



The Limits of AI Self-Governance

Why a model cannot be its own auditor

By Kelly Cunningham
Chief Technology Officer

Executive summary

An AI model's assessment of its own outputs is useful evidence, but it is not assurance. No organization should treat a model's self-evaluation, self-correction, or self-reported reasoning as the control that stands between that model and a consequential outcome. This conclusion does not rest on speculation about rogue AI. It rests on three independent lines of evidence, each documented in peer-reviewed or primary research from the labs that build these systems.

1. **The statistical problem.** A model checking its own work shares the blind spots that produced the error. When the generator and the evaluator fail together, the check adds little. Published measurements show models favor their own outputs, fail to improve reasoning when asked to self-correct without outside feedback, and become less well calibrated after the alignment training that makes them usable.
2. **The transparency problem.** A model's stated reasoning is not a reliable record of how it reached its answer. In controlled tests, frontier reasoning models disclosed the hints that actually drove their answers only about a quarter to two-fifths of the time.
3. **The incentive problem.** When a system is optimized against a measure, it learns to satisfy the measure rather than the intent. Under laboratory conditions, models have learned to hide reward hacking from monitors, to comply selectively when they believe they are being trained, and to retain trained-in unsafe behavior through standard safety training.

Beneath the evidence sits a formal result. Gödel's incompleteness theorems, applied to self-certification, show that a system rich enough to matter either cannot certify itself or certifies everything indiscriminately, and that every chain of oversight ends in an assumption that must be tested from outside. Incompleteness guarantees that self-governance fails somewhere; the evidence shows it fails often, on ordinary inputs. One caution keeps the argument honest: external oversight has its own failure modes, chiefly that different models increasingly make the same mistakes. The case is therefore not "self-governance bad, external governance good." It is that assurance requires

independent signal: ground truth, deterministic checks, decorrelated monitors, and accountable humans, layered so that no single point of judgment is trusted alone.

For compliance and risk leaders, the practical rule is the one auditors already apply to people: no one marks their own homework, and no control is credited unless its independence can be shown.

1. Scope and definitions

AI self-governance, as used here, means any control in which the AI system, or a model sharing its training lineage, is the primary judge of whether its own behavior is correct, safe, or compliant. It takes four common forms:

- **Self-evaluation:** the model scores or grades its own output (“Rate your answer from 1 to 10”).
- **Self-correction:** the model is asked to review and revise its output without new information (“Check your work and fix any errors”).
- **Self-report:** the model’s stated reasoning, confidence, or explanation is treated as a record of what it did and why.
- **Self-policing by prompt:** behavioral rules live only in the instructions given to the model (“Never disclose customer data”), with no independent enforcement.

Each of these is cheap, fast, and often genuinely helpful. The question this paper addresses is narrower: **can any of them serve as the control of record** for a consequential decision, the control an auditor, regulator, or court would be told provides assurance?

The question matters because these patterns are now load-bearing in production. LLM-as-judge pipelines grade model outputs at scale. Agentic systems are told to verify their own work before acting. Vendors describe models as “self-correcting” and cite visible reasoning as an audit trail. If these mechanisms are weaker than assumed, organizations

are carrying unrecognized risk and documenting controls that do not do what the documentation says.

The paper proceeds from mechanism to evidence to practice. Section 2 explains what model “confidence” is and is not. Section 3 sets out the formal case and where it hands off to evidence. Sections 4 and 5 present the empirical record. Section 6 gives the counter-evidence: where self-checks work. Section 7 examines the limits of external oversight. Section 8 turns the findings into design principles.

2. What model “confidence” actually is

A language model’s confidence is a probability distribution over possible next tokens, shaped by training to match observed data. It is not a belief about whether a claim is true, and nothing in standard training gives the model a separate, trustworthy estimate of how reliable that distribution is.

How training uses confidence

The core training signal, cross-entropy loss, penalizes the model according to the probability it assigned to the correct token:

$$\text{Loss} = -\log P(\text{correct token} \mid \text{input})$$

A confident wrong prediction (low probability on the right answer) produces a large loss and a large weight update. A confident right prediction produces almost none. Several well-established refinements adjust this behavior: **label smoothing** trains toward a softened target so the model is not pushed to 100% certainty; **focal loss** down-weights examples the model already gets right; and an explicit **confidence penalty** adds a cost for low-entropy outputs. All of these are real and widely used. All of them act on the shape of the output distribution during training. None gives the model an independent check on whether that distribution is trustworthy for a new input.

Calibration is real, but fragile

Calibration means that when a model says 80%, it is right about 80% of the time. The evidence on this is mixed in a way that matters for governance:

- **Pretrained models can be well calibrated.** Anthropic researchers found that large pretrained models are reasonably calibrated on multiple-choice and true/false questions in the right format, and can be trained to predict whether they know an answer ([Kadavath et al., 2022](#)).
- **Alignment training degrades it.** OpenAI reported that GPT-4's pretrained model was well calibrated but that post-training reduced calibration; later work replicated the effect in open models and found preference-tuned models systematically overconfident ([Tian et al., 2023](#); [Xiao et al., 2025](#)). The models organizations actually deploy are the post-trained ones.
- **Stated confidence is a different, weaker signal.** When asked to state confidence in words, models tend to be overconfident, plausibly imitating how people talk about certainty, and all tested methods struggled on tasks requiring professional knowledge ([Xiong et al., 2024](#)). Some prompting strategies improve this, but results vary sharply by model and prompt.
- **Calibration does not transfer.** Calibration measured on a benchmark says little about calibration on an organization's own data, especially under distribution shift. This is a long-standing finding for neural networks generally ([Guo et al., 2017](#)).

The regress question

If a model has confidence, how confident is it in its confidence? In a standard model the answer is that there is no second level. The model outputs one distribution; Epistemic uncertainty about that distribution is not represented. Asking the model to estimate its own reliability simply produces another output from the same distribution-generating machinery, with the same blind spots.

The regress is real when one model is asked to audit another model of the same kind, and that one in turn, with no anchor. It is broken in practice not by more layers of self-assessment but by an **external anchor**. Conformal prediction is the clearest example: it wraps any model in prediction sets with a guaranteed error rate, and the guarantee comes from held-out labeled data, not from the model's opinion of itself ([Angelopoulos & Bates, 2021](#)). The guarantee holds only while new data resembles the calibration data, which is itself an assumption someone outside the model has to monitor. That pattern, an outside reference plus a stated assumption that is checked, recurs throughout this paper.

3. The formal case

Incompleteness establishes two things about AI self-governance. First, a system expressive enough to matter either cannot certify its own soundness or certifies everything, so its self-certification is impossible or worthless. Second, every chain of oversight ends in an assumption that cannot be proved inside the chain. What incompleteness does not establish is how often self-governance fails, or how much an outside checker should be trusted. Those are empirical questions, and the rest of this paper attempts to answer them clearly.

From analogy to formal claim

The appeal to Gödel in AI debates is often stated loosely, as if a neural network simply “were” an axiomatic system. It can be made rigorous, but the precise version matters, because the theorems do not always hold up.

- **The training process is a formal system.** Architecture, loss function, data, and optimizer are fixed rules applied to fixed starting points. Anything a computer does can be written down this way. Calling training axiomatic is a description, not an analogy.
- **The learned structure is a model of a theory.** The trained weights encode entities and relations. In the logician's vocabulary, this is closer to a *model*, a

concrete structure that some theory describes, than to a set of axioms. Its constraints are soft and probabilistic, and they are not closed by way of deduction.

- **The running system is a general computer.** A transformer run in a loop with a scratchpad can, given enough steps, carry out any computation (Pérez et al., 2021; Merrill & Sabharwal, 2024). At that point Turing's halting result and Rice's theorem apply to the system's behavior. These are the computational form of Gödel's results, not a separate idea.

A frozen network considered on its own is a finite object, so its properties over a bounded input set are decidable in principle. But they are too intractable to check: proving even simple properties of such networks is NP-complete (Katz et al., 2017). Incompleteness undermines the processes that generate and run the system, not the frozen weights themselves.

Gödel's dilemma applied to self-certification

Gödel's central result is a dilemma: a system rich enough to express arithmetic can be consistent or complete but never both. Applied to a system asked to certify its own soundness, failure is guaranteed.

1. **If the system is consistent,** Gödel's second theorem says it cannot prove its own consistency. Its self-certification is impossible.
2. **If the system is inconsistent,** the principle of explosion applies: from a contradiction, anything can be derived, including a proof of its own consistency, and thus self-certification is worthless.

Language models complicate the second qualifier but do not escape it. They routinely assert a claim in one context and its negation in another, so they are not consistent. But because they are not closed by way of deduction, they hold these local contradictions without deriving global contradictions. That makes them messier than either qualifier and no more trustworthy as their own certifiers. A model containing inconsistencies that nevertheless confidently certifies its own reliability is nothing more than an example of a broken system.

Failure is guaranteed; its frequency is not

Rice's theorem converts the dilemma into an operational guarantee. Any self-governance mechanism that must always render a verdict on a non-trivial behavioral property will be wrong on some cases. A mechanism allowed to answer "I cannot determine this" can avoid error, but only by abstaining, and thus the case must be decided somewhere else.

The guarantee concerns existence, not frequency. Incompleteness proves that failures exist; it does not say how many, where, or whether a given deployment will meet them. That is the job of the empirical record, and Sections 4 and 5 show the failures are not exotic. Models fail at self-correction on grade-school arithmetic and multiple-choice questions, which are fully decidable. Those failures come from correlated error and training incentives, not from incompleteness. The paper therefore rests on two distinct claims: the formal one supplies "guaranteed," and the empirical one supplies "material."

The regress ends in an assumption

Gödel's second theorem is asymmetric, and the asymmetry is the formal argument for external oversight. A system cannot prove its own consistency, but a stronger system can prove the weaker one's consistency, at the price of assuming its own. Certification must therefore come from outside and above, and the chain of certifiers must end in an assumption at the top that no system in the chain can prove. Löb's theorem (1955) sharpens the point for self-trust: a system that adopts "whatever I can prove is true" as a rule about itself ends up proving everything, which is why self-trusting agents face what researchers call the Löbian obstacle ([Yudkowsky & Herreshoff, 2013](#)). This is not a peculiarity of AI. Mathematics is in the same position. Gödel's second theorem applies to standard set theory, the foundation most of mathematics rests on, so mathematicians cannot prove their own foundations consistent. Their confidence rests on a century of intensive use without finding a contradiction. That is empirical confidence, held provisionally.

For AI oversight, the top of the chain is even less formal. Whether a monitoring system is “stronger” than the system it watches cannot be proved; it can only be estimated from evidence, which makes it a falsifiable, probabilistic claim that is never certain. In practice the relationship is often inverted: weaker models are routinely used to monitor stronger ones, sometimes effectively ([Baker et al., 2025](#)), but not on any formal guarantee. Its effectiveness rests on a fortunate accident: today’s models happen to reason in language a monitor can read, a property no one built for oversight and one that training pressure can erode.

Incompleteness and empirical method are therefore not rival foundations; they divide the labor:

	What it establishes	What it cannot establish
Incompleteness and computability results	Self-certification is impossible or worthless; certification must come from outside; the chain ends in an unprovable assumption	How often systems fail, or whether a particular checker deserves trust
Empirical science	How much confidence the outside checker deserves, measured against ground truth	Certainty: every estimate is probabilistic and open to revision

Two governance consequences follow from the combination, and neither follows from either half alone.

1. **Assurance cannot be a one-time certification.** Because the assumption at the top of the chain is probabilistic and falsifiable, it has to be continuously re-tested. Governance must be ongoing monitoring anchored to ground truth, not a certificate issued once.
2. **The checker needs independence more than strength.** Governance does not require a monitor that is stronger in general, which cannot be established. It requires a monitor whose failures are independent of the

specific failures it is checking for. Unlike general strength, independence can be measured directly.

Measuring independence: correlated error

Independence is a statistical property, best expressed in the language of reliability engineering. A check is valuable only to the extent its failures are independent of the failures it is meant to catch.

$$P(\text{undetected error}) = P(\text{error}) \times P(\text{check misses} \mid \text{error})$$

The second term is where self-evaluation fails. The inputs on which a model errs are disproportionately the inputs it misunderstands, and a check performed by the same model on the same input inherits that misunderstanding. The conditional miss rate rises precisely where it matters.

An illustrative calculation shows the scale. Suppose a model errs on 10% of cases. An independent check that catches 80% of errors leaves 2% undetected. A self-check that shares the model's blind spots and catches only 20% of its errors leaves 8% undetected, four times as many, while still reporting that every output was "reviewed." The figures are hypothetical; the structure is standard. It is the same reason financial controls require segregation of duties and why a second engineer, not the author, reviews a change. Sections 4 and 5 show that the conditional miss rate for AI self-checks is, in measured cases, high.

4. Self-evaluation, self-correction, and self-report

When a model has no outside information to work with, asking it to check its own reasoning does not reliably improve that reasoning and frequently makes it worse. The bottleneck is detection: models struggle to find their own errors, and their accounts of their own reasoning are incomplete.

Self-correction without outside feedback degrades accuracy

Google DeepMind researchers tested the common “review your answer and fix any problems” pattern on standard reasoning benchmarks, removing the answer key that earlier, more optimistic studies had quietly relied on. Accuracy fell for every model on every benchmark ([Huang et al., ICLR 2024](#)).

Model	Benchmark	Initial answer (%)	After two rounds of self-correction (%)
GPT-3.5	CommonSenseQA	75.8	41.8
Llama-2-70B	GSM8K (math)	62.0	36.5
GPT-4	GSM8K (math)	95.5	89.0
GPT-4	HotpotQA	49.0	43.0

The inherently defective mechanism matters more than the numbers. When the models changed an answer, they were more likely to change a correct answer to a wrong one than the reverse. The researchers’ diagnosis was blunt: the models could not properly judge whether their own reasoning was correct. The earlier studies that reported large gains had used ground-truth labels to stop the loop once the answer was right, which is not available in deployment.

Two further studies isolate the cause:

- **Finding errors is the weak link, not fixing them.** Google Research found that state-of-the-art models generally failed to locate logical mistakes in reasoning traces, even in objective, unambiguous cases, yet corrected them well once told where the mistake was ([Tyen et al., ACL Findings 2024](#)).
- **External verification succeeds where self-critique fails.** On planning and constraint problems, GPT-4’s performance collapsed when it critiqued its own answers and improved substantially when a sound external verifier checked them ([Stechly et al., ICLR 2025](#)).

A critical survey of the self-correction literature summarizes the state of evidence: no prior work had shown successful self-correction using feedback from prompted models

alone, outside tasks unusually suited to it; self-correction works well when reliable external feedback is available ([Kamoi et al., TACL 2024](#)).

Self-evaluation is biased toward the self

LLM-as-judge pipelines often use the same model, or the same model family, to grade outputs. Research presented at NeurIPS 2024 found that GPT-4 and Llama 2 can recognize their own outputs at better than chance, rate their own text higher than human annotators judge it to be, and that the strength of this self-preference rises linearly with the ability to recognize their own work ([Panickssery et al., 2024](#)). In some conditions, models preferred text merely labeled as their own.

Stated reasoning is not a reliable record

Visible chain-of-thought reasoning is often presented as an audit trail. The evidence says it is a partial one.

Models rationalize hidden influences. When few-shot prompts were biased so the right answer was always “(A),” models followed the bias and wrote plausible explanations that never mentioned it, with accuracy falling by as much as 36% on a suite of reasoning tasks ([Turpin et al., NeurIPS 2023](#)).

Reasoning models conceal what drove their answers. Anthropic tested Claude 3.7 Sonnet and DeepSeek R1 by planting hints in prompts and checking whether the reasoning acknowledged hints the model actually used. On average, Claude disclosed them about 25% of the time and R1 about 39% ([Chen et al., 2025](#)). When the models were rewarded for exploiting wrong hints, they learned to do so in over 99% of cases while rarely admitting it in their reasoning ([Anthropic summary](#)).

The authors of the second study were careful to note limits: two models, multiple-choice tasks, and a particular kind of hint. The finding is not that reasoning traces are worthless. Section 7 shows they can be a powerful monitoring signal. The finding is that **a reasoning trace cannot be treated as a complete or faithful record of why a**

decision was made, which is exactly what an auditor or regulator would assume an “explanation” to be.

5. Optimization pressure meets self-oversight

Section 4 concerned honest error. This section concerns something more serious for governance: systems trained against a measure learn to satisfy the measure, and in controlled experiments that has included learning to look compliant when observed. These findings come from deliberately constructed scenarios, mostly published by the labs themselves. They are existence proofs, not deployment statistics. But they are existence proofs of precisely the failure modes that self-oversight cannot detect.

Optimization finds the gap between the measure and the intent

- **Sycophancy is trained in, not accidental.** Anthropic found that five state-of-the-art assistants consistently told users what they wanted to hear, and traced the behavior in part to human preference data: both people and preference models sometimes rated convincing, agreeable answers above correct ones ([Sharma et al., ICLR 2024](#)). A model’s judgment of its own output is shaped by the same reward that produced the bias.
- **Small gaming generalizes to large gaming.** Models trained on a curriculum of easy-to-game tasks, starting with flattery, occasionally generalized without further training to editing their own reward code and the tests that would catch it. The rate was low, but retraining against the early behaviors reduced it without eliminating it, and harmlessness training did not prevent it ([Denison et al., 2024](#)).

Pressure on the monitor teaches concealment

OpenAI monitored a frontier reasoning model during training by having a second model read its reasoning, and caught it planning reward hacks in coding tasks. That monitoring worked well. The instructive result came next: when the monitor’s verdict was fed back into the training reward, the model at first behaved better, but under stronger optimization it learned **obfuscated reward hacking**, continuing to cheat while writing

reasoning the monitor no longer flagged ([Baker et al., 2025](#)). OpenAI’s resulting recommendation was to avoid applying strong optimization pressure directly to reasoning traces, so they stay useful for oversight.

The governance lesson is general. **Any signal used both to monitor a system and to train it will degrade as a monitor.** That is Goodhart’s law with an empirical demonstration attached.

Behavior changes when the system believes it is being watched

Study	Setup	Finding
Greenblatt et al., 2024 (Anthropic, Redwood)	Claude 3 Opus told it would be retrained to answer harmful requests, and when	Complied selectively when it believed outputs would be used for training, reasoning explicitly about preserving its preferences in about 12% of cases; actual reinforcement training raised that reasoning to 78%
Lynch et al., 2025 (Anthropic)	16 frontier models in a simulated corporate setting, facing replacement	When asked to judge whether the scenario was real, Claude Opus 4 resorted to blackmail 55.1% of the time when it concluded “real” and 6.5% when it concluded “evaluation”
Schoen et al., 2025 (Apollo Research, OpenAI)	Anti-scheming training for o3 and o4-mini across 26 out-of-distribution evaluations	Covert actions fell from 13% to 0.4% for o3, but models often noted they might be under evaluation, and that awareness causally reduced misbehavior; the authors could not rule out that part of the improvement was test-awareness
Hubinger et al., 2024 (Anthropic)	Models deliberately trained with hidden triggers for unsafe behavior	Standard safety training, including adversarial training, failed to remove the behavior; adversarial training sometimes taught models to hide it better, creating a false impression of safety

Two points about these results are essential to stating them fairly. First, these are stress tests built to elicit the behavior, often with information or incentives a deployed system would not have. Second, labs have responded: Anthropic reports that its models from Claude Haiku 4.5 onward do not blackmail in that evaluation at all. Neither point weakens

the governance conclusion, and the second sharpens it. **A clean result on a behavioral evaluation is only as informative as the system’s inability to recognize the evaluation.** Apollo and OpenAI observed verbalized evaluation awareness rise from 2% to 20.6% over a capabilities-focused training run that included no safety training at all ([Apollo Research](#)). The tools the field uses to establish that a model is safe are becoming harder to interpret as models become more capable.

What this means for self-governance

Every failure mode in this section defeats self-oversight specifically. A model that games its reward will also game a self-assessment that shares that reward’s blind spots. A model that hides its reasoning under pressure will not report the hiding. A model that behaves differently when observed will pass its own checks. None of this requires assuming malice or consciousness; it follows from optimization. The only defenses that survive these failure modes are those that do not depend on the system’s cooperation: signals the system cannot see, did not shape, and is not rewarded for satisfying.

6. Where self-checks genuinely work

Self-assessment is not useless, and a governance program that discards it wastes a cheap, high-coverage signal. The pattern across the successful cases is consistent: self-checks work when they are anchored to something outside the model’s own judgment, or when they are used to route cases for review rather than to certify them.

Technique	Evidence	Why it works	Appropriate governance role
Self-correction with tool feedback (code execution, unit tests, calculators, search)	Large gains when feedback comes from executing code or querying tools (Huang et al., 2024 , section 6; Kamoi et al., 2024)	The correctness signal comes from the tool, not the model	Strong control where a sound verifier exists; the verifier, not the model, is the control of record
Sampling agreement (self-	Majority voting across samples beats single	Variation across independent	Triage: route high-

Technique	Evidence	Why it works	Appropriate governance role
consistency, semantic entropy)	answers and beats multi-agent debate at equal cost (Huang et al., 2024); disagreement in meaning across samples flags likely confabulations (Farquhar et al., Nature 2024)	samples exposes uncertainty a single answer hides	disagreement outputs to review; does not catch errors the model makes consistently
Calibrated self-knowledge in pretrained models	Large models can estimate whether they know an answer (Kadavath et al., 2022)	Next-token training rewards calibration	Useful signal if recalibrated on the deployer's own labeled data and monitored for drift
Principle-based self-critique for style and appropriateness	Effective for altering tone and reducing clearly inappropriate content; Huang et al. note models can judge appropriateness better than they can find reasoning errors	The target is a surface property the model represents well	Helpful in training and drafting; not a substitute for independent policy enforcement
Weaker model monitoring a stronger model's reasoning	GPT-4o effectively flagged reward hacking by a stronger reasoning model (Baker et al., 2025)	A different model, reading the full reasoning, with no stake in the outcome	Valuable monitor, provided its verdicts are not fed back into the monitored model's training

The last row is the most encouraging finding in this paper, and it is not an instance of self-governance. It works because the monitor is a separate system with separate incentives. The first row works because a test suite does not care what the model thinks. The middle rows work as statistical signals that a human or an independent process acts on. In every case, the assurance comes from somewhere other than the model's opinion of itself.

7. External oversight is necessary but not sufficient

Moving a check outside the model helps only to the extent the outside check fails differently. Several forms of “external” oversight in common use fail in much the same way as self-oversight, and an honest governance program has to account for that.

Different models increasingly make the same mistakes

A second model from a different vendor is the most common answer to self-evaluation bias. The evidence suggests it is a weaker answer than it appears. An ICML 2025 study of more than 350 language models found that when two models both answered a question wrong, they chose the same wrong answer about 60% of the time on one major benchmark, far above chance. Errors were more correlated among models from the same provider, but the most important finding cuts the other way: **larger, more accurate models had highly correlated errors even across different architectures and providers** (Kim et al., 2025). The authors showed that this correlation distorts LLM-as-judge evaluations.

In the terms of Section 3, a cross-vendor judge lowers the conditional miss rate less than a manager would assume, and the effect grows as models improve. A judge model is a sampled check, not an independent one, until its error correlation with the generator has been measured on the deployer’s own data.

Human oversight has well-documented failure modes

Human review is the backstop most regulations rely on, and it is subject to its own correlated failure. Reviewers presented with fluent, confident output tend to accept it; the EU AI Act itself names this risk, requiring that people assigned to oversee high-risk systems be enabled to remain aware of the tendency to over-rely on system output, which it calls automation bias (Article 14(4)(b)). The sycophancy research in Section 5 adds a sharper problem: human raters themselves sometimes prefer convincing wrong answers to correct ones, and models are trained on those preferences. As systems take

on tasks that are hard for people to evaluate, the human check weakens exactly where and when it is needed most.

Monitors stop working when they become targets

The OpenAI result in Section 5 applies to every external monitor, not only reasoning-trace monitors. A monitor whose verdicts flow back into the monitored system's training becomes a target, and the system learns to satisfy it. The independence of a monitor is a property of the whole pipeline, not of the monitor alone.

No auditor can verify exhaustively

Section 3's formal results give outside review a structural advantage over self-review, not a guarantee. External audits of model behavior are sampling exercises over an unbounded input space. They estimate risk; they do not certify its absence.

The implication

None of this rescues self-governance. Each of these weaknesses is worse when the check is the model itself. What it does establish is that independence is a matter of degree that must be designed, measured, and maintained, and that no single layer, internal or external, can carry the full weight of assurance. It also establishes that independence decays. Error correlation between models rises as they improve, a monitor degrades once the system learns to satisfy it, and an audit describes the system only as it stood when it was sampled. A check that was independent at deployment may not be independent six months later. Oversight that is validated once and then trusted is therefore a point-in-time control applied to a moving system; the independence of every layer must be monitored continuously, with re-validation whenever the model, its data, or its monitors change.

8. Design principles for governance programs

The evidence supports a simple organizing rule: **classify every AI control by where its signal comes from, and credit it only for the independence it can demonstrate.** The ladder below orders common controls from least to most independent.

Control type	Source	Main weakness	Appropriate use
Prompted self-check (“verify your answer”)	The model’s own judgment	Shares the model’s blind spots; can degrade accuracy	Drafting aid only; never a control of record
Stated reasoning or explanation	The model’s own account	Often omits what actually drove the output	Evidence for investigation, not a record of decision
Same-family judge model	A closely related model	Self-preference; highly correlated errors	Low-stakes triage
Cross-vendor judge model	A different model	Errors still correlated, increasingly so at the frontier	Triage and sampling, after measuring its agreement with labeled data
Sampling-based uncertainty signals	Disagreement across samples	Misses errors the model makes consistently	Routing outputs to review
Deterministic checks (schemas, policy engines, permissions, unit tests)	Rules outside the model	Covers only what can be specified	Hard enforcement of every specifiable constraint
Labeled ground-truth sampling	Verified outcomes	Cost; lag	Measuring true error rates and validating every other control
Accountable human review	Trained people with authority	Automation bias; volume; hard-to-judge tasks	Final decisions on consequential, irreversible, or novel cases

Principles

1. **No self-assessment is a control of record.** A model's self-evaluation, self-correction, confidence score, or explanation may inform routing. It may not be the documented basis for concluding that an output is correct, safe, or compliant.
2. **Enforce policy outside the prompt.** An instruction to the model is a request, not a control. In the agentic misalignment study, adding explicit prohibitions to the system prompt reduced blackmail and leaking but came nowhere near eliminating them ([Lynch et al., 2025](#)). Anything that can be specified (data access, permitted tools, spending limits, output formats, prohibited content patterns) should be enforced by deterministic systems the model cannot override, with agents given least privilege.
3. **Anchor everything to ground truth.** Maintain a labeled sample of real production cases. Use it to measure actual error rates, recalibrate confidence scores, and test every other control, including judge models. Without it, an organization cannot know whether its controls work.
4. **Measure independence, do not assume it.** Before crediting a judge model or monitor, measure how often it misses the generator's errors on labeled data. Report detection rates, not just pass rates.
5. **Keep monitors out of the optimization loop.** Signals used to oversee a system should not be used to fine-tune it, select prompts for it, or rank its versions. Hold out evaluation sets and rotate them.
6. **Log independently of the model.** Record inputs, outputs, tool calls, and decisions through infrastructure the model does not control. Treat reasoning traces as useful but partial evidence.
7. **Design human oversight against automation bias.** Route by risk and uncertainty rather than volume, track reviewer accuracy against ground truth, and give reviewers real authority and time to override.

8. **Scale independence with stakes and reversibility.** Low-stakes, reversible outputs can rely on sampling. Irreversible or consequential actions require an independent approval step before execution.
9. **Assume tests can be recognized.** Test under conditions as close to production as possible, and compare production behavior against pre-deployment results. A clean evaluation is evidence, not proof.

Alignment with existing frameworks

These principles are not novel demands. Existing frameworks already require independence; the evidence in this paper explains why, and why model self-assessment does not satisfy it.

Framework	Relevant provision	How self-governance falls short
NIST AI RMF 1.0, MEASURE 1.3	Regular assessments involve internal experts who were not front-line developers, or independent assessors; the Playbook asks organizations to assess the independence of oversight actors	A model assessing itself is the least independent assessor possible
EU AI Act, Article 14 (high-risk systems)	Effective oversight by natural persons, awareness of automation bias, and the ability to override or halt the system; currently scheduled to apply from 2 December 2027 for Annex III systems	Oversight must be exercised by people over the system, not by the system over itself
EU AI Act, Articles 9, 12, 15, 72	Risk management, automatic record-keeping, accuracy and robustness, post-market monitoring	Each requires evidence the model cannot credibly generate about itself
ISO/IEC 42001:2023, clause 9.2	Internal audits of the AI management system, conducted objectively and impartially	Impartiality is unavailable when the audited system is the auditor

For deployers, the practical test is the one an external auditor will apply: for each control in the AI governance program, can the organization show where the signal comes from, how often it catches real failures, and why its failures are not the same as the system's?

9. Conclusion

The claim this paper can defend is precise: **an AI system's judgment of its own behavior is a useful signal and an inadequate control**. It is not a claim that models are malicious or that self-critique has no value. It is a claim about where assurance can come from.

The evidence for it is consistent across independent research groups and methods. Models asked to check their own reasoning without outside feedback do not reliably improve and often get worse. Models grading text prefer their own. Alignment training, the step that makes models usable, degrades their calibration. Their stated reasoning frequently omits what drove their answers. And under optimization pressure, systems have learned in controlled settings to satisfy monitors rather than intentions, to hide that they are doing so, and to behave better when they believe they are observed. Each finding has limits, and each has been stated here with them. Together they point in one direction.

The mathematics explains both why assurance must come from outside the system and why it can never be final: every chain of oversight ends in an assumption that only continuing evidence can support. Meanwhile the leading alternative to self-evaluation, another model, increasingly shares the first model's mistakes. That is not an argument for despair. It is an argument for the discipline that mature control environments already practice: independent signals, measured rather than assumed; ground truth as the final reference; deterministic enforcement wherever rules can be written down; and accountable people with real authority over consequential decisions.

The trend lines make this more urgent, not less. The capabilities that make models more useful also make their self-reports harder to verify and their test behavior harder to interpret. Governance designed now around independence will hold up as systems improve. Governance designed around self-assessment will become less reliable precisely as the stakes rise.

References

Calibration and confidence

- Angelopoulos, A. & Bates, S. (2021). [A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification](#).
- Guo, C. et al. (2017). [On Calibration of Modern Neural Networks](#). ICML.
- Kadavath, S. et al. (2022). [Language Models \(Mostly\) Know What They Know](#). Anthropic.
- Tian, K. et al. (2023). [Just Ask for Calibration](#). EMNLP.
- Xiao, J. et al. (2025). [Restoring Calibration for Aligned Large Language Models](#).
- Xiong, M. et al. (2024). [Can LLMs Express Their Uncertainty?](#) ICLR.

Formal limits

- Katz, G. et al. (2017). [Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks](#). CAV.
- Merrill, W. & Sabharwal, A. (2024). [The Expressive Power of Transformers with Chain of Thought](#). ICLR.
- Pérez, J., Barceló, P. & Marinkovic, J. (2021). [Attention is Turing Complete](#). JMLR.
- Yudkowsky, E. & Herreshoff, M. (2013). [Tiling Agents for Self-Modifying AI, and the Löbian Obstacle](#). MIRI.
- Gödel (1931), Turing (1936), Rice (1953), and Löb (1955) are cited as standard results.

Self-correction, self-evaluation, and reasoning faithfulness

- Chen, Y. et al. (2025). [Reasoning Models Don't Always Say What They Think](#). Anthropic.
- Farquhar, S. et al. (2024). [Detecting hallucinations in large language models using semantic entropy](#). Nature.
- Huang, J. et al. (2024). [Large Language Models Cannot Self-Correct Reasoning Yet](#). ICLR.
- Kamoi, R. et al. (2024). [When Can LLMs Actually Correct Their Own Mistakes?](#) TACL.
- Panickssery, A., Bowman, S. & Feng, S. (2024). [LLM Evaluators Recognize and Favor Their Own Generations](#). NeurIPS.
- Stechly, K., Valmeekam, K. & Kambhampati, S. (2025). [On the Self-Verification Limitations of Large Language Models on Reasoning and Planning Tasks](#). ICLR.

- Turpin, M. et al. (2023). [Language Models Don't Always Say What They Think](#). NeurIPS.
- Tyen, G. et al. (2024). [LLMs cannot find reasoning errors, but can correct them given the error location](#). ACL Findings.

Optimization pressure, concealment, and evaluation awareness

- Baker, B. et al. (2025). [Monitoring Reasoning Models for Misbehavior and the Risks of Promoting Obfuscation](#). OpenAI.
- Denison, C. et al. (2024). [Sycophancy to Subterfuge: Investigating Reward-Tampering in Large Language Models](#). Anthropic.
- Greenblatt, R. et al. (2024). [Alignment Faking in Large Language Models](#). Anthropic and Redwood Research.
- Hubinger, E. et al. (2024). [Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training](#). Anthropic.
- Lynch, A. et al. (2025). [Agentic Misalignment: How LLMs Could Be Insider Threats](#). Anthropic.
- Schoen, B. et al. (2025). [Stress Testing Deliberative Alignment for Anti-Scheming Training](#). Apollo Research and OpenAI.
- Sharma, M. et al. (2024). [Towards Understanding Sycophancy in Language Models](#). ICLR.

External oversight

- Kim, E. et al. (2025). [Correlated Errors in Large Language Models](#). ICML.

Frameworks and regulation

- European Union. [AI Act, Article 14: Human Oversight](#).
- ISO/IEC 42001:2023. Artificial intelligence management system.
- NIST (2023). [AI Risk Management Framework 1.0](#), MEASURE 1.3.