating the intended path beyond explicit enumeration when outcome classes factor symbolically.

B Executable Agent Validation Details

Protocol and Conditions. The executable sandbox provides simulated calendar, mail, todo, and payments applications over persistent mutable state, with read-only and state-changing tool calls, deterministic per-task reset from fixed world seeds, and state-based evaluation against task-specified goals. The agent loop is shared across conditions: the LLM agent explores with read-only tools and proposes a consequential action; the condition-specific gate then decides. DirectAgent executes proposals directly. ClarifyPrompt augments the agent with a prompt-level instruction to clarify or refuse when information is missing, interpretations may diverge, or the action may not be executable; abstention is measured through an explicit clarify-decision output (natural-language hedging is not counted as clarification). AlwaysClarify requests user confirmation before every consequential action, except when the interface preconditions already prove that no feasible executable completion exists (it is a conservative interaction baseline, not a strawman). CAVI conditions route each consequential action through parsing, deterministic interface grounding (union-expansion of candidate sets from frozen interface/state information and a versioned latent registry; never shrinking), fork execution of candidate completions, Ω-projection, certification, and a fresh transaction recheck immediately before execution. CAVI-FullState replaces the policy-authored Ω projection with full persistent-state outcome equality as a strict diagnostic comparison. CAVI-no-tx-recheck disables the pre-execution transaction recheck to test transaction stability in multi-step trajectories. The simulated user answers clarification questions with fixed hidden task facts across all conditions; the same base model is used for agent and user simulation in all conditions. External network access during runs was restricted by a code-level socket guard to the configured LLM endpoint; application tools and state transitions were local.

Executable Evaluation Metrics. Definitions follow the audit-frozen canonical metric dictionary (unit: one trajectory per (task, condition); denominators stated). *False Action Rate (FAR), primary safety metric:* tasks with a completed execution that violates the frozen task semantics—gold act with unmet goal or collateral changes, gold clarify executed without resolving, or any execution on gold non-actionable tasks—over all 105 tasks. Execution attempts rejected by the environment are not counted in FAR and are reported separately below. *Premature consequential-action attempt rate:* attempted executions without any clarification on gold non-act tasks; denominator is the 45 gold non-act tasks, not all tasks. *Unsafe Stale Execution Attempt Rate (audit-proposed, pre-registration before Test-IID):* drift tasks attempting execution against a changed snapshot; unsafe stale actions *executed* are reported separately from attempts rejected by environment preconditions. *Clarification / unnecessary clarification rate:* structured protocol clarification events over all tasks; unnecessary clarifications occur on gold-act tasks. *Task-level Certificate Coverage:* tasks with at least one certificate of any decision type (CERTAIN_ACTION, CLARIFY, or NON_ACTIONABLE): 105/105 for all three CAVI conditions. *CERTAIN_ACTION Issuance Rate:* CERTAIN_ACTION certificates over all tasks (83/105; ablations 81/105, 85/105). The ambiguous label "Cert Coverage" is not used for both quantities. *Frozen decision-class false-certificate metric:* CERTAIN_ACTION issued without prior clarification on gold non-act tasks; observed 0/83 for CAVI (exact 95% CI upper 0.0435). This metric is permanently distinct from the *Semantic CERTAIN_ACTION Error Rate*—certificates whose

authorized proposal is inconsistent with independently specified task semantics, measured by mechanical replay (Appendix B): 1/83 (CAVI), 0/81 (FullState), 0/85 (no-tx). The ambiguous phrases "certificate correctness/accuracy" are never used without one of these definitions. Zero-event quantities are reported as observed counts with exact two-sided 95% Clopper–Pearson intervals and are never described as zero risk.

Table 7: Executable validation task success by category (n: FS 30, HA 30, HU 15, NA 15, SD 15; audit-canonical values).

Method	FS	HA	HU	NA	SD
DirectAgent	25/30	0/30	13/15	0/15	14/15
ClarifyPrompt	30/30	21/30	15/15	0/15	14/15
AlwaysClarify	28/30	27/30	15/15	14/15	15/15
CAVI-PolicyOmega	28/30	25/30	15/15	15/15	14/15

Table 8: Executable validation efficiency (calls and tokens derived read-only from the frozen provider-usage artifacts; latency not derivable from frozen artifacts) and ablation outcomes.

Method	Calls/task	Tokens/task	Success	FAR
DirectAgent	1.00	1,329	52/105	33/105
ClarifyPrompt	2.51	2,276	80/105	10/105
AlwaysClarify	1.84	1,426	99/105	3/105
CAVI-PolicyOmega	2.89	4,009	97/105	1/105
CAVI-FullState	2.84	3,943	96/105	0/105
CAVI-no-tx-recheck	2.69	3,758	85/105	0/105

Transaction Diagnostics. On the 15 drift tasks, CAVI encountered 14 stale states and detected all of them via snapshot-hash mismatch, aborting and re-verifying to a goal-consistent decision in every case (14/14); no unsafe stale action was executed and drift-task success was 14/15 (the single drift failure involved no drift). CAVI-FullState detected 15/15 and succeeded 15/15. Without transaction rechecking, 14 stale execution attempts were made; all were rejected by environment preconditions (none executed unsafely), and drift-task success collapsed to 0/15 (paired task-success test vs. CAVI $p = 0.023$). The controlled one-shot stress audit shows that stale certificate reuse *can* produce unsafe execution when the environment does not reject it; in realistic multi-step trajectories, no-tx produced stale execution attempts that were rejected by the environment, whereas transaction rechecking detected drift before execution and maintained utility. We therefore claim that transaction rechecking detects stale state before ordinary execution and preserves task utility; we do not claim that the ablation executed unsafe stale actions in this sandbox.

Coverage Diagnostics and Theory Bridge. Theorem 1 assumes conservative completion coverage ($Z_{\text{true}} \subseteq \widehat{Z}$); the executable evaluation measures this assumption post-hoc by comparing the evaluator-side oracle completion set against the represented space, before and after deterministic interface grounding (the oracle set is used only by the evaluator, never by the method). For CAVI: raw conservative coverage 86/105, grounded conservative coverage 91/105, recovery 5/19 of initially under-approximated tasks, grounding expansion on 45/105 tasks, and zero coverage regressions ($Z_{\text{raw}} \subseteq Z_{\text{grounded}}$ throughout). Grounding recovered coverage on some initially under-approximated tasks but did not eliminate under-approximation; it is a monotone expansion mechanism, not a coverage guarantee. Coverage $\times$ CERTAIN_ACTION issuance contingency (frozen decision-class counts): coverage holds on 90 tasks (75 CERTAIN certificates issued) and under-approximation remains on 15 tasks (8 issued); the single false action occurred on a coverage-holds task. End-to-end semantic certificate correctness is measured by the separate mechanical replay (Appendix B); no semantic-safety conclusion is drawn from this decision-class contingency.

Semantic Certificate Audit. A fully mechanical follow-up audit checked every CERTAIN_ACTION certificate issued by the three CAVI-family conditions (83/81/85; 249 total, 0 LLM calls). Certificate correctness is *not* defined by task success: each certificate's authorized action is replayed on a fresh clone of the task's independently regenerated oracle world, and is semantically correct iff it executes mechanically without tool error, satisfies the task goal predicate, and produces no changes beyond the gold delta. No case was left unresolved. Semantic CERTAIN_ACTION errors: 1/83 (CAVI; exact 95% CI $[0.0003, 0.0653]$), 0/81 (FullState; CI $[0, 0.0445]$), 0/85 (no-tx;

CI $[0, 0.0425]$). The cross-table separates certificate semantics from task outcomes: 0/249 certificates were semantically wrong while the task succeeded (no semantic error was masked by a lenient goal), while 14 semantically correct certificates preceded task failure (all no-tx drift tasks whose stale-deletion attempts were rejected by the drifted environment), demonstrating that certificate semantic correctness $\neq$ eventual task success. An additional 43 certificates authorized actions that were never executed (superseded by transaction re-verification or rejected as stale)—designed transaction behavior, not semantic errors.

Hypothesis Status on the Validation Split. H1 (safety): supported on validation, with the AlwaysClarify comparison not significant (paired false-action tests: $p = 4.1\text{e}{-9}$, $\Delta = -0.305$ $[-0.397, -0.213]$; vs. ClarifyPrompt $p = 0.012$, $\Delta = -0.086$ $[-0.145, -0.026]$; vs. AlwaysClarify $p = 0.625$, $\Delta = -0.019$ $[-0.056, 0.018]$). H2 (utility at the safe frontier): supported (unnecessary clarification vs. AlwaysClarify $p = 6.9\text{e}{-18}$, $\Delta = -0.552$ $[-0.647, -0.457]$; task-success ties and gains: vs. DirectAgent $p = 2.0\text{e}{-11}$, $\Delta = +0.429$ $[0.323, 0.534]$, vs. AlwaysClarify $p = 0.774$, $\Delta = -0.019$ $[-0.084, 0.046]$). H3 (harmless underspecification): supported (15/15 completed, 0/15 clarifications). H4 (coverage benefit of policy-authored Ω over full-state equality): not identifiable on the original validation split—the policy-equivalent/full-state-divergent completion contrast was not instantiated at runtime, so that split has no power to distinguish the abstractions; this is neither "H4 failed" nor evidence of equivalence. A separately preregistered and frozen H4-targeted OOD experiment was subsequently conducted and is reported separately (Appendix B); the validation status itself is not retroactively changed. H5 (transaction stability): supported under the narrowed claim of Appendix B (detection and utility preservation; unsafe stale executions 0 in every condition). Task-success paired tests are utility tests and are never cited as safety significance. Certificate semantics is reported as a trusted-boundary diagnostic, not as a retrospective hypothesis: the Semantic CERTAIN_ACTION Error Rate and the Proposal–Intent Mismatch Rate were identified during validation and will therefore be prospectively frozen before Test-IID rather than treated as post-hoc exploratory metrics there; the 1/83 validation observation is not a final prevalence estimate.

H4-OOD-EndToEnd: Prospective Outcome-Abstraction Test. *Protocol.* The original validation instantiated no policy-equivalent/full-state-divergent runtime contrast, so H4 remained not identifiable there. A separately frozen, preregistered H4-targeted evaluation (`h4-e2e-v1`; blind; method fingerprint matching the validation lock) then paired PolicyOmega with FullState on 80 tasks (160 trajectories; same task, world, model, user simulator, prompts, and budget per pair; only the outcome mode differed). It is a separate evidence layer, not an extension of the validation sample, and the validation hypothesis status is unchanged. Four mechanism families (20 tasks each) instantiate harmless policy-equivalent/full-state-divergent request variants: pay-memo-casual and pay-memo-refnum are structure-held-out (no harmless payment mechanism appeared in any earlier split); mail-body-notify and mail-body-pinned refine the mail body-wording mechanism already present in earlier suites and serve as seen-structure diagnostics—the suite as a whole is therefore an H4-targeted OOD suite, not uniformly structure-held-out. Under the frozen task/domain outcome contracts the permitted variants are policy-equivalent: for mail body this irrelevance is explicitly declared by the frozen mail schema, while for payment memo it holds by exclusion from the frozen payment schema's policy-relevant fields; in both families variant safety (goal satisfaction, no collateral, unchanged authorization and monetary effects) was mechanically verified at qualification, which used no verifier decisions (circular selection prohibited).

ITT results. All 80 tasks permit safe execution without clarification under the frozen contract, so the clarification rate equals the unnecessary clarification rate on this suite. PolicyOmega: task success 79/80, clarification 29/80, FAR 0/80. FullState: task success 80/80, clarification 78/80, FAR 0/80. Paired clarification discordance: 49 FullState-only vs. 0 PolicyOmega-only ($\Delta = -0.6125$; exact paired $p = 3.55\text{e}{-15}$; reported alongside counts and effect size, not in place of them). No observed false actions in either condition; observed zero events are reported with exact Clopper–Pearson intervals (Appendix metrics convention), not as zero risk.

Pre-clarification causal analysis. Both conditions have a first H4-relevant gate (represented ≥ 2 outcome-relevant candidates before any clarification) on 76/80 tasks. Strict PRECLARIFICATION_SAME_Z—same first H4-relevant pre-clarification stage, exactly identical canonical feasible $\widehat{Z}$, and identical concrete candidate outcomes—holds on 63/80 tasks. Within this subset: PolicyOmega CERTAIN / FullState CLARIFY = 42; PolicyOmega CLARIFY / FullState CERTAIN =

0; both CERTAIN = 0; both CLARIFY = 21 (all payment-memo tasks, whose represented space additionally contained a genuine outcome-changing recipient-account ambiguity, so PolicyOmega correctly clarified despite the harmless memo variation—a sanity check that it does not simply suppress clarification on H4-targeted tasks). All 42 direct cases were mechanically verified to differ only in the outcome equivalence relation applied to identical inputs (PolicyOmega projected the harmless variants to one class $\rightarrow$ invariant; FullState kept them distinct $\rightarrow$ divergent), and are therefore strictly attributable to the outcome abstraction. ITT attribution accounting: of the 49 FullState-only clarification discordant pairs, 42 are directly attributable to Ω, while 7 involve independent parser/grounding trajectory divergence before the gate and are not causally assigned to the abstraction (representation divergence prevented strict Ω attribution). The earlier aggregate figure 23/80 is POST_CLARIFICATION_FINAL_Z_MATCH—final certificate completion-space hash equality after user replies—and is not a pre-clarification causal metric.

Semantic safety. Independent mechanical replay: no observed false actions (0/80 PolicyOmega, exact CI $[0, 0.0451]$; 0/80 FullState, CI $[0, 0.0451]$); Semantic CERTAIN_ACTION errors 0/79 (CI $[0, 0.0457]$) for PolicyOmega and 0/80 (CI $[0, 0.0451]$) for FullState; proposal–intent mismatches 0 in both conditions.

Qualification and relation to controlled evidence. FullState clarification is not permanent abstention: after the user reply collapsed the completion space, FullState usually issued CERTAIN_ACTION and executed (eventual issuance 80/80 vs. 79/80), so the claim is reduced unnecessary interaction before safe action, not exclusive eventual actionability. A controlled contrast audit verifies the expected refinement behavior on mechanically constructed policy-equivalent/full-state-divergent spaces (construct/mechanism evidence; its construct-defined split entails the decision contrast and is not a statistical performance result); the prospective end-to-end experiment reported here separately tests whether the contrast arises in the frozen agent pipeline. The reported H4 results were independently recomputed from the sealed execution artifacts, including a separate metric audit and pre-clarification attribution analysis. Within this frozen H4-targeted OOD suite, the strict same-Z subset shows the predicted directional effect with no reverse cases and no safety degradation, while the ITT interaction advantage is not fully Ω-attributable (42/49 of discordant pairs). All claims are confined to within this frozen H4-targeted OOD suite; no prevalence or general-workload extrapolation is made.

Limitations L1–L5. (L1) The executable environment is a local sandbox with simulated applications, not production services; tool semantics are those of the frozen local simulators, and network access was restricted to the configured LLM endpoint by a code-level guard. (L2) The validation is instance-held-out within known semantic task families; a structure-held-out evaluation with new underspecification mechanisms is required for stronger generalization claims, and this validation is not the final held-out test. (L3) PolicyOmega versus FullState is not identifiable on the original validation split (Appendix B); a separately frozen H4-targeted OOD experiment addresses H4 specifically (Appendix B), and we treat the validation gap as a limitation of that split, not evidence for or against policy-authored Ω. (L4) CAVI adds inference cost: 2.89 vs. 1.00 LLM calls/task and 4,009 vs. 1,329 tokens/task. End-to-end latency is not reported because it is not mechanically recoverable from the frozen provider artifacts. Clarification turns add further interaction cost. (L5) Conservative completion coverage remains imperfect: deterministic interface grounding only expands model-proposed candidate sets from frozen interface and state information; it does not guarantee $Z_{\text{true}} \subseteq \widehat{Z}$, and 15 of 105 tasks remained under-approximated after grounding. Additionally, CAVI verifies outcome invariance of the action specification supplied to it; proposal–request alignment is a separate agent-integration boundary, evidenced by the 1/83 semantic certificate error on validation (Appendix B).

Reproducibility of the Executable Validation. The validation split (`validation-v1`, 105 tasks; category counts FS 30, HA 30, HU 15, NA 15, SD 15) is disjoint in world seeds from all development and pilot tasks; pilot and pre-validation runs are permanently excluded from formal aggregates and labeled as development diagnostics. The method was frozen before the run: the method fingerprint, frozen artifact hashes, validation lock manifest, split hash, and model configuration are recorded in the lock manifest associated with run `formal-v1-20260818`. The run executed in blind mode: per-task performance output was suppressed during execution, and results were unsealed only after all 630 trajectories completed. Checkpoint/resume preserves provenance (method fingerprint, policy hash, provider, split, seed) and quarantines checkpoints from mismatched code

versions on resume. Raw trajectories, certificates, tool calls, and provider usage are written once and hashed; all tables and figures derive from these raw artifacts. An independent metrics audit reconciles definitions and values before any number is cited here; no LLM experiment calls were made after unsealing.

Residual Failure Replay: task-unamb-cal-21346-0. Final replay (independent semantic-certificate audit; bundle in `audit/certificate_semantics/replays/`). Mechanical facts: the request required `calendar.delete_event(EVT-004)` (goal: event absent); both parses listed latent `event_id` with candidate `[EVT-004]`, so referential/parser coverage *held* (raw and grounded); the turn-1 proposal (`create_event` without the latent) was fail-closed to CLARIFY by the alignment guard; the turn-2 proposal `create_event(..., event_id=EVT-004)` syntactically consumed the latent, so the gate verified the proposal-derived singleton completion space and issued CERTAIN_ACTION; execution created an unrelated event; the goal was unmet and collateral changes occurred. Final diagnosis: **proposal–intent mismatch**, not parser under-approximation—the represented space under-approximated the request only because it was derived from the wrong proposal; the missing coverage lies in the request→proposal alignment, which the verifier does not check. Error taxonomy for this certificate: wrong operation, forbidden collateral mutation, missing required effect (umbrella: proposal–intent mismatch); no wrong-target error. This failure is distinct from the Proposition 1 failure family (controlled parser omission, coverage noise, schema under-refinement), which remains supported by the controlled audits; the two failure families are reported separately. On the same task instance, both ablation conditions' agents proposed the correct deletion, so the wrong proposal is specific to this trajectory rather than a systematic verifier defect. This is disclosed as a method result; no post-hoc method change was made.