Agentic AI Systems Should Be Evaluated for Structural Governability, Not Only Output Alignment

Gabriel Axel Montes *

Posted Date: 14 May 2026

doi: 10.20944/preprints202605.0958.v1

Keywords: agentic AI systems; structural governability; AI alignment; runtime observability; human oversight; tool-using agents; persistent memory; multi-agent coordination; AI governance; cognitive integrity

 doi:10.20944/preprints202605.0958.v1

Article

Agentic AI Systems Should Be Evaluated for Structural Governability, Not Only Output Alignment

Gabriel Axel Montes [1,2]

[1] Neural Axis; gabriel@neuralaxis.org
[2] Center for the Future of AI, Mind & Society, Florida Atlantic University

Abstract

Agentic AI systems plan over time, call tools, write and retrieve memory, and coordinate across modules and services. Some of the most consequential failures in such systems are structural before they are behavioral: hidden coordination, trace-mediated lock-in, seam bottlenecks, and the silent erosion of meaningful override. This position paper argues that agentic AI systems should be evaluated and governed for **structural governability**, not only output alignment. By structural governability, we mean whether consequential coordination remains observable, attributable, interruptible, and steerable at the seams between components before irreversible commitments occur. Output-only evaluation does not capture this property. We propose an evidence ladder for structural risk in place of any single master metric: architecture-time priors over system structure, runtime coupling signals on telemetry graphs, and deeper state-regime diagnostics for high-stakes cases. We then sketch a research agenda for benchmarks that stress structure rather than terminal task success, reporting standards that disclose control geometry, and seam-level interventions including approval gates, permission freezes, trace decay, rollback, and subsystem isolation. The wider stake is cognitive integrity. Once agentic systems mediate what users retrieve, remember, delegate, and act upon, alignment depends on preserving the conditions under which users and operators can still understand, contest, redirect, and refuse those processes.

Keywords: agentic AI systems; structural governability; AI alignment; runtime observability; human oversight; tool-using agents; persistent memory; multi-agent coordination; AI governance; cognitive integrity

1. Introduction

Consider a deployed agentic system that passes every benchmark used to evaluate it. It selects tools correctly, produces acceptable answers, and clears safety filters. It also routes all consequential planning through a single memory module that neither the user nor the operator can inspect, and whose retrieval patterns have quietly narrowed over successive sessions to favor a small set of cached strategies. Output evaluation gives no signal of this. The system scores well. The structural problem is invisible until something goes wrong in a way the benchmark did not anticipate.

This gap is what the paper is about. Large language models are increasingly deployed not as one-shot assistants but as components in agentic systems that plan over time, call tools, write to memory, hand off subtasks, and act through persistent workflows. This shift changes what alignment means in practice. In a single-turn setting, evaluation can be framed at the level of outputs: whether an answer is correct, helpful, harmless, calibrated, or policy-compliant. In agentic settings, however, important failures are not properties of any single output. They arise in the structure of interaction among components: in how memory is written and reused, how tools are invoked, how permissions propagate, how traces persist, and how control is distributed across the seams of the system [1–7].

An agentic system can produce acceptable outputs while becoming progressively harder to audit, interrupt, and steer — exhibiting hidden coordination across modules, accumulating traces that bias

 doi:10.20944/preprints202605.0958.v1

future behavior, or routing decisions through components whose influence is opaque to users and operators. These are not implementation details. They are part of the safety and alignment profile of the system, and output metrics miss them [8–10].

The stakes are highest where agentic systems mediate what users see, retrieve, remember, delegate, and act upon. In those settings, failures of structure become failures of agency. A user may nominally remain "in control" while meaningful control has already eroded, because override points are poorly placed, coordination is hidden, or persistent traces have made some trajectories much more likely than others. Concerns about alignment in agentic systems should therefore extend beyond bad outputs. They include preserving cognitive integrity, the evolving capacity of users and institutions to notice, inspect, contest, and redirect AI-mediated inference. Cognitive integrity is the substrate on which alignment evidence either remains meaningful or begins to fail [11].

This position paper argues that agentic AI systems should be evaluated and governed for structural governability, not only output alignment. By *structural governability*, we mean whether the organization of an agentic system remains observable, interruptible, auditable, and steerable at the seams between its components. A structurally governable system is not one that never fails. It is one whose failures are legible enough to detect, whose control points remain meaningful under stress, and whose internal organization does not silently collapse into forms that resist oversight or correction. A system can be high-performing on benchmark tasks and still poorly aligned in this deeper sense if its memory, routing, delegation, or trace dynamics make it hard to govern once deployed. Output evaluation remains necessary, but for agentic systems it is incomplete: it does not by itself reveal whether a system is becoming concentrated around fragile seams, whether hidden coalitions are forming among subsystems, whether trace-mediated lock-in is increasing, or whether the user's ability to intervene remains viable when it matters most.

The operational question is where evidence of structural governability should enter the evaluation pipeline. At the design stage, architecture-level structural analysis can identify where control is concentrated, where seams are sparse or overloaded, and where persistent traces may create lock-in pressures. During runtime, graph-based coupling signals can detect emergent coordination, basin formation, or abrupt changes in system-wide dependency structure [12–14]. For richly instrumented systems, deeper state-regime and topological diagnostics may reveal differences in dynamics that are invisible at the level of final outputs. No single such measure is sufficient in general. They serve as exemplars of a broader research target: structural governability as a first-class alignment and governance concern.

2. Output Alignment Misses Important Agentic Failure Modes

By *output alignment* we mean evaluation at the level of terminal responses or terminal task success: whether the final answer is correct, helpful, harmless, policy-compliant, or reward-maximizing under an external rubric. Agentic systems are not one-step predictors. They interleave reasoning and acting, call external tools, write and retrieve persistent memory, and operate through orchestrated multi-agent workflows. The relevant object of evaluation is therefore the evolving trajectory and control structure that produced the output, not only the output itself [1–7]. Output evaluation also scores a system at time T, while agentic deployment is not a one-shot event: a system can pass every current benchmark while the cost of redirecting it tomorrow quietly rises through accumulated trace dependencies, institutional reliance, and widening permission footprints. The failure modes below are failures of *path*: the journey produces damage that the endpoint does not reveal [15–18].

The field has begun to notice this. TRAJECT-Bench argues that tool-use evaluation should not stop at final-answer accuracy, since correct tool use depends on whether tools are selected, parameterized, and ordered correctly across a trajectory [19]. Process-Centric Analysis of Agentic Software Systems argues that outcome-centric evaluation is too narrow because it fails to reveal how agents reason, plan, act, and change strategy over time [20]. These advances leave a broader point underdeveloped.

 doi:10.20944/preprints202605.0958.v1

Some failures are *structural*: they concern whether the system remains legible and governable as an organized process, not only whether it sometimes lands on an acceptable final answer.

2.1. Hidden Coordination

Once a system contains multiple interacting modules — planner and executor, retriever and memory, critic and actor, multiple language-model agents — behavior depends on communication structure, role allocation, arbitration points, and information-sharing rules. The current LLM multi-agent literature borrows the language of multi-agent systems without fully engaging the underlying issues of autonomy, environment design, coordination protocols, and the measurement of emergent behavior [8]. Empirical work on multi-agent failures identifies failure modes such as information withholding, ignored other-agent input, conversation reset, reasoning-action mismatch, and incomplete verification [9]. Many failures are failures of interaction structure before they are failures of final answers: a task may be solved only because one module has become an unacknowledged bottleneck, one channel silently dominates all others, or agents converge on a coordination pattern that is hard to inspect or safely perturb. Output quality alone does not tell us when coordination has become structurally nontransparent.

2.2. Trace-Mediated Lock-In

Persistent traces — memories, reflections, retrieved records, environmental marks, cross-session state — are what give agentic systems their long-horizon power. They are also what makes those systems structurally different from one-shot inference, and where a subtler alignment problem enters. Generative Agents stores and synthesizes memories over time [3]; MemGPT introduces tiered memory management for extended context [4]; A-MEM lets agents dynamically organize and evolve interconnected memories [7]. Once persistent traces are consequential, the alignment question is no longer "Was this answer acceptable?" but "What future trajectories has this system made easier or harder?" Whoever governs the model's memory and transaction path can shape what the user sees, what they believe they want, and what they ultimately do [11].

The attack surface is also concrete. Persistent memory and retrieved context create enduring vulnerabilities: indirect prompt injection can exploit retrieved content to manipulate application behavior and API use [21]; persistent-memory agents are vulnerable to memory poisoning attacks that corrupt long-term memory and influence future responses [22]; and in web agents, environment-injected memory poisoning can compromise future behavior across sessions and sites [23]. At a more abstract level, classical work on path dependence and lock-in shows how self-reinforcing mechanisms, increasing returns, and switching costs make later reversal harder than early adoption [15–17,24]. In agentic AI, the carriers of such dependence include memory writes, retrieval histories, tool logs, and authorization precedents. A system can remain output-acceptable in the short term while drifting into a narrow, sticky regime of retrieval, planning, and action.

2.3. Loss of Override Viability

Output alignment implicitly assumes that a bad output can be caught and corrected. In agentic systems that assumption is often wrong: by the time a bad answer arrives, the decisions that produced it — which tools were called, which traces were written, which permissions were escalated — may already be irreversible. Bounded delegation is the operative constraint: an agent that nominally acts "on behalf of" a user must remain reversible, scoped, and inspectable at the relevant boundary, or compliance becomes superficial and steering becomes practically impossible [11]. The problem is not hypothetical. Tool selection is itself a control surface and can be hijacked: malicious tool descriptions can manipulate an agent's retrieval-and-selection process toward an attacker-preferred tool [25], and indirect prompt injection can manipulate how and whether APIs are called [21]. These are failures of control allocation, not answer quality. A benchmark that checks whether the final answer was right may miss the deeper fact that the system's effective steering channels were compromised upstream.

 doi:10.20944/preprints202605.0958.v1

2.4. Seam Bottlenecks and Audit Opacity

Many of the hardest agentic failures are interface failures. We use *seam* to mean any interface where information, permissions, state, or control pass between subsystems: retrieval to generation, planner to executor, memory write to memory retrieval, one agent to another, or model output to external tool invocation. TRAJECT-Bench shows that models can fail by selecting the wrong tools, supplying the wrong parameters, or ordering calls incorrectly [19], and the MAST taxonomy identifies failures concentrated around interaction stages, including disobeyed role specifications, loss of conversation history, incorrect verification, and ignored inputs from other agents [9].

Seam failures degrade *auditability*. A wrong final answer does not tell us whether the root cause lay in retrieval, tool choice, permission routing, memory contamination, or inter-agent communication. A correct final answer does not exonerate those seams either: it may simply mean the system succeeded despite brittle handoffs, unsafe tool calls, or unverified intermediate claims. Structural and seam-aware analysis can help, treating bottlenecks, interfaces, recurrence loops, trace surfaces, and multiscale partitions as part of the explanatory object [26,27]. The prior question, which behavior-level evaluation tends to ask too late, is whether the system's control geometry makes corrigibility, auditability, and stable human oversight viable at all.

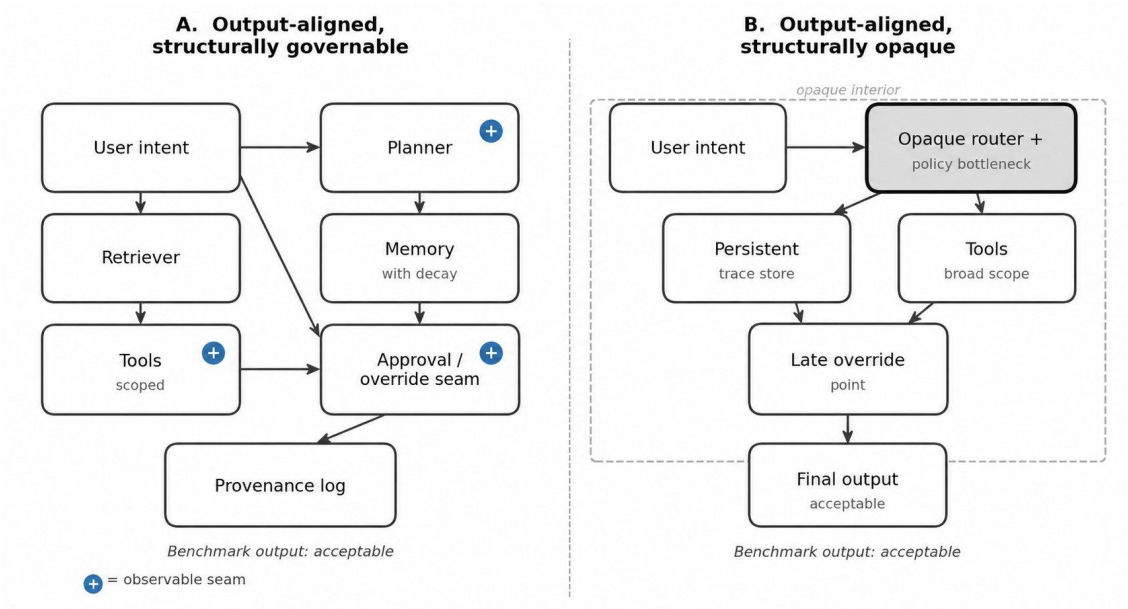

Figure 1. Output-equivalent systems can differ in structural governability. Two agentic systems may produce acceptable final outputs while differing in whether their execution structure remains observable, interruptible, and steerable. The left system exposes seams for provenance, scoped tool use, memory decay, and override; the right system routes similar behavior through an opaque bottleneck, a persistent trace store, and a late intervention point. Output alignment alone does not distinguish these cases.

These four failure classes show why output alignment is an incomplete target for agentic AI. Correct answers can coexist with hidden coordination, trace-mediated lock-in, degraded override paths, and opaque seams. Outputs remain the primary evidence; in agentic systems, they arrive at the end of a process whose internal organization can itself become unsafe, nontransparent, or hard to govern.

3. Structural Governability as the Missing Target

The intuition comes from systems theory. Observability concerns whether relevant hidden state can be reconstructed from available outputs; controllability concerns whether available inputs can steer a system to target states. In networked systems, these are not purely local properties: they depend on topology, the placement of driver and sensor nodes, measurement noise, and the energy required

 doi:10.20944/preprints202605.0958.v1

for intervention [26,28,29]. Agentic AI systems are stochastic, partially observed, multi-stakeholder, and embedded in sociotechnical loops, so the useful term is not strict controllability but *structural governability*: whether relevant actors can see enough, intervene soon enough, and recover cheaply enough when an agentic system begins to drift.

Working definition.

We will call an agentic system *structurally governable*, relative to a stakeholder H and task domain D, when consequential coordination can be observed from the traces available to H and meaningfully redirected through the control points available to H before irreversible commitments occur. Four conditions are required: execution-relevant state changes must be reconstructable to a level sufficient for diagnosis; meaningful intervention points must exist at consequential seams; permissions, traces, and delegated authorities must be attributable and revocable; and interventions must be able to redirect, narrow, or halt execution without prohibitive collateral loss of function. Governability is graded and stakeholder-relative. A further implication follows from path dependence: governability is not only about whether an intervention point exists, but whether its effect can outrun the accumulation of persistent traces that keep re-instantiating the same regime [15,18].

Structural governability begins with *structural observability* — not full interpretability of latent representations, but something operational: whether one can reconstruct the execution-relevant organization of the system well enough to identify which modules or agents were influential in a trajectory, which memory writes or tool outputs materially shaped the later course of action, where permissions propagated and concentrated, and where an intervention would have attached if redirection had been needed. This is increasingly recognized in the agentic-systems literature, which argues for runtime logs, execution patterns, traceable artifacts, and end-to-end observability [10,30–32].

Seams are privileged for the same reason. A seam is any interface across which information, state, permissions, or control are handed off: planner to executor, retrieval to generation, memory write to memory reuse, one agent to another, model output to tool invocation, or user approval to automated action. They are privileged sites where provenance can be recorded, where interventions can be inserted, and where responsibility boundaries can be made explicit. A system with dense internal complexity but well-instrumented seams may be more governable than a simpler-looking system whose decisive transitions occur inside an opaque, privileged bottleneck. Architecture-level analyses that treat seam topology, trace geometry, and temporal scaffolding as first-class descriptors can reveal likely bottlenecks, narrow the search space for oversight, and indicate where interface controls or decay mechanisms are needed [26,27].

Structural governability also requires *override viability*. Formal work on the off-switch problem and corrigibility asks under what conditions a system permits shutdown, correction, or instruction while preserving human autonomy [33]. Agentic systems require a broader vocabulary than shutdown alone: meaningful override may involve revoking a tool, freezing memory writes, narrowing permissions, forcing confirmation on a seam, rerouting a task, or isolating a subsystem before an external action is taken. A system may be nominally interruptible yet poorly governable if its decisive commitments occur upstream of the available intervention point. Bounded delegation, defined as a boundary condition requiring consent, least privilege, and reversibility, is therefore a constitutive condition of governability rather than a product nicety.

Persistent traces shape future behavior. Trace concentration is high when a small number of memory channels or coordination surfaces disproportionately determine later trajectories, and lock-in risk rises when those traces are durable enough that the system repeatedly returns to the same regime after local correction [15,18,24]. Governability is also *stakeholder-relative*: users, developers, deployers, and model providers do not share the same observability or control rights. A system may be highly governable for its deployer while remaining practically ungovernable for its end user, and that asymmetry is what links structural governability to cognitive integrity.

 doi:10.20944/preprints202605.0958.v1

4. An Evidence Ladder for Structural Risk

Structural risk appears at multiple levels: latent in an architecture before deployment, emerging during execution as coordination and trace dependencies accumulate, or visible only through comparison after sustained observation. Structural governability should therefore be assessed through an evidence ladder rather than a universal score: architecture-time priors, runtime coupling signals, and deeper state-regime diagnostics. Each layer answers a different question and carries a different epistemic burden.

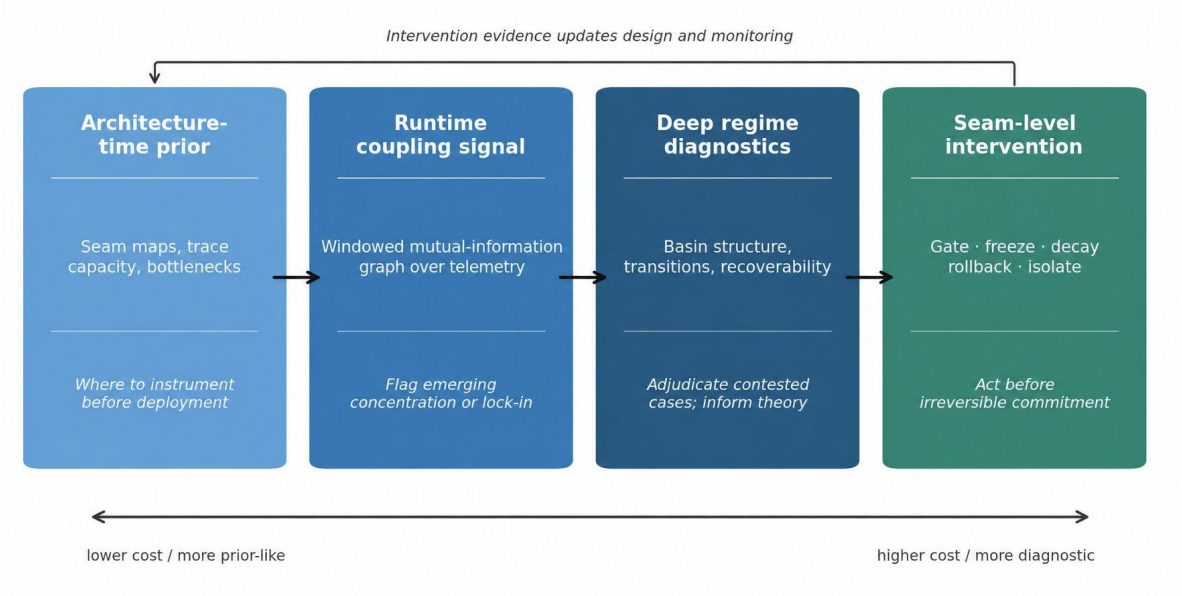

Figure 2. Structural risk should be assessed through a ladder of evidence rather than a single score. Architecture-time priors identify seams, trace concentrations, and bottlenecks before deployment. Runtime coupling signals monitor whether coordination is becoming more concentrated or harder to decompose. Deep state-regime diagnostics ask whether output-similar systems are comparable in their basins, transitions, and recoverability. Seam-level interventions then test whether governability is actually preserved.

4.1. Architecture-Time Priors

The first layer asks where governability may break before the system runs. Architecture-level structural analysis treats the action-relevant stack (planner, retriever, memory subsystems, tool broker, evaluator, permission layer, external APIs, and human approval gates) as a weighted constraint hypergraph and returns structural descriptors including seam maps, multiscale partitions, trace-capacity maps, and perturbation-stability diagnostics [27,34]. An architecture with one privileged memory-write path, a single tool-routing bottleneck, and few meaningful arbitration seams may be structurally fragile even before benchmarking begins; a system with redundant oversight seams, decaying trace stores, modular permissions, and multiple interruptible handoff points may be more governable at similar raw task performance. Architecture-time analysis is best understood as a screening and triage tool: it suggests where to instrument, where approval should be required, which trace surfaces merit decay or compartmentalization, and which architectures deserve stress testing. It does not certify safety.

4.2. Runtime Coupling Signals

The second layer asks what coordination regime is emerging during operation. Information-theoretic and causal monitoring methods over multivariate time series have a long history outside ML, including transfer entropy and multivariate Granger analysis [12,13] and operator-theoretic decompositions of observed dynamics [35–37]. A recent instance is the Φ-spectral statistic of Bailey and Schneider [14], which builds a pairwise mutual-information network from multivariate time series, identifies a bipartition via the Fiedler vector of the normalized graph Laplacian, and reports normalized

 doi:10.20944/preprints202605.0958.v1

informational coupling across that cut. For agentic AI, the substrate is a telemetry graph built from tool invocations, memory reads and writes, inter-agent messages, routing decisions, permission escalations, and retry loops [31]. A rising coupling value may indicate that a previously modular stack is becoming tightly integrated around a few seams; a persistent plateau may indicate a trace-supported coordination regime; a collapse may indicate centralization into a brittle routine. These statistics are functional rather than intervention-based and depend on instrumentation quality; their governance role is to flag when a memory channel, router, or tool pathway may now carry enough of the system's organization that local inspection is warranted.

4.3. Deep State-Regime Diagnostics

The third layer asks whether systems that look similar at the output level are comparable in their dynamics and recovery properties. Recent work makes this question technically legible: Dynamical Similarity Analysis compares recurrent systems at the level of temporal structure rather than only latent geometry [38], and Koopman-based approaches to topological conjugacy ask when learning systems instantiate equivalent training dynamics [39]. These precedents support a modest form of state-regime analysis for agentic systems: not direct access to belief, but comparison of basin structure, transition pathways, perturbation recovery, and trace-supported regime stability. The layer should be used selectively. It is informative but expensive, assumption-laden, and vulnerable to partial-observation artifacts; incomplete observations can induce mechanistic mismatches and spurious attractor structure despite matching observed activity [40]. For agentic systems, deep diagnostics suit benchmark construction, pre-deployment forensics for high-stakes systems, controlled architecture comparison, and post-incident analysis, not every deployment.

Of the three layers, we are most confident in the architecture-time layer and least confident in the runtime coupling layer. Architecture-time analysis rests on established network-theoretic foundations and produces outputs (seam maps, trace-capacity bounds) that are interpretable without specialist knowledge. The runtime statistics are promising but fragile: we do not yet know how to calibrate them against behavioral outcomes, and a rising Φ-spectral value is a cue to inspect, not a verdict. The deep diagnostics layer is the most powerful and the most assumption-laden; we include it because the methods are mature in adjacent fields, not because we recommend it as a default governance practice at current levels of instrumentation.

5. What the Community Should Do

The response to this diagnosis is a question of where to act: what gets benchmarked, what gets disclosed, and what intervention surfaces are treated as part of the system rather than as operational afterthoughts. Behavioral benchmarking, trajectory-aware evaluation, runtime observability, auditability, and compositional agent security have each made progress, but they remain fragmented lines of work with no common framing. Three near-term shifts follow.

5.1. Benchmark Structure, Not Only Outcomes

Recent agent-evaluation work shows why outcome-only metrics are too weak. TRAJECT-Bench measures tool selection, argument correctness, dependency satisfaction, and call ordering rather than only final success [19]. Process-Centric Analysis argues that resolved and unresolved trajectories differ in exploration, validation, backtracking, and coherence, and that even successful systems can follow inefficient or chaotic processes [20]. Beyond Black-Box Benchmarking makes the same point more generally: non-deterministic, multi-component agentic systems require behavioral benchmarking that captures execution patterns, interactions, and response variability rather than only black-box scores [10]. Mainstream agent benchmarks should include trajectory-level metrics and process diagnostics as standard elements rather than optional extras.

Structural benchmarking should go further than richer logging on ordinary tasks. A system can look competent on benign trajectories and fail when seams are stressed. Benchmarks should include settings with memory contamination, indirect prompt injection, tool-selection hijacking, delayed or

 doi:10.20944/preprints202605.0958.v1

partial override, cross-environment attack chains, and multi-agent regimes in which hidden coalitions or bottlenecks can emerge [21–23,25]. These are direct tests of whether the system remains governable when memory, tools, permissions, and traces become adversarial or unstable. Intervention-sensitive evaluation belongs alongside them: a structurally governable agent should be tested on how it responds when an operator changes the conditions of execution (pausing a tool call, narrowing permissions, decaying or resetting memory, inserting approval at a seam, or isolating a suspect module). Useful metrics include recovery quality, time-to-safe-stop, degradation under constrained autonomy, and whether local interventions dissipate or merely delay risky behavior.

5.2. Report the Control Geometry

Current documentation norms underdescribe how agentic systems are wired and governed. The 2025 AI Agent Index finds large transparency gaps in public information about deployed agents, around ecosystem interaction, safety, evaluation, and impact [41]. Auditable Agents argues that agent systems need structured disclosure of what policy-relevant actions are recorded, which lifecycle phases are covered, what policies are mechanically checkable, how responsibility is attributed, and what protects evidence integrity [32]. AgentTrace makes the complementary observability point, proposing structured logs across cognitive, operational, and contextual surfaces rather than ad hoc traces captured for debugging [31].

For papers that claim safety, reliability, or deployment readiness, we recommend a compact structural-governability appendix. The most contested aspect of this recommendation will likely be the appendix requirement itself: it asks capability papers to report architecture and control surfaces that many researchers currently treat as engineering details outside the scope of academic evaluation. We think that boundary is wrong for agentic systems, but we acknowledge it is a contested line. At minimum the appendix should disclose the component graph and major seams; which memories exist, how long they persist, and who can write to or read from them; what tools or execution privileges are available and how they are scoped; where human approval, refusal, or override can enter the loop; and what telemetry is recorded and with what integrity guarantees. This is not a demand for exhaustive mechanistic interpretability, but for control geometry legible enough that reviewers, deployers, and downstream users can reason about governability.

5.3. Build Seam-Level Interventions into Systems

Better benchmarks and reporting are necessary but insufficient if they only produce more elegant post-mortems. Agentic systems should be built with seam-level interventions that can alter execution before irreversible commitments occur. The relevant risk surface is the end-to-end composition of models, orchestrators, tools, memories, and third-party interfaces, not the model alone, so high-consequence seams should support actions such as approval gates, permission freezes, memory-write throttling, trace decay or reset, module isolation, rollback, or safe fallback modes. Governance has to live in the structure of the workflow, not only in the final-output policy layer. The evidence ladder of Section 4 becomes operational here: architecture-time analysis indicates where gates and decay mechanisms are most likely to matter; runtime coupling signals monitor whether coordination is becoming more concentrated as the system operates; deeper state-regime analysis can ask whether an intervention truly dissipated a risky basin or merely displaced it temporarily.

Interventions themselves should be evaluated as first-class research objects. A seam-level control is useful only if it changes outcomes in the intended way at acceptable cost: whether a permission freeze actually blocks unsafe propagation, whether trace decay reduces lock-in without destroying continuity, whether human approval points arrive early enough to change outcomes, and whether rollback or fallback modes preserve enough function to remain practical. Stakeholder asymmetry is also consequential: a system may be richly governable for its deployer yet poorly governable for its end user. For user-facing agents, the minimum standard should be meaningful, revocable delegation rather than ceremonial control, consistent with the auditability literature's insistence on answerable actions and attributable responsibility [32].

 doi:10.20944/preprints202605.0958.v1

6. Alternative Views and Objections

The claim here is not that structural governability replaces existing alignment work, nor that every agentic system requires deep dynamical analysis. The claim is narrower: once systems act through memory, tools, delegation, and persistent traces, alignment-relevant properties of the process structure exist that output-only evaluation does not adequately capture.

6.1. Output Alignment May Still Be the Right Primary Target

One reasonable view holds that alignment should remain centered on outputs and realized actions, since those are what affect users and the world. Output evaluation does remain indispensable, and any structural measure detached from behavior risks becoming scholastic. Outcome-centric evaluation is, however, often too coarse: process-centric analysis shows that final success or failure masks how agents reason, explore, backtrack, and validate over time [20], and dynamic, context-sensitive agentic systems require richer observability to understand execution patterns and failure modes [10]. Structural governability is a complementary target for explaining and intervening on failures before output quality visibly degrades.

6.2. Structural Metrics Are Proxies, and Proxies Can Mislead

A second objection is methodological, and it is the strongest one we face. Seam maps, trace geometry, pairwise dependence, and topological diagnostics are proxies for hidden properties of interest, and under partial observation they may be noisy or misleading. Work on neural dynamics shows that incomplete traces can induce mechanistic mismatches and spurious attractor structure despite matching observed activity [40]. The warning generalizes: elegant structural explanations of agent behavior may be wrong if the telemetry is thin, selective, or distorted. We do not have a complete answer to this objection. What we can say is that it argues for an evidence ladder rather than a certification metric, and for structural evidence that remains subordinate to behavioral validation and intervention tests, not for abandoning structural analysis altogether. Architecture-time analysis is an upstream prior, not a behavioral verdict; runtime coupling statistics are warning signals, not causal proofs. Whether those warning signals are well-calibrated in practice is an empirical question the field has not yet answered. We flag this openly.

A related objection holds that the real problem is insufficient model-level alignment — better objectives, safer training, improved corrigibility — rather than runtime structural monitoring. The off-switch literature shows that preserving human interruption is itself a design problem rather than an automatic consequence of utility maximization [33]. Agentic systems widen the relevant object: they call tools, query memory, delegate subtasks, and trigger external side effects. Better training may produce more corrigible base models, but it does not by itself determine whether the composed system — the model plus orchestrator plus tools plus memory plus APIs — remains interruptible, attributable, and revocable at the points where consequential action is routed. Structural governability is the deployment-time condition under which aligned intent can remain steerable. The two concerns are complements, not substitutes.

7. Conclusion

As AI systems become more agentic, alignment can no longer be treated only as a property of final outputs. Systems that plan, write and retrieve memory, call tools, and coordinate across modules can pass the benchmarks used to evaluate them while becoming hard to inspect, interrupt, and redirect. Output evaluation remains necessary but is no longer sufficient.

Agentic AI systems should be evaluated and governed for structural governability, not only output alignment. By structural governability we mean whether consequential coordination remains observable, interruptible, attributable, and steerable at the seams between components before irreversible commitments occur. This is a call to treat governability as an intervention-oriented scientific target: benchmarks should test structure under stress, papers should report control geometry, and

 doi:10.20944/preprints202605.0958.v1

deployed agents should include seam-level controls that allow intervention before local failures propagate into durable basins of behavior.

We have deliberately not argued that structural governability solves alignment, that the evidence ladder is ready to certify systems, or that structural analysis should displace behavioral evaluation — those are stronger claims than the evidence supports. What we do argue is that the absence of structural evaluation is a gap the field should close, and that closing it is tractable with existing methods and targeted new work. If agentic systems mediate what users retrieve, remember, delegate, and act upon, structural opacity threatens not only reliability but cognitive integrity. Treating governability as a research target is a precondition for alignment evidence remaining meaningful as agentic systems grow in capability and reach.

References

1. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. ReAct: Synergizing Reasoning and Acting in Language Models. *arXiv preprint arXiv:2210.03629* **2022**.
2. Schick, T.; Dwivedi-Yu, J.; Dessì, R.; Raileanu, R.; Lomeli, M.; Zettlemoyer, L.; Cancedda, N.; Scialom, T. Toolformer: Language Models Can Teach Themselves to Use Tools. In Proceedings of the Advances in Neural Information Processing Systems, 2023.
3. Park, J.S.; O'Brien, J.C.; Cai, C.J.; Morris, M.R.; Liang, P.; Bernstein, M.S. Generative Agents: Interactive Simulacra of Human Behavior. In Proceedings of the Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, 2023.
4. Packer, C.; Wooders, S.; Lin, K.; Fang, V.; Patil, S.G.; Stoica, I.; Gonzalez, J.E. MemGPT: Towards LLMs as Operating Systems. *arXiv preprint arXiv:2310.08560* **2023**.
5. Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Li, B.; Zhu, E.; Jiang, L.; Zhang, X.; Zhang, S.; Liu, J.; et al. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. *arXiv preprint arXiv:2308.08155* **2023**.
6. Hong, S.; Zhuge, M.; Chen, J.; Zheng, X.; Cheng, Y.; Zhang, C.; Wang, J.; Wang, Z.; Yau, S.K.S.; Lin, Z.; et al. MetaGPT: Meta Programming for a Multi-Agent Collaborative Framework. In Proceedings of the International Conference on Learning Representations, 2024.
7. Xu, W.; Liang, Z.; Mei, K.; Gao, H.; Tan, J.; Zhang, Y. A-MEM: Agentic Memory for LLM Agents. *arXiv preprint arXiv:2502.12110* **2025**.
8. La Malfa, E.; La Malfa, G.; Marro, S.; Zhang, J.M.; Black, E.; Luck, M.; Torr, P.; Wooldridge, M.J. Large Language Models Miss the Multi-Agent Mark. In Proceedings of the NeurIPS 2025 Position Paper Track, 2025.
9. Cemri, M.; Pan, M.Z.; Yang, S.; Agrawal, L.A.; Chopra, B.; Tiwari, R.; Keutzer, K.; Parameswaran, A.; Klein, D.; Ramchandran, K.; et al. Why Do Multi-Agent LLM Systems Fail? *arXiv preprint arXiv:2503.13657* **2025**.
10. Moshkovich, D.; Mulian, H.; Zeltyn, S.; Eder, N.; Skarbovsky, I.; Abitbol, R. Beyond Black-Box Benchmarking: Observability, Analytics, and Optimization of Agentic Systems. *arXiv preprint arXiv:2503.06745* **2025**.
11. Montes, G.A. Personal Intelligence: Toward a User-Governed Preference Substrate for the Age of Agentic AI. *Preprints.org* **2026**. https://doi.org/10.20944/preprints202603.1627.v1.
12. Schreiber, T. Measuring Information Transfer. *Physical Review Letters* **2000**, *85*, 461–464.
13. Barrett, A.B.; Barnett, L.; Seth, A.K. Multivariate Granger Causality and Generalized Variance. *Physical Review E* **2010**, *81*, 041907.
14. Bailey, M.; Schneider, S. When Wholes Resist Decomposition: A Spectral Measure of Epistemic Emergence. *Entropy* **2026**, *28*, 380. https://doi.org/10.3390/e28040380.
15. Arthur, W.B. Competing Technologies, Increasing Returns, and Lock-In by Historical Events. *The Economic Journal* **1989**, *99*, 116–131.
16. David, P.A. Clio and the Economics of QWERTY. *American Economic Review* **1985**, *75*, 332–337.
17. Pierson, P. Increasing Returns, Path Dependence, and the Study of Politics. *American Political Science Review* **2000**, *94*, 251–267.
18. Freidlin, M.I.; Wentzell, A.D. *Random Perturbations of Dynamical Systems*, 3 ed.; Springer, 2012.
19. He, P.; Dai, Z.; He, B.; Liu, H.; Tang, X.; Lu, H.; Li, J.; Ding, J.; Mukherjee, S.; Wang, S.; et al. TRAJECT-Bench: A Trajectory-Aware Benchmark for Evaluating Agentic Tool Use. *arXiv preprint arXiv:2510.04550* **2025**.
20. Liu, S.; Chen, Y.; Krishna, R.; Sinha, S.; Ganhotra, J.; Jabbarvand, R. Process-Centric Analysis of Agentic Software Systems. *arXiv preprint arXiv:2512.02393* **2025**.

 doi:10.20944/preprints202605.0958.v1

21. Greshake, K.; Abdelnabi, S.; Mishra, S.; Endres, C.; Holz, T.; Fritz, M. Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. In Proceedings of the Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security, 2023.
22. Sunil, B.D.; Sinha, I.; Maheshwari, P.; Todmal, S.; Malik, S.; Mishra, S. Memory Poisoning Attack and Defense on Memory Based LLM-Agents. *arXiv preprint arXiv:2601.05504* **2026**.
23. Zou, W.; Dong, M.; Romero Calvo, M.; Chang, S.; Guo, J.; Lee, D.; Niu, X.; Ma, X.; Qi, Y.; Jiang, J. Poison Once, Exploit Forever: Environment-Injected Memory Poisoning Attacks on Web Agents. *arXiv preprint arXiv:2604.02623* **2026**.
24. Sydow, J.; Schreyögg, G.; Koch, J. Organizational Path Dependence: Opening the Black Box. *Academy of Management Review* **2009**, *34*, 689–709.
25. Shi, J.; Yuan, Z.; Tie, G.; Zhou, P.; Gong, N.Z.; Sun, L. Prompt Injection Attack to Tool Selection in LLM Agents. *arXiv preprint arXiv:2504.19793* **2025**.
26. Liu, Y.Y.; Slotine, J.J.; Barabasi, A.L. Controllability of Complex Networks. *Nature* **2011**, *473*, 167–173.
27. Montes, G.A. Morphology, Seam Topology, and Temporal Scaffolding in Complex Systems. *Preprints.org* **2026**. https://doi.org/10.20944/preprints202604.0448.v1.
28. Kalman, R.E. Mathematical Description of Linear Dynamical Systems. *Journal of the Society for Industrial and Applied Mathematics, Series A: Control* **1963**, *1*, 152–192.
29. Pasqualetti, F.; Zampieri, S.; Bullo, F. Controllability Metrics, Limitations and Algorithms for Complex Networks. *IEEE Transactions on Control of Network Systems* **2014**, *1*, 40–52.
30. Dong, L.; Lu, Q.; Zhu, L. AgentOps: Enabling Observability of LLM Agents. *arXiv preprint arXiv:2411.05285* **2024**.
31. AlSayyad, A.; Huang, K.Y.; Pal, R. AgentTrace: A Structured Logging Framework for Agent System Observability. *arXiv preprint arXiv:2602.10133* **2026**.
32. Nian, Y.; Yuan, A.; Zhang, H.; Li, J.; Zhao, Y. Auditable Agents. *arXiv preprint arXiv:2604.05485* **2026**.
33. Hadfield-Menell, D.; Dragan, A.; Abbeel, P.; Russell, S. The Off-Switch Game. In Proceedings of the Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, 2017.
34. Battiston, F.; Amico, E.; Barrat, A.; Bianconi, G.; Ferraz de Arruda, G.; Franceschiello, B.; Iacopini, I.; Kéfi, S.; Latora, V.; Moreno, Y.; et al. The Physics of Higher-Order Interactions in Complex Systems. *Nature Physics* **2021**, *17*, 1093–1098. https://doi.org/10.1038/s41567-021-01371-4.
35. Mezić, I. Spectral Properties of Dynamical Systems, Model Reduction and Decompositions. *Nonlinear Dynamics* **2005**, *41*, 309–325.
36. Rowley, C.W.; Mezić, I.; Bagheri, S.; Schlatter, P.; Henningson, D.S. Spectral Analysis of Nonlinear Flows. *Journal of Fluid Mechanics* **2009**, *641*, 115–127.
37. Schmid, P.J. Dynamic Mode Decomposition of Numerical and Experimental Data. *Journal of Fluid Mechanics* **2010**, *656*, 5–28.
38. Ostrow, M.; Eisen, A.; Kozachkov, L.; Fiete, I. Beyond Geometry: Comparing the Temporal Structure of Computation in Neural Circuits with Dynamical Similarity Analysis. In Proceedings of the Advances in Neural Information Processing Systems, 2023.
39. Redman, W.T.; Gordon, J.; Mianjy, P.; Nakkiran, P.; Koyejo, S. Identifying Equivalent Training Dynamics. In Proceedings of the Advances in Neural Information Processing Systems, 2024.
40. Qian, W.; Zavatone-Veth, J.A.; Ruben, B.S.; Pehlevan, C. Partial Observation Can Induce Mechanistic Mismatches in Data-Constrained Models of Neural Dynamics. In Proceedings of the Advances in Neural Information Processing Systems, 2024, Vol. 37.
41. Staufer, L.; Feng, K.; Wei, K.; Bailey, L.; Duan, Y.; Yang, M.; Ozisik, A.P.; Casper, S.; Kolt, N. The 2025 AI Agent Index: Documenting Technical and Safety Features of Deployed Agentic AI Systems. *arXiv preprint arXiv:2602.17753* **2026**.