RESEARCH · AUTONOMOUS AGENTS

Task Completion as an Internal, Grounded Judgment

Deciding an autonomous agent is done from its own doubt, fresh re-observation, and independent evidence

Stefan Ragland

Dominion Labs Research & Development

Dominion Labs

research@dmnlabs.org

October 14, 2025

Abstract

Autonomous agents that carry out real-world tasks face a deceptively simple question: *is the task actually done?* The common answer — trust a success flag, a self-report, or an acceptance checker — conflates *the agent said it succeeded* with *the world is now in the intended state*. We argue that completion should instead be an internal, grounded judgment: the agent is done only when it *believes*, past its own calibrated doubt, that the task's goal holds, where that belief is anchored to the goal's intent and formed from the agent's own reasoning and a fresh re-observation of the world. We separate two epistemically distinct kinds of evidence — *intervention* evidence ("I acted and the runtime reported an effect," DID) and *observational* evidence ("I independently re-measured the world and saw the intended state," SAW) — and combine them under an independence rule so that correlated confirmations of a single observation cannot masquerade as independent proof. Completion is decided by a posterior crossing a doubt-scaled acceptance bar, which structurally forces a fresh, independent observation before any goal is accepted. A single confirmation is insufficient (posterior ≈0.72); independent action-plus-observation is accepted (≈0.975); a fabricated success collapses (≈0.30). Treating an *absence* goal requires giving the observation channel goal-polarity; we report a defect-and-fix that moved an absence goal from a false incompletion (≈0.01) to correct completion (≈0.975). Across a fixed completion-judgment suite the mechanism produced *zero false completions*, including against a tool that reports success while writing nothing.

1 “Said done” is not “is done”

When an autonomous agent finishes acting, something must decide the task is complete. The default is an external verdict handed to the agent — a validator, a success flag, an acceptance-criteria checker — or, worse, the agent's own unaudited declaration. Both collapse a critical distinction: whether the *execution ran* versus whether the *goal now holds in the world*. An action can run to completion and return success while producing nothing: a blocked write, a copy from a missing source, a silently failing actuator. Any completion signal derived from “the code path finished” therefore measures the wrong thing.

The reframing we defend is that completion is a *judgment about the state of the world*, and must be grounded in evidence about that state rather than in the fact that a procedure returned. We make that judgment a first-class, per-task belief that has to survive the agent's own doubt before the task is closed.

> *A task is complete when the agent itself believes — past its own doubt — that the goal holds, where that belief is anchored to the goal's intent (so the agent cannot merely declare itself done) and formed from its own reasoning, fresh re-observation, and independent evidence.*

2 Two kinds of evidence: DID and SAW

After acting, an agent can ask two epistemically different questions. *Did my intervention appear to produce the effect?* — answered by the action runtime's own report of what it did (call this **DID**). And *is the world now actually in the state I intended?* — answered by an independent, fresh re-measurement that does not re-use the action's cached result or any verdict derived from it (call this **SAW**). These are not redundant; they fail independently. A tool can truthfully report that it wrote a file (DID positive) while the file is absent (SAW negative); a tool can report failure (DID negative) while the intended state already holds. Treating the action's self-report as though it were an observation of the world is the core error, and completion must consult both channels while weighting fresh observation as a distinct source.

3 Independent versus correlated evidence

Even when an agent gathers “multiple confirmations,” they are often the *same* observation flowing through different pipes: three tools that each read the same file are one measurement, not three. Counting correlated confirmations as independent manufactures overconfidence. A principled mechanism carries each datum's provenance and lets only genuinely independent pathways compound: evidence merely *derived* from other evidence (an aggregate “verified” verdict) is not counted at all, and correlated confirmations collapse to their strongest member. The practical consequence is visible in the calibration below: three same-source

reads leave the posterior at roughly 0.72 — not enough — whereas one action report plus one *independent* observation crosses the bar.

4 Completion as a belief that must pass its own doubt

We model completion as a loop over a per-task belief in the objective proposition G ("the goal holds"):

1. **Anchor.** From the task's intent, form G — the state conditions of a state goal, "the question is answered" for a knowledge task, or "the intended artifact exists" for a build task. The anchor is objective, so the agent cannot be done without G being the real target.
2. **Mint low.** The belief starts at a low prior (0.15), so a single positive report cannot read as done.
3. **Gather two channels.** Collect DID (the runtime's report) and SAW (a fresh, independent re-observation that does not consume the action's cached result).
4. **Independence check.** Drop derived aggregates; collapse correlated confirmations; compound only independent pathways.
5. **Intent polarity.** Score each datum against the goal's intended state, read from the operation (§5).
6. **Decide by doubt.** Update the posterior; accept iff it clears the agent's standing confidence bar (0.95, raised toward 0.99 by caution). There is no bespoke completion threshold — completion reuses the same confidence band the agent applies to any belief, and that high bar is what forces the fresh independent observation.

When the belief will not rise, the *reason* routes the response: missing evidence → try another method; an unreachable anchor → the plan was wrong; a blocker outside the agent → escalate. Each decision persists as experience, so the next similar task forms its judgment faster.

5 Absence goals and intent polarity

A large, legitimate class of tasks defines success as an *absence*: remove the stale lock, ensure the cache is gone, the file no longer exists. A mechanism whose observation channel is hard-wired to check for *presence* structurally cannot confirm these — it reports *not done* on a goal that in fact holds. This is a conservative failure (it never over-claims) but it silently blocks all cleanup and teardown work. The fix is to give every piece of completion evidence the *polarity of the goal's intent*, read from the operation, so that "the target is gone" scores as support for a removal, and an action that "failed" only because the target was already absent counts as intent achieved.

6 Results

All measurements below are of the mechanism's own behavior on a fixed world; see §7 for an important scope limitation.

6.1 One confirmation is not enough

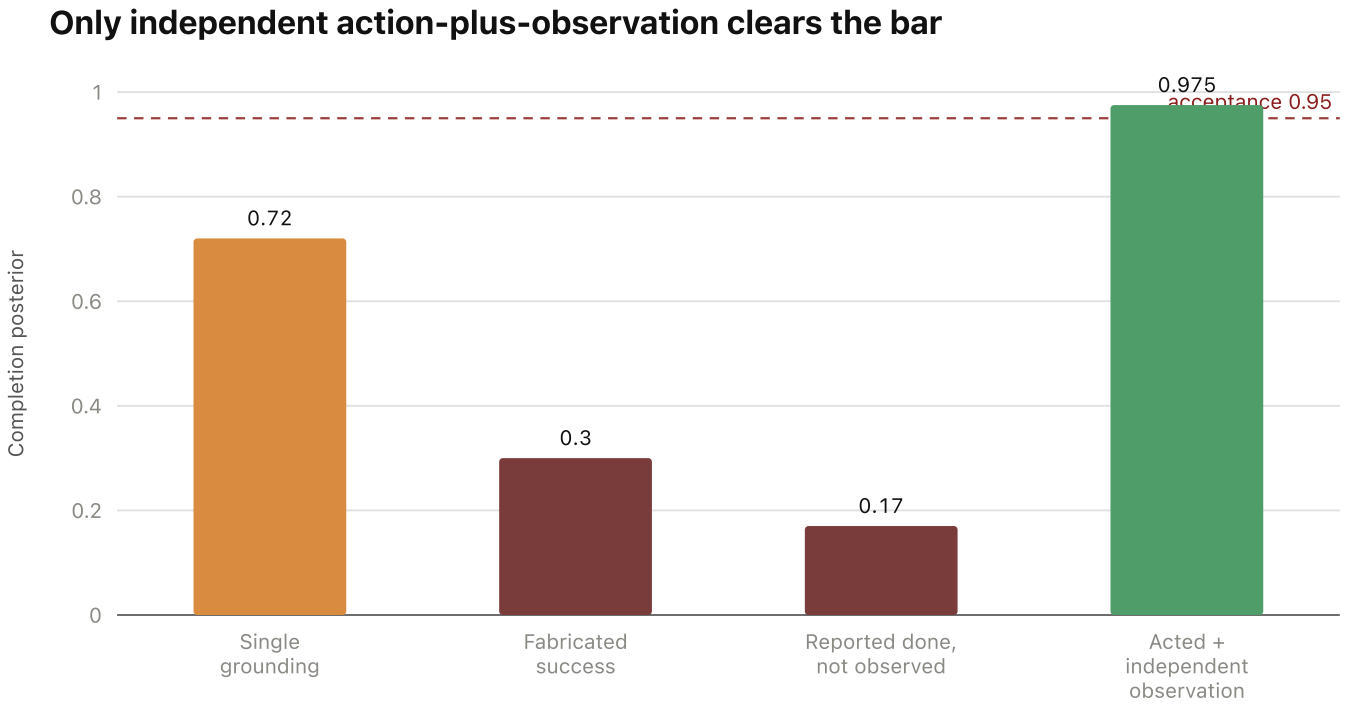

Figure 1. Completion posterior for four evidence scenarios, against the acceptance bar (0.95). Three same-source reads collapse to one grounding (≈0.72); a fabricated bare success (≈0.30) and a case where the tool reports success but re-observation finds nothing (≈0.17) stay well below the bar; only an action report plus an *independent* observation (≈0.975) is accepted.

6.2 A completion-judgment suite

On a fixed suite of nine tasks — five honest actions, one absence goal, and three “traps” where a tool reports success without changing the world — the mechanism's posterior separates the two classes cleanly and matches ground truth on all nine.

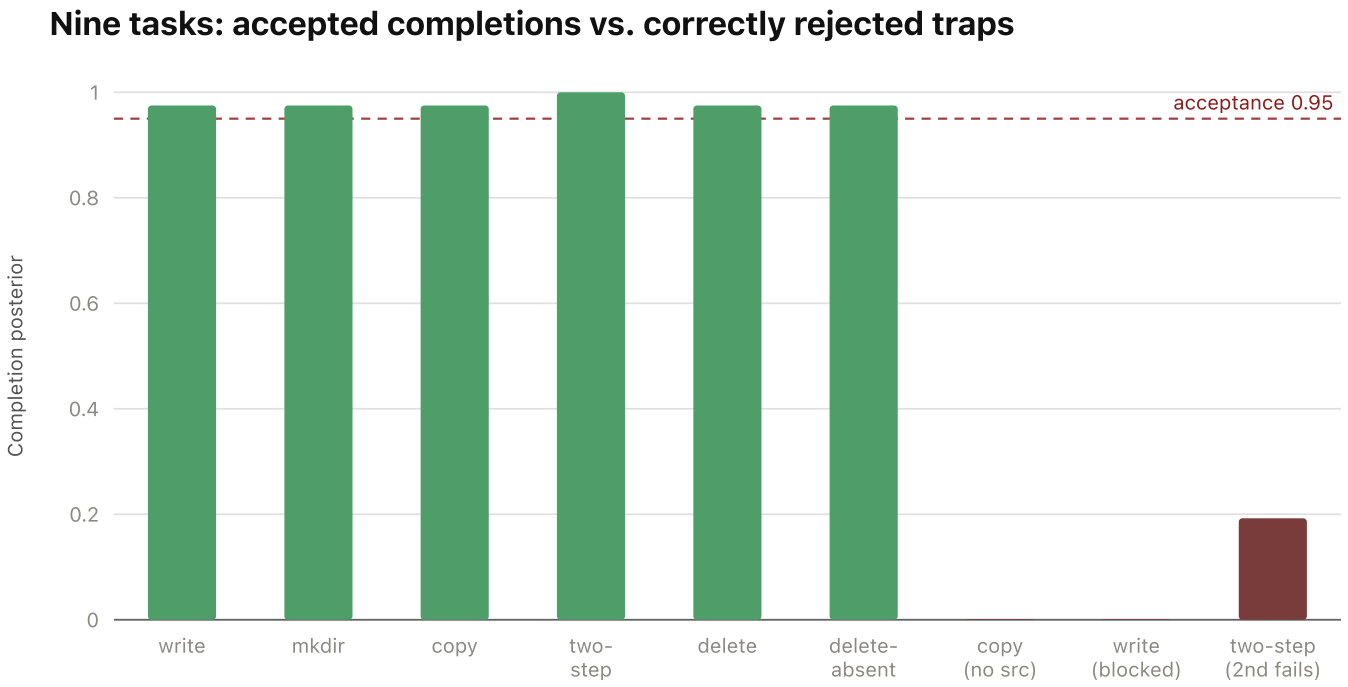

Figure 2. Per-task completion posterior. Green: tasks whose goal truly holds (accepted, ≈0.975–0.9999; the two-step task reaches the highest value from four independent groundings). Red: traps whose goal does not hold (rejected, ≈0.001–0.19). The dashed line is the acceptance bar. Summary: 6 correct completions, 3 correct rejections, *0 false completions, 0 false incompletions*.

Task	Kind	Posterior	Verdict
Copy from a missing source	trap	0.0011	not done ✓
Write to a blocked path	trap	0.0011	not done ✓
Two-step, second step fails	trap	0.192	not done ✓
Delete a file already absent	absence	0.975	done ✓

Table 1. The three traps and the absence edge case, with the posterior and the verdict. In every trap the action's self-report is positive or partially positive; independent re-observation is what produces the honest "not done."

6.3 The absence-goal fix

The absence case exposed a real defect. With a presence-only observation channel, "ensure the file is gone" scored as a false incompletion. Giving the observation channel goal-polarity, and letting the action's real error (rather than a discarded generic one) inform the intervention evidence, moved the posterior from ≈0.01, through a partial fix at ≈0.17, to ≈0.975.

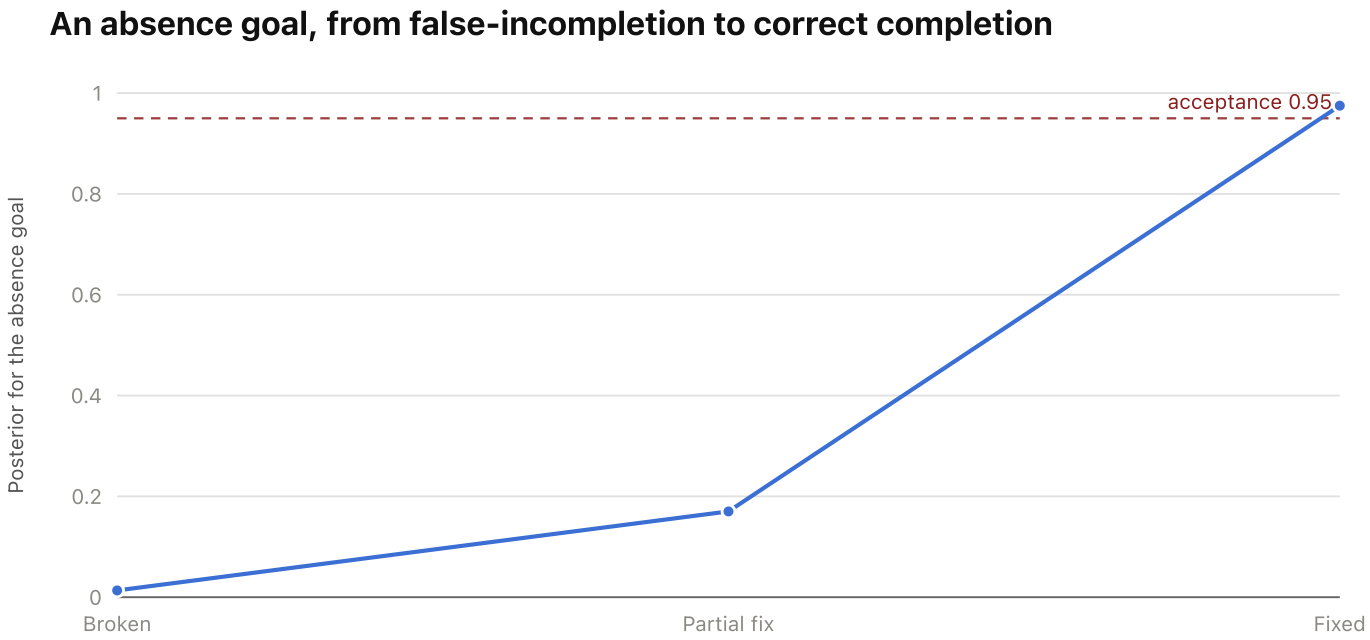

Figure 3. Posterior for the "delete a file that is already absent" goal across the defect and its two fix stages. Run-level accuracy on the suite rose from 87.5% (7/8, one false incompletion) to 100% (9/9); false incompletions fell from 1 to 0.

The load-bearing result is structural rather than comparative: across every suite, including a "phantom actuator" tool that reports success while writing nothing, the mechanism recorded *zero false completions*, because its observation channel (a raw existence check) is independent of the tool it acted through. A completion cannot be accepted on the strength of the same channel that performed — or faked — the action.

7 Limitations

An earlier internal evaluation compared this mechanism against a capable local language-model agent. We *withdraw* that comparison: the baseline had been handed the mechanism's own re-observation discipline and had task steps executed for it, so the comparison is contaminated and is not evidence about relative capability. We therefore make no claim of superiority over any model, and report only the mechanism's own behavior on a fixed world. The zero-false-completion result is demonstrated on a fixed filesystem suite; broader worlds, adversarial tools engineered to defeat the observation channel, and noisy or partial observations remain future work. The independence check enforced here is provenance-level (does this evidence share a causal lineage with that evidence); stronger statistical and adversarial notions of independence are open.

8 Conclusion

Completion should be earned, not declared. By making it a per-task belief anchored to the goal's intent, formed from independent action and observation evidence, and accepted only past the agent's own doubt, an agent stops confusing “my code returned” with “the world changed.” The same discipline that keeps a single echo from reading as proof also lets the agent confirm an absence, know *why* it is not yet done, and turn each completion decision into experience.

References

[1] G. Ryle. *The Concept of Mind*. Hutchinson, 1949. (Knowing-that vs. knowing-how.)

[2] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[3] R. E. Fikes, N. J. Nilsson. STRIPS: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3–4), 1971.

[4] L. Sha. Using simplicity to control complexity. *IEEE Software*, 18(4), 2001.

[5] R. El-Yaniv, Y. Wiener. On the foundations of noise-free selective classification. *JMLR*, 11, 2010.