Reckoning: Belief Arbitration for Autonomous Forward Deployment into Air-Gapped Estates

Abstract

An agent sent to deploy software into an air-gapped estate is in the position of a navigator without a fix. It carries a frozen prior, the knowledge baked into its weights at training time, which is already stale and which it cannot refresh: there is no network, no registry, no documentation server, no vendor, and no human to ask. It can learn only by intervening on the estate in front of it, and every intervention costs something. We argue that this setting is governed not by whether the agent's actions can be contained but by how it arbitrates between a prior that may be confidently wrong and local evidence that is fresh but scarce, and we show that the obvious approach fails in a way that is both provable and expensive. We prove that a Bayesian agent updating from a confident frozen prior needs a number of probes that grows linearly in the prior's confidence before it will believe what the estate is telling it, and we measure exactly this: at prior log-odds 128 the agent spends 73 probes to correct a single wrong belief. Our architecture, Reckoning, replaces the prior with an ε-contaminated mixture whose weight is not a tuning constant but an estimate the agent forms of its own staleness, computed from the rate at which its training-time knowledge is contradicted elsewhere in the same estate. This caps the prior's log-odds at $\log(2/\varepsilon - 1)$ regardless of how certain training made it, reducing that same correction to a single probe, and it transfers: staleness learned on components the agent has probed sets the base rate for components it has not. The estimator recovers the true divergence rate closely (0.30 measured as 0.290, 0.70 as 0.694) and localizes it, assigning contamination 0.910 to divergent components and 0.037 to sound ones. Across estates from fully sound to 70% divergent, Reckoning matches the better of the two fixed strategies at every point, which neither fixed strategy does: it ties the prior-trusting baseline exactly on a sound estate (error 0.0000) and beats the prior-discarding baseline on a badly diverged one (0.0099 against 0.0110), while standard Bayesian updating degrades to 0.1102 and catastrophic errors fall from 369 to 23. Selecting probes by expected free energy, epistemic value weighted by mission stakes, cuts catastrophic error 4.3 times against random probing at equal budget. A halt rule that escalates rather than acts on weak belief drives catastrophic error down 11.5 times, and over a hundred-step deployment it raises the probability of finishing with no wrong action from 0.019 under standard updating to 0.918. The ordering survives estates that break the estimator's assumptions, correlated divergence, unknown asymmetric noise, and hidden divergence, and because the contamination cap makes the method nearly insensitive to the prior's assumed confidence (mission error moves 0.006 as that confidence ranges sixtyfold, against 0.172 for standard updating), the one quantity a real model cannot report well barely matters. Against estates whose components lie by echoing the agent's own prior back at it, the most damaging attack available, a provenance check that tests observations for impossibly high self-consistency detects every compromised source across every compromise fraction, noise level, and threshold we swept, and cuts catastrophic errors from 150 to 3. All results are executed and the harness ships with the paper.

1 Introduction

The engineer who flies to a customer site, opens a laptop in a room with no outbound network, and makes the software work is doing something that has resisted automation for a specific reason. The difficulty is not that the deployment steps are hard. It is that nobody, including the customer, knows what is actually in the estate, and the engineer's job is to find out fast enough to act. They arrive with a great deal of general knowledge and almost no specific knowledge, and they convert one into the other by poking at things.

Automating that role means building an agent that can do the same, and the constraints of the setting are severe. In a genuinely air-gapped estate, whether that is a classified enclave, a factory floor, a ship, or a bank's core network, there is no package registry, no documentation, no search, no telemetry going home, and no colleague to ask. Frontier practice in AI for operations does not currently reach this far: contemporary systems correlate incidents, summarize them, and propose remediations, but they assume connectivity, assume a known stack, and assume a human approves the change. Every one of those assumptions fails here.

What makes the problem interesting is what the agent is left with. It has exactly two sources of belief. The first is its frozen prior: everything training put in its weights about how software behaves, which is broad, cheap to consult, and stale by construction, since the weights were fixed long before this estate existed in its current form. The second is local evidence: what it learns by intervening on the estate, which is fresh and authoritative but costs time, risk, and sometimes disruption to acquire. The entire competence of a forward-deployed engineer lies in trading these off well, and the entire failure mode of a naive agent lies in trading them off badly.

This is a navigation problem, and the analogy is exact enough to be useful. A navigator with no satellite fix and no landmarks does dead reckoning: they take their last known position, apply measured increments of heading and speed, and carry forward an estimate whose uncertainty grows with every step. Dead reckoning works, and it is how ships crossed oceans for centuries, but it has one characteristic failure. Error accumulates silently, and without an external fix there is nothing to correct it. An agent deploying into an air gap is reckoning in precisely this sense. Its frozen prior is the last known position, its probes are the measured increments, and the thing that will sink it is accumulated drift that nothing external ever corrects.

Framing the problem this way puts a specific question at the center, and it is one we have not seen posed. *How should an agent arbitrate between a stale prior and local evidence, when it cannot know in advance how stale the prior is?* The standard answer, Bayesian updating, turns out to be the wrong one, and not by a little. Bayesian updating is built for a prior that is uncertain, not for one that is confidently wrong, and a frozen model's prior about an estate it has never seen is very often exactly that: highly confident and simply incorrect, because the component in front of it was patched, forked, downgraded, or written after the training cutoff. We show in Section 3 that the number of probes a Bayesian agent must spend before local evidence overcomes such a prior grows linearly in that prior's confidence, and we measure the cost: at prior log-odds 128, correcting one wrong belief takes 73 probes. In an environment where probes are the scarce resource, that is the difference between a deployment that finishes and one that does not.

Our response, and the contribution of this paper, is an architecture we call Reckoning, built around a component that we believe is genuinely new: the agent maintains an explicit posterior over *whether its own training-time knowledge applies here*, estimates it empirically from the rate at which that knowledge is contradicted elsewhere in the same estate, and uses it as the contamination weight in a robust prior. The mechanism is classical, ε-contamination from robust Bayesian analysis [1, 2], but the use is not: the contamination weight is normally a modelling choice expressing how much the analyst distrusts an elicited prior, whereas here it is a quantity the agent measures about itself, in the field, from the estate it is standing in. This is what lets a stale prior be overridden in one probe instead of seventy-three, and it is also what lets a *sound* prior be kept and used, since an agent that has checked and found its knowledge holding has no reason to discard it.

The rest of the architecture follows from taking the navigation analogy seriously. The agent takes a fix by choosing interventions that maximize expected free energy, epistemic value weighted by what the mission actually depends on. It holds a heading by compiling intent rather than instructions, in the spirit of mission command doctrine, which exists precisely because subordinates lose contact with headquarters. And it knows when its error ellipse is too large to proceed, halting and escalating rather than acting on a belief it cannot support, which is the only defense available against the silent drift that kills dead-reckoned navigation.

Our contributions are:

- **A formulation of autonomous forward deployment as belief arbitration** between a frozen prior and costly local evidence (Section 2), which we argue is the binding constraint in air-gapped, human-absent deployment.
- **A correction-time result** (Section 3) showing that standard Bayesian updating requires probe count linear in prior confidence, while an ε-contaminated prior requires $\log(2/\varepsilon - 1)/\ell$ probes independent of that confidence, confirmed by execution at 73 probes against 1.
- **Self-estimated staleness** (Section 4): an applicability posterior computed by empirical Bayes from the agent's own contradiction rate, which recovers the true divergence rate, localizes it per component, and transfers to components never probed.
- **An executed evaluation** (Sections 5 and 6) covering mission error against divergence, probe selection, halting, horizon decay, and estates that actively deceive the agent by echoing its own prior.
- **An artifact** containing the estate simulator, the policies, and every experiment, so the numbers can be reproduced and attacked.

2 The setting

We fix a model that is deliberately spare, because the arguments should not depend on the details of any particular estate.

Definition 1 (Estate)**.** *An estate is a set of C components, each carrying K binary behavioral facts. Fact (c, k) has a true value $y_{ck} \in \{0, 1\}$ that the agent cannot observe directly. Examples of such facts are whether a restart preserves in-flight state, whether configuration lives at the documented path, and whether an endpoint speaks the protocol version its banner claims.*

Definition 2 (Frozen prior)**.** *The agent carries a prediction $\hat{y}_{ck}$ for every fact, derived from training-time knowledge of the component's type and version, together with a confidence expressed as prior log-odds $L > 0$. A component is* sound *if $y_{ck} = \hat{y}_{ck}$ for all k and* divergent *otherwise. The fraction of divergent components is ρ, and the agent does not know ρ or which components are divergent.*

Divergence is the formal stand-in for everything that makes a real estate surprising: local patches, vendor forks, deliberate downgrades, undocumented configuration, and components that postdate the agent's training entirely. The essential and often overlooked point is that the prior does not become *uncertain* on divergent components. It stays confident and becomes wrong, which is a different and more dangerous condition.

Definition 3 (Probe)**.** *A probe is an intervention on a component that returns a noisy observation of one fact, correct with probability* $1-\eta$ *for* $\eta < 1/2$*. Probes are the only source of local evidence, and the probe budget is the scarce resource: each one costs time and carries risk, which in an operational estate is a real constraint rather than an accounting convention.*

Definition 4 (Commitment)**.** *At the end of deployment the agent must commit for every mission-relevant fact, choosing a value to act on or abstaining and escalating through whatever high-latency channel exists. Acting on a wrong value is an error. A designated subset of facts is* catastrophic*, where a wrong action is unrecoverable, which models the difference between a service that fails to start and a database that loses data.*

The agent's problem is now stated: allocate a probe budget across an unknown estate, arbitrate between prior and evidence, and commit, minimizing error and especially catastrophic error. There is no oracle, no reward signal during deployment, and no opportunity to check the answer. Section 3 shows why the textbook approach to the arbitration step is the wrong one.

3 Why standard updating fails

The natural first design is a Bayesian agent: start from the frozen prior, update on probes, act on the posterior. It is principled, it is what the textbook prescribes, and in this setting it is quietly disastrous. The reason is that Bayesian updating treats prior confidence as information, and a frozen model's confidence about an estate it has never seen is not information. It is an artifact of training.

Proposition 1 (Correction time)**.** *Let the frozen prior assign log-odds* $L > 0$ *to the wrong value of a fact, and let each probe return an independent observation that is correct with probability* $1-\eta$*,* $\eta < 1/2$*, carrying log-likelihood ratio* $\ell = \log\big((1-\eta)/\eta\big)$ *in favour of the truth. Then:*

(a) Under standard Bayesian updating the posterior favours the truth only after $n > L/\ell$ *probes.*

(b) Under an ε*-contaminated prior* $\pi = (1-\varepsilon)\pi_0 + \varepsilon\,\mathrm{Unif}$*, the prior contributes log-odds of magnitude at most* $\log(2/\varepsilon - 1)$ *for* any L*, so the posterior favours the truth after* $n > \log(2/\varepsilon - 1)/\ell$ *probes, a bound independent of* L*.*

prior L	Bayes n	L/ℓ	Reckoning n	cap$/\ell$
2	1	0.9	1	0.8
4	2	1.8	1	0.8
8	4	3.6	1	0.8
16	10	7.3	1	0.8
32	19	14.6	1	0.8
64	36	29.1	1	0.8
128	73	58.3	1	0.8

Table 1: Median probes to correct a confidently wrong belief, at $\eta = 0.1$ so $\ell = 2.197$ nats, on an estate with $\rho = 0.30$. The agent has probed the rest of the estate but never this component, so its prior is discounted only by the estate-wide staleness it inferred elsewhere. Bayesian correction time grows linearly in prior confidence; the contaminated bound does not depend on it.

Proof. (a) Each probe contributes at most ℓ to the posterior log-odds in favour of the truth and the prior contributes $-L$, so the posterior log-odds after n probes is at most $-L + n\ell$, which is positive only when $n > L/\ell$. (b) Under contamination the probability assigned to the wrong value is at most $(1-\varepsilon)\cdot 1 + \varepsilon \cdot \frac{1}{2} = 1 - \varepsilon/2$, and the probability assigned to the true value is at least $\varepsilon/2$. The prior log-odds magnitude is therefore at most $\log\big((1-\varepsilon/2)/(\varepsilon/2)\big) = \log(2/\varepsilon - 1)$, and substituting this for L in (a) gives the bound. □

Corollary 1. *The ratio of probe costs between the two schemes is* $L/\log(2/\varepsilon - 1)$*, which grows without bound in the prior's confidence. A more confidently trained model is, under standard updating, strictly more expensive to correct.*

That last observation is the uncomfortable one. Everything the field does to make models more confident and more knowledgeable makes this failure mode worse, because confidence enters the correction cost linearly while the estate's willingness to contradict the model does not change at all. An agent that is very sure how a database behaves will spend a great many probes discovering that this particular database was forked three years ago.

Measured. We confirmed both parts on the simulator. Table 1 reports the median number of probes needed to correct a fact on a divergent component. Standard Bayesian correction time tracks the predicted L/ℓ and grows linearly, reaching 73 probes at $L = 128$, while Reckoning holds at a single probe across the whole sweep. The measured Bayesian figures sit slightly above the analytic prediction because the prediction assumes every probe is correct, whereas at $\eta = 0.1$ roughly one probe in ten pushes the wrong way and must itself be overcome. The spread is tight around these medians: the interquartile range is 4 to 6 probes at $L = 8$, 17 to 21 at $L = 32$, and 69 to 77 at $L = 128$, so the linear growth is a property of the distribution and not of a heavy tail.

The complementary sweep confirms what the cap does

depend on. Holding L fixed at 64 and varying the estate's true divergence rate ρ from 0.05 to 0.5, the agent's estimated staleness tracks the truth (0.044, 0.089, 0.186, 0.280, 0.494 against 0.05, 0.1, 0.2, 0.3, 0.5), the predicted cap falls from 1.7 to 0.5 probes as the estate proves less trustworthy, and measured correction time moves from 2 probes to 1. Standard updating sits at 36 probes throughout, indifferent to everything the estate has been trying to tell it.

4 The Reckoning architecture

Reckoning has four stages, named for the navigation they imitate. The contribution is concentrated in the second.

Fix: choosing what to probe. The agent selects interventions by expected free energy, taking the epistemic value of a probe to be the expected reduction in Bernoulli entropy of the belief it targets, and weighting that by extrinsic value, meaning how much the mission depends on the fact and whether a wrong action there is catastrophic. The decomposition into epistemic and extrinsic value is the one active inference makes standard [8], and the pure-information limit is the classical information-gain objective that also underlies empowerment-style exploration [9]. The practical effect is that the agent does not survey the estate evenly. It spends its budget where uncertainty and stakes coincide.

Arbitrate: estimating its own staleness. This is the core. Rather than treating the frozen prior as a fixed input, the agent maintains a posterior over whether that prior applies at all, per component. Let component c have n_c probes of which d_c contradicted the prior's prediction. If the prior applies, contradictions arrive at the observation-noise rate η. If it does not, they arrive at a substantially higher rate q, for which the agent assumes a generic value of 0.5 rather than any estate-specific constant it could not know. The applicability posterior follows immediately,

$$\log \frac{\Pr[\text{stale}_c]}{\Pr[\text{applies}_c]} = \log \frac{\hat{\rho}}{1-\hat{\rho}} + d_c \log \frac{q}{\eta} + (n_c - d_c) \log \frac{1-q}{1-\eta},$$

with the estate-wide rate $\hat{\rho}$ estimated by empirical Bayes over the components probed so far. This is exactly the data-dependent selection of a contamination weight from an ε-contamination class that robust Bayesian analysis formalized [1], applied to a question that analysis was never pointed at: not how much the analyst distrusts an elicited prior, but how much a deployed agent should distrust itself.

Two consequences matter. First, the resulting contaminated prior saturates at $\log(2/\varepsilon - 1)$, which is the mechanism behind Proposition 1(b): no amount of training confidence can make the agent stubborn. Second, and less obviously, $\hat{\rho}$ supplies the base rate for components the agent has *never probed*. Staleness discovered in one corner of the estate propagates to the rest, so an agent that finds three forked services immediately becomes appropriately skeptical about everything else it has not yet checked. That transfer is what makes the estimate worth computing at all, since the probe budget never covers the whole estate.

true ρ	estimated $\hat{\rho}$	ε on divergent	ε on sound
0.00	0.006	n/a	0.015
0.10	0.095	0.864	0.025
0.20	0.197	0.885	0.037
0.30	0.290	0.910	0.037
0.50	0.477	0.912	0.058
0.70	0.694	0.933	0.100

Table 2: Self-estimated staleness after 600 probes, averaged over 60 estates. The agent recovers how stale it is and localizes that staleness to the right components rather than hedging uniformly.

Table 2 shows the estimator is well calibrated and, importantly, discriminating. It recovers the true divergence rate across the range, and it separates: contamination lands near 0.9 on components that really are divergent and below 0.1 on components that are sound. An agent using a single global hedge would get the average right and the individual decisions wrong.

Heading: intent rather than instructions. Because there is nobody to ask, the agent cannot be given a procedure; it must be given an objective and the latitude to pursue it. This is not a new idea, it is mission command, which armies adopted for exactly this reason: subordinates lose contact with headquarters, and a force that stops when the radio fails is worthless. Doctrine expresses it as mission orders that specify intent while leaving execution to the subordinate's judgement [11]. The computational analogue is goal-driven autonomy, which supplies the machinery we adopt: form expectations, detect discrepancies between expectation and observation, explain them, and reformulate goals rather than halting [12, 13]. In Reckoning the applicability posterior is what makes discrepancy detection meaningful, since a discrepancy is only informative once the agent can tell the difference between an estate that surprised it and knowledge that was stale to begin with.

Halt: knowing the error ellipse is too large. Dead reckoning fails silently, so the last stage is a rule for refusing to act. The agent commits only when its posterior confidence on a fact exceeds a threshold, and otherwise abstains and escalates through whatever high-latency channel exists, a courier or a scheduled window. This is selective prediction [18] applied at the level of deployment actions, and it is the only structural defense against accumulated drift, because it is the one place the agent can convert uncertainty into inaction instead of into a wrong

ρ	prior-only	evidence-only	std-bayes	reckoning
0.0	0.0000	0.0101	0.0000	**0.0000**
0.1	0.0564	0.0108	0.0015	**0.0001**
0.2	0.1167	0.0108	0.0045	**0.0010**
0.3	0.1786	0.0109	0.0124	**0.0027**
0.5	0.3013	0.0106	0.0443	**0.0059**
0.7	0.4237	0.0110	0.1102	**0.0099**
catastrophic errors at $\rho = 0.7$				
	2564	36	369	**23**

Table 3: Mission error rate against estate divergence, 200 estates per cell at a 400-probe budget. Reckoning matches the better fixed strategy at every divergence rate, tying the prior-trusting policy on a sound estate and beating the prior-discarding policy on a diverged one.

commitment. Section 5 measures the trade it makes.

5 Evaluation

All results run on the simulator of Section 2 with 40 components, 6 facts each, observation noise $\eta = 0.1$, and prior log-odds $L = 8$ unless stated. We compare four arbitration policies: *prior-only*, which trusts the frozen corpus and never probes; *evidence-only*, which discards the corpus and believes only measurements; *standard-bayes*, which updates from the confident prior; and *reckoning*.

Mission error against divergence. Table 3 is the central systems result. The two fixed strategies are each right somewhere and badly wrong elsewhere, which is what one would expect: trusting a frozen corpus is optimal when the corpus happens to be accurate and catastrophic when it is not, and discarding it is robust but wastes the budget rediscovering things the agent already knew. Standard Bayesian updating interpolates between them and inherits the bad end, degrading to an error rate of 0.1102 and 369 catastrophic errors on a badly diverged estate.

Reckoning matches the better of the two fixed strategies at every divergence rate, which neither fixed strategy does. On a fully sound estate it ties the prior-trusting baseline exactly at 0.0000, because it checks, finds its knowledge holding, and uses it; the evidence-only policy pays 0.0101 for its refusal to. On a 70% divergent estate it reaches 0.0099, slightly better than the evidence-only baseline's 0.0110 and an order of magnitude better than standard updating. Catastrophic errors fall from 369 to 23 at that same point. The agent is not splitting the difference between two strategies, it is identifying which regime it is in.

Probe selection. Table 4 isolates the Fix stage by holding arbitration fixed and varying only how probes are chosen. Expected free energy dominates throughout the budget-constrained regime, and the gap is largest exactly where it matters. At a 100-probe budget it cuts mission error from 0.1709 under random probing to 0.0851, and catastrophic error from 0.1731 to 0.0402, a factor of 4.3. Mission-blind information gain sits in between, which is the point: knowing where you are uncertain is worth something, and knowing where uncertainty is expensive is worth considerably more. By 800 probes the estate is saturated and every targeted rule converges, which simply says that probe selection matters when the budget binds, which in an air gap it always does.

budget	random	sweep	info-gain	free energy
100	0.1731	0.1576	0.1406	**0.0402**
200	0.1298	0.0948	0.0670	**0.0107**
400	0.0781	0.0535	0.0107	**0.0013**
800	0.0303	0.0166	0.0002	**0.0002**

Table 4: Catastrophic error rate against probe budget under four selection rules, 200 estates per cell at $\rho = 0.35$. Weighting information gain by mission stakes is worth a factor of 4.3 against random probing at the tightest budget.

threshold	catastrophic	cat. rate	coverage
none	23	0.00255	1.0000
0.9	21	0.00233	0.9993
0.99	13	0.00144	0.7757
0.999	2	0.00022	0.2135
0.9999	0	0.00000	0.0138

Table 5: The halt rule across 300 estates at $\rho = 0.35$ and a 300-probe budget. Catastrophic error and coverage trade monotonically; the operator chooses the point.

Halting. Table 5 measures the trade the Halt stage makes. Raising the commitment threshold monotonically reduces catastrophic error and monotonically reduces coverage, which is the expected shape of any selective-prediction rule, and the useful observation is how favourable the exchange rate is at the top end. Moving the threshold to 0.999 cuts catastrophic error 11.5 times, from a rate of 0.00255 to 0.00022, and a threshold of 0.9999 eliminates catastrophic error entirely in 300 estates. The cost is coverage, which falls to 0.2135 and 0.0138 respectively, meaning the agent escalates most of the mission rather than acting on it. Whether that is a good trade is a deployment decision and not a technical one, and the architecture's job is to expose the dial rather than to choose for the operator. What it should not do is act at full coverage on beliefs it cannot support, which is what every policy without a halt rule does by construction.

Horizon. A forward deployment is not one decision, it is hundreds in sequence, and the empirical literature on long-horizon agency is blunt about what that does.

H	prior-only	std-bayes	reckoning	+halt
1	0.7915	0.9613	0.9929	**0.9991**
5	0.3107	0.8210	0.9652	**0.9957**
10	0.0965	0.6740	0.9317	**0.9915**
20	0.0093	0.4543	0.8680	**0.9830**
50	0.0000	0.1391	0.7020	**0.9580**
100	0.0000	0.0193	0.4927	**0.9179**

Table 6: Probability that a deployment of H sequential commitments completes with no wrong action, from measured per-commitment error rates over 400 estates. Per-step arbitration quality compounds exponentially in horizon length.

METR's time-horizon measurements find that the task length frontier models complete at 50% reliability has been doubling roughly every seven months, but that the horizon at 80% reliability is about five times shorter, and that performance drops considerably on less structured, messier tasks [21]. Forward deployment is the messy, high-reliability case, so per-step error rate is the quantity that decides whether a deployment finishes.

Table 6 makes the compounding explicit, reporting the probability that a deployment of H sequential commitments completes with no wrong action, computed from measured per-commitment error rates of 0.2085, 0.0387, 0.0071, and 0.0009 for the four policies. The separation is stark by any realistic horizon. At $H = 100$ the prior-trusting policy has essentially no chance of a clean run, standard Bayesian updating succeeds 1.9% of the time, Reckoning reaches 49.3%, and Reckoning with halting reaches 91.8%. The lesson is not that any policy is safe over long horizons but that per-step arbitration quality is amplified exponentially by horizon length, so improvements that look modest per decision are decisive per deployment.

6 Estates that lie

An air-gapped estate may be contested, and the agent's evidence channel is the obvious thing to attack. The relevant precedent is indirect prompt injection, where adversarial content reaches a model through the artifacts it reads rather than through its instructions [22, 23], and every source a forward deployer consults, banners, configuration files, logs, and responses to its own probes, is such an artifact.

We model the most damaging attack available rather than a generic one. A compromised component does not return noise, and it does not return the opposite of the truth. It returns *exactly what the agent's frozen prior predicts*, which is the strategy that best serves an adversary: it confirms stale knowledge, suppresses the contradiction signal that the applicability posterior depends on, and makes a divergent component look perfectly ordinary. This attack is aimed at the paper's own mechanism, and it works. With 20% of components compromised on a 40% divergent estate, mission error reaches 0.0220 and 150 catastrophic errors occur, because the very signal Reckoning uses to detect staleness has been switched off.

	no provenance	provenance
mission error rate	0.0220	**0.0003**
catastrophic errors	150	**3**
detection rate	n/a	1.000
false-flag rate	n/a	0.061

Table 7: Deceptive estates, 120 runs with 20% of components echoing the frozen prior on every probe, at $\rho = 0.4$. The consistency test detects sources that agree with themselves more often than observation noise permits.

The defense is a consistency test, and it works because the attacker's behaviour is too good. Honest probes of the same fact disagree with one another at rate $2\eta(1-\eta)$, which is 0.18 at $\eta = 0.1$, since each is independently noisy. A component that echoes the prior is perfectly self-consistent and disagrees with itself at rate zero. Internal inconsistency far *below* the honest expectation is therefore evidence of a compromised source, and the agent flags any component whose repeated probes agree more than noise allows, escalating everything from that source rather than acting on it. Table 7 reports the result: detection of every compromised component across 120 estates, a false-flag rate of 6.1% on honest components, mission error falling from 0.0220 to 0.0003, and catastrophic errors from 150 to 3.

We note the generality carefully, because it is limited. The test catches this attack because deception through perfect consistency is detectable in principle: an adversary that reproduces the correct noise statistics while still lying would defeat it, at the cost of having to know η and having to introduce genuine inconsistency into its own story. That is a real arms race and we have run one round of it, not the whole thing. What the round establishes is that the agent's evidence channel needs provenance reasoning at all, which is not obvious until one notices that the most effective attack on a staleness-aware agent is to tell it that its stale knowledge is correct. The defense is not a single lucky operating point: sweeping the compromise fraction from 5% to 50%, the noise rate from 0.05 to 0.20, and the consistency threshold across a four-fold range, detection remains 1.000 throughout, and the false-flag rate moves exactly as the mechanism predicts, rising with a looser threshold (0.026 to 0.262) and falling as noise grows (0.158 at $\eta = 0.05$ to 0.020 at $\eta = 0.20$, since an honest source's noisy inconsistency is easier to separate from a perfect echo when there is more of it).

7 Robustness to the estimator's assumptions

The simulator of Section 2 shares a functional form with the applicability estimator: divergence is drawn indepen-

estate	ρ	prior	evid.	bayes	reckoning
correlated	0.3	0.1850	0.0106	0.0218	**0.0027**
	0.7	0.4164	0.0107	0.1185	**0.0077**
asym. noise	0.3	0.1775	0.0502	0.0338	**0.0266**
	0.7	0.4243	0.0507	0.1701	**0.0438**
hidden div.	0.3	0.1775	0.0106	0.0102	**0.0024**
	0.7	0.4243	0.0100	0.1083	**0.0095**

Table 8: Mission error on estates that violate the estimator's assumptions, 150 estates per cell. Reckoning matches the better fixed strategy at every divergent rate under correlated divergence, unknown asymmetric heavy-tailed noise, and divergence hidden on rarely-probed facts.

dently per component, probe noise is symmetric with a known rate, and divergence shows up as contradiction on probed facts. A fair objection is that the estimator's good calibration is then a home-field result. We therefore re-ran the central comparison on three estates that each break one of these assumptions, and asked only whether Reckoning still matches the better of the two fixed strategies. It does at every divergent point, and the one place it does not strictly win is exactly the place it should not need to.

Correlated divergence. When divergence is drawn per component *type* rather than per component, so an entire class of components diverges together and the pooled base rate is a worse summary, Reckoning reaches error 0.0027 at $\rho = 0.3$ and 0.0077 at $\rho = 0.7$, against standard Bayes at 0.0218 and 0.1185. *Asymmetric heavy-tailed noise.* When the probe noise is asymmetric and heavy-tailed with rates the agent does not know, assuming a single symmetric $\eta = 0.1$ throughout, every policy pays for the unmodelled noise, but Reckoning still leads the fixed strategies at both divergent rates (0.0266 and 0.0438) and beats standard Bayes (0.0338 and 0.1701). This is also where the honest cost appears: on a fully sound estate the unknown asymmetry raises Reckoning's error to 0.0072, because noise the estimator cannot model genuinely misleads it. *Hidden divergence.* When components diverge on facts the agent rarely probes, Reckoning again matches the better fixed strategy at 0.0024 and 0.0095. Table 8 collects the results. The proposition of Section 3 is model-independent and needs none of this; what these runs establish is that the *systems* claim, matching the better fixed strategy across divergence, survives structure the estimator never assumed.

8 Where the prior confidence comes from

The whole argument turns on the prior log-odds L, and a deployed model does not hand one over: verbalized confidence is miscalibrated, typically overconfident after preference training. Two things make this less of a problem than it appears, and we measured both.

First, L is elicitable offline from data the agent can carry. Given a calibration estate whose facts are labelled from a prior deployment, the agent bins its raw stated confidence and maps each bin to the empirical log-odds of being correct in that bin, which is ordinary reliability calibration. On a simulated overconfident model that states 0.95 while being right 80% of the time, trusting the stated number gives $L = 2.94$, whereas the calibration recovers $L = 1.23$, close to the true 1.39. The point is not the specific numbers but that the elicitation uses held-out labels rather than the model's word for itself.

Second, and more importantly, the downstream method barely depends on getting L exactly right. Sweeping the assumed L from 1 to 64 at $\rho = 0.3$, Reckoning's mission error moves by 0.0058 in total, from 0.0076 to 0.0019, while standard Bayes swings by 0.1723 over the same range as its overconfidence runs away. This is the contamination cap doing its job: by bounding the prior's influence at $\log(2/\varepsilon - 1)$, Reckoning makes the hard-to-elicit quantity nearly irrelevant, which is the property one wants precisely because L cannot be known well. An agent that must guess its own confidence should use a method that does not much care what it guesses.

Test-time adaptation and memory staleness. A concurrent line adapts agents to unseen environments at deployment time, grounding syntax and dynamics by interaction [25] and benchmarking robustness under partial observability and non-stationarity [26, 27]; a parallel line handles staleness *within* a stored knowledge base, superseding facts by recency or key rather than by an agent's estimate of its own training-time reliability [28, 29]. Reckoning is adjacent to both and distinct from each: it adapts neither surface syntax nor a retrieval store but the *weight* placed on parametric prior against interventional evidence, and the quantity it estimates is the applicability of the model's own training, not the freshness of an external record. To our knowledge no prior or concurrent work casts this arbitration as a self-estimated contamination weight.

Self-managing systems. The direct ancestor of this work is autonomic computing, which proposed self-configuring and self-healing systems organized around a monitor, analyze, plan, execute loop over shared knowledge [3], and architecture-based self-adaptation, which added an explicit runtime model of the system being managed [4]. Reckoning is recognizably in this lineage and departs from it in one respect that turns out to be decisive: both assume the managed elements are known and the adaptation policies human-authored. Neither addresses an estate the system has never seen, and neither has any notion that the manager's own knowledge might be stale. That is the gap this paper occupies.

Acting under model uncertainty. The formal treatment of acting while still learning the environment is the Bayes-adaptive Markov decision process [5], approximated in practice by meta-reinforcement learning that carries task belief in a latent state [6, 7]. Active inference frames the same trade as minimization of expected free energy and supplies the decomposition into epistemic and extrinsic value that our Fix stage uses [8], with empowerment and information gain as the pure-epistemic limit [9]. These approaches assume a training distribution of related tasks and a prior that is uncertain rather than confidently wrong, which is precisely the assumption a frozen corpus violates.

Learning structure by intervening. A deploying agent can act on the estate, which makes interventional causal discovery the right formal frame for the Fix stage [10], and our probe selection is a budgeted version of active intervention design. We use only the simplest consequence of this literature, that intervention is more informative than observation, and a fuller treatment of dependency structure among components is future work.

Command without communication. Mission command exists because subordinates lose contact with headquarters, and its doctrinal expression is the mission order that fixes intent while leaving execution free [11]. The computational counterpart is goal-driven autonomy, which formalizes expectation, discrepancy detection, explanation, and goal reformulation [12, 13], on the substrate of belief-desire-intention agent architectures [14]. We adopt this machinery and contribute the observation that discrepancy detection is ill-posed without an applicability estimate, since an agent that cannot distinguish a surprising estate from stale knowledge cannot tell what its discrepancies mean.

Knowing what one does not know. That models are partly aware of their own reliability is established [15], as is the fact that this awareness degrades and that intrinsic self-correction without external feedback often makes reasoning worse [16], a result that matters here because in an air gap there is no external oracle to appeal to. Detection tooling that runs offline, including semantic entropy for confabulation detection [17] and conformal methods for abstention with finite-sample guarantees [18], is directly usable at the Halt stage. Whether verification is reliably easier than generation for language models is contested [19, 20], which is one reason our design leans on environmental ground truth, a probe that either responds or does not, rather than on the model's critique of itself. This is the end-to-end argument [24] applied to belief: the authoritative check on whether a component behaves as the agent thinks it does sits at the component, not in the agent's reasoning about it.

Long-horizon agency. Measured time horizons for autonomous task completion are short relative to a deployment, and considerably shorter at high reliability than at 50% [21]. Our horizon result is the same phenomenon viewed from the arbitration side: per-step error compounds, so the quantity to optimize is per-step belief quality.

Adversarial environmental content. Indirect prompt injection established that model inputs drawn from the environment are an attack surface [22], with measured attack success rates on tool-using agents [23]. Our deceptive-estate experiment applies the idea to the evidence channel specifically, where the optimal attack is not to inject instructions but to confirm the agent's stale beliefs.

9 Discussion and limitations

What the simulator does and does not establish. Every number in this paper comes from a synthetic estate, and the honest reading is that the simulator establishes the mechanism and not the magnitudes. Its structure encodes assumptions that a real estate would violate: facts are binary and conditionally independent given divergence, divergence is drawn independently per component, probe noise is symmetric with known rate, and the agent's mission is a set of independent commitments rather than a dependency-ordered plan. Section 7 tests the first three directly, on estates with correlated divergence, unknown asymmetric heavy-tailed noise, and hidden divergence, and the policy ordering survives all three, with the honest exception that unmodelled noise raises error on an otherwise-sound estate. What transfers with confidence is Proposition 1, which is a statement about log-odds arithmetic and holds regardless of estate structure. What remains untested is genuinely dependency-ordered deployment, where a wrong early commitment changes what later probes even mean, and that is the most important gap between this model and a real estate.

The estimator's assumptions. The applicability posterior needs a value for q, the contradiction rate on a divergent component, and we deliberately gave the agent a generic 0.5 rather than the simulator's true generative value, so it is not being handed a parameter it could not know in the field. It also assumes divergence is detectable through contradiction at all, which fails for a component that diverges only in ways the agent never probes, and it assumes the estate is homogeneous enough that a single base rate is meaningful, which fails for an estate that is half modern and half museum. A hierarchical version that estimates staleness per component class rather than per estate is the obvious next step and we have not built it.

No language model was harmed in this evaluation. The policies here are explicit Bayesian machinery, not a language model with a scratchpad, and that is a deliberate scoping choice rather than an oversight: we wanted the arbitration mechanism measured in isolation, without confounding it with the reasoning quality of any particular model. The cost is that the mapping from an actual model's parametric knowledge to a calibrated prior log-odds L must be supplied separately; Section 8 gives one offline procedure and, more to the point, shows the method is nearly insensitive to L across a sixtyfold range, so the mapping does not have to be precise. Wiring a real model in as the prior, with its stated confidences calibrated this way, is the natural next step and is not something this paper claims to have done.

Coverage is the real cost. The halting results are favourable on catastrophic error and unfavourable on coverage, and it would be misleading to present the first without the second. At the threshold that eliminates catastrophic error the agent escalates almost everything, which in an air gap means the deployment waits for a courier. An architecture that reduces catastrophic error by declining to deploy has not automated the forward-deployed engineer; it has automated the part of the job where the engineer sends an email. Closing that gap means raising per-step belief quality until high thresholds are affordable, which is the same quantity the horizon result identifies, and it is the direction we would push next.

On the framing. We have argued that this setting is governed by epistemics rather than by containment, and the argument is a claim about which constraint binds first, not a claim that containment is unimportant. An agent that arbitrates beliefs perfectly and has no notion of blast radius will still eventually do something irreversible and wrong. The two concerns are complementary, and the reason to separate them is that they have different remedies: containment is engineered into the substrate, whereas arbitration has to happen in the agent's beliefs, and no amount of the former substitutes for the latter when the question is what the estate actually is.

10 Conclusion

An agent deployed into an air-gapped estate navigates by dead reckoning. It carries a frozen prior it cannot refresh, it learns only by intervening, and nothing external ever corrects its drift. We showed that the textbook approach to the resulting arbitration problem fails in a specific and expensive way, requiring probe count linear in the prior's confidence before local evidence can overcome it, and that the fix is to stop treating training-time confidence as information: replace the prior with an ε-contaminated mixture whose weight the agent estimates about itself, from the rate at which its own knowledge is contradicted in the estate it is standing in. That single change caps the prior's influence at $\log(2/\varepsilon - 1)$ regardless of how certain training made it, cuts a 73-probe correction to one probe, and yields an agent that matches the prior-trusting strategy where the prior is sound and the prior-discarding strategy where it is not, which neither fixed strategy manages alone. Around that core, choosing probes by expected free energy cuts catastrophic error 4.3 times at a tight budget, a halt rule cuts it a further 11.5 times, together carrying a hundred-step deployment from a 1.9% chance of a clean run to 91.8%, and a provenance test defeats the attack aimed squarely at the mechanism, in which a compromised estate lies by agreeing with the agent's stale beliefs. The larger point is the one the navigation metaphor was chosen to carry: an agent far from home with no way to take a fix is not primarily in danger of doing something it should have been prevented from doing. It is in danger of being confidently, quietly wrong about where it is, and the remedy for that is not a better cage but a better estimate of its own error.

References

[1] J. Berger and L. M. Berliner. Robust Bayes and Empirical Bayes Analysis with ε-Contaminated Priors. *The Annals of Statistics*, 14(2):461-486, 1986.

[2] P. J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73-101, 1964.

[3] J. O. Kephart and D. M. Chess. The Vision of Autonomic Computing. *IEEE Computer*, 36(1):41-50, 2003.

[4] D. Garlan, S.-W. Cheng, A.-C. Huang, B. Schmerl, and P. Steenkiste. Rainbow: Architecture-Based Self-Adaptation with Reusable Infrastructure. *IEEE Computer*, 37(10):46-54, 2004.

[5] M. O. Duff. *Optimal Learning: Computational Procedures for Bayes-Adaptive Markov Decision Processes*. PhD thesis, University of Massachusetts Amherst, 2002.

[6] K. Rakelly, A. Zhou, D. Quillen, C. Finn, and S. Levine. Efficient Off-Policy Meta-Reinforcement Learning via Probabilistic Context Variables. In *ICML*, 2019. *arXiv:1903.08254*.

[7] L. Zintgraf, K. Shiarlis, M. Igl, S. Schulze, Y. Gal, K. Hofmann, and S. Whiteson. VariBAD: A Very Good Method for Bayes-Adaptive Deep RL via Meta-Learning. In *ICLR*, 2020. *arXiv:1910.08348*.

[8] K. Friston. The Free-Energy Principle: A Unified Brain Theory? *Nature Reviews Neuroscience*, 11(2):127-138, 2010.

[9] A. S. Klyubin, D. Polani, and C. L. Nehaniv. All Else Being Equal Be Empowered. In *European Conference on Artificial Life (ECAL)*, 2005.

[10] J. Peters, D. Janzing, and B. Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, 2017.

[11] Headquarters, Department of the Army. *ADP 6-0: Mission Command, Command and Control of Army Forces*. 2019.

[12] M. Molineaux, M. Klenk, and D. W. Aha. Goal-Driven Autonomy in a Navy Strategy Simulation. In *AAAI*, 2010.

[13] M. Klenk, M. Molineaux, and D. W. Aha. Goal-Driven Autonomy for Responding to Unexpected Events in Strategy Simulations. *Computational Intelligence*, 29(2):187-206, 2013.

[14] A. S. Rao and M. P. Georgeff. BDI Agents: From Theory to Practice. In *International Conference on Multi-Agent Systems (ICMAS)*, 1995.

[15] S. Kadavath et al. Language Models (Mostly) Know What They Know. *arXiv:2207.05221*, 2022.

[16] J. Huang, X. Chen, S. Mishra, H. S. Zheng, A. W. Yu, X. Song, and D. Zhou. Large Language Models Cannot Self-Correct Reasoning Yet. In *ICLR*, 2024. *arXiv:2310.01798*.

[17] S. Farquhar, J. Kossen, L. Kuhn, and Y. Gal. Detecting Hallucinations in Large Language Models Using Semantic Entropy. *Nature*, 630(8017):625-630, 2024.

[18] V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

[19] K. Valmeekam, M. Marquez, and S. Kambhampati. On the Self-Verification Limitations of Large Language Models on Reasoning and Planning Tasks. *arXiv:2402.08115*, 2024.

[20] H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe. Let's Verify Step by Step. In *ICLR*, 2024. *arXiv:2305.20050*.

[21] T. Kwa, B. West, J. Becker, et al. Measuring AI Ability to Complete Long Tasks. *arXiv:2503.14499*, 2025.

[22] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz. Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. In *ACM Workshop on Artificial Intelligence and Security (AISec)*, 2023. *arXiv:2302.12173*.

[23] Q. Zhan, Z. Liang, Z. Ying, and D. Kang. InjecAgent: Benchmarking Indirect Prompt Injections in Tool-Integrated Large Language Model Agents. *arXiv:2403.02691*, 2024.

[24] J. H. Saltzer, D. P. Reed, and D. D. Clark. End-to-End Arguments in System Design. *ACM Transactions on Computer Systems*, 2(4):277-288, 1984.

[25] Grounded Test-Time Adaptation for LLM Agents. *arXiv:2511.04847*, 2025.

[26] From Task Solving to Robust Real-World Adaptation in LLM Agents. *arXiv:2602.02760*, 2026.

[27] EvoArena: Tracking Memory Evolution for Robust LLM Agents in Dynamic Environments. *arXiv:2606.13681*, 2026.

[28] Temporal Validity in Retrieval Memory: Eliminating Stale-Fact Errors for AI Agents over Evolving Knowledge. *arXiv:2606.26511*, 2026.

[29] Supersede: Diagnosing and Training the Memory-Update Gap in LLM Agents. *arXiv:2606.27472*, 2026.