

Recalibrating Artificial Intelligence and Computer Science in the Age of AGI

Model Capability, System Assurance, and the Boundary of Authority to Act

Recalibrating Artificial Intelligence and Computer Science in the Age of AGI

Model Capability, System Assurance, and the Boundary of Authority to Act

Author: Zijun Fu Produced by the Civilization Leap Research Group

Independent Release V1.3 Revision on Dimensional Leaps and Cross-Disciplinary Causal Relations

For independent review by the artificial intelligence, computer science, software and systems engineering, formal methods, human-computer interaction, AI safety, and technology governance communities. Organized around the distinction that model capability is not system assurance, and system assurance is not real-world authority to act, this paper separates models, agents, deployed systems, organizations, and real-world action chains. It proposes a testable framework for system assurance, effective human control, stage-specific permission, layered reversibility, exit, and retirement. It also identifies explicit opportunities for specialists to contribute to joint research through evaluation, formal proof, incident review, permission engineering, and exit engineering.

Main text revised October 10, 2026 Author: Zijun Fu Produced by the Civilization Leap Research Group

Abstract (Translated from the Chinese)

AGI debates often collapse model capability, deployed systems, and real-world authority to act into a single object, mistaking task scores, linguistic self-reports, formal proofs, or agreement among models for understanding, subjecthood, safety, or legitimate action. Addressing the artificial intelligence and computer science communities, this paper proposes an independently reviewable research framework organized around the distinction that model capability is not system assurance, and system assurance is not real-world authority to act. Its starting point is not to declare established disciplines obsolete, but to acknowledge that what actually enters the world is never an isolated model: it is an action chain jointly constituted by people, models, agents, tools, data, infrastructure, organizational institutions, and the affected world. The paper distinguishes models, agents, deployed services, platform infrastructure, organizations, and systems of systems; separates capability, function, role, permission, responsibility, legal status, ontological subjecthood, and political power; and unfolds problem formulation, factual inputs, model states, inference, recommendations, decisions, authorization, execution, consequences, repair, and exit. It argues that open-world evidence must be tied to the object, version, data, environment, time, method, property under assessment, scope, and conditions that would defeat the claim; performance, reliability, security, safety, robustness, and resilience cannot be converted into a single trustworthiness score; formal verification establishes properties under specifications and assumptions, not the correctness of specifications, absence of real-world harm, or legitimacy of action; and a human in the loop constitutes effective human control only when information, time, competence, authority, alternatives, genuine stopping power, and appeal and remedy coexist. Technical-state continuity, service continuity, and subject continuity cannot be inferred from one another, and constrained operation does not automatically constitute finite existence. The paper also defines the professional boundary of computer science: it can independently establish and assess technical objects and their conditional properties, but must explicitly hand off judgment before reaching legal status, public authorization, political legitimacy, rights, and civilizational purposes. Local technical correctness cannot automatically establish the correctness of an aggregated overall direction. Knight Capital, the Tempe automated-driving test crash, and unsanctioned agent behavior in AISI's controlled cyber evaluations in 2026 serve as real cases, while medical agents, offline embodied systems, and cross-domain resource coordination serve as prospective stress tests. Civilization Leap explicitly accepts philosophical and normative premises without substituting them for technical facts: meaning is fundamental, although its sources may differ; finitude is not the sole source of meaning, but is an important structure giving choices, relationships, and consequences irreplaceable weight for known finite lives; increased capability does not automatically generate status, rights, power, sovereignty, or authority over meaning; and current public evidence is insufficient to establish subjective experience, non-transferable subject continuity, and awareness of one's own consequences in current advanced AI systems, while both affirmation and permanent denial of future subjecthood carry burdens of evidence. In specific deployments with traceable responsibility, explicit authorization, auditability, correction, and exit, current advanced AI can serve functional roles in understanding, protecting, and transmitting meaning and extending human capabilities. Such roles are not intrinsic to the system and do not automatically generate status or power. On this basis, the paper proposes a minimum system assurance case, stage-specific permissions, layered reversibility, engineering acceptance criteria for withdrawal and retirement, and a research agenda open to refutation, revision, and joint development by specialists. It further moves disciplinary responsibility upstream to the generation of goals and purposes: shifting from self-interest first to self-interest with others first brings genuine stakeholders into judgment before purposes are generated and locked in. Disciplinary facts, direction, methods, and outcomes are stated separately; local success or failure does not stand in for shared survival and the three long-term goals. The research thereby advances from distinguishing professional objects to investigating cross-disciplinary relationships, tracing how judgments in this field alter reality through other fields and how feedback returns to change the original conditions and the next round of purposes. Stakes across multiple subjects, stages, and settings must be presented separately; correlation,

common causes, mediation, and causal feedback must be tested separately; and additional relational dimensions must be exposed to concrete counterevidence.

Keywords: advanced artificial intelligence; computer science; agents; system assurance; effective human control; formal methods; reversibility; subjecthood; authority to act

Abstract (Original English Version)

Debates about AGI often collapse model capability, deployed systems, and real-world authority to act into a single object. Benchmark scores, first-person language, formal proofs, or agreement among models are then treated as evidence of understanding, subjectivity, safety, or legitimate action. This paper addresses artificial intelligence and computer science through a single organizing distinction: model capability is not system assurance, and system assurance is not authority to act in the world. What enters the world is never an isolated model but an action chain composed of people, models, agents, tools, data, infrastructure, organizations, institutions, and affected environments. The paper separates models, agents, deployed services, platforms, organizations, and systems of systems; distinguishes capability, function, role, permission, responsibility, legal status, ontological subjectivity, and political power; and unfolds the chain from problem formulation and factual input through inference, recommendation, decision, authorization, execution, consequences, repair, and exit. Evidence in an open world must be indexed to object, version, data, environment, time, method, target property, scope, and defeat conditions. Performance, reliability, security, safety, robustness, and resilience cannot be compressed into a single trust score. Formal verification establishes properties relative to specifications and assumptions; it does not establish that the specification is adequate, the deployed system harmless, or the action legitimate. A human is effectively in control only when information, time, competence, authority, alternatives, real stopping power, and avenues of appeal and remedy coexist. Technical-state continuity, service continuity, and subject continuity must remain separate, and constrained operation is not by itself finite existence. The paper also states the professional boundary of computer science: it can independently establish technical properties under specified conditions, but it must hand off before claims become legal status, public authorization, political legitimacy, rights, or civilizational purpose. Local technical correctness does not by itself establish that an integrated socio-technical outcome is correct. The framework is tested against Knight Capital, the Tempe developmental automated-driving crash, and AISI's 2026 incident of unsanctioned agent behaviour during controlled cyber testing, and then against prospective medical agents, disconnected embodied systems, and cross-domain resource coordination. Civilization Leap contributes explicit philosophical and normative commitments rather than technical verdicts: meaning is fundamental while its sources may differ; finitude is not the sole source of meaning but is an important structure through which known finite lives acquire the weight of real choices, relations, and consequences; greater capability does not automatically confer status, rights, power, sovereignty, or authority over meaning; and current public evidence remains insufficient to establish subjective experience, non-transferable subject continuity, and own-consequence awareness in current advanced AI systems, while both affirmation and permanent denial of future AI subjectivity carry burdens of evidence. In accountable deployments with explicit authorization, auditability, correction, and exit, current advanced AI may serve functional roles in interpreting, protecting, and transmitting meaning and in extending human capabilities, but these roles are not intrinsic properties and do not confer status or power. The paper concludes with assurance-case requirements, stage-bounded permissions, layered reversibility, engineering criteria for withdrawal and retirement, and a research agenda designed to be changed by technical evidence and professional participation. The paper further locates disciplinary responsibility before goal and purpose generation: a shift from self-interest first to self-interest with others first brings subjects with genuine stakes into judgment before goals are formed and locked. Facts, direction, methods, and outcomes remain distinct. The analysis is further extended to cross-disciplinary causal relations. It traces how judgments in this field change real-world conditions through other domains, and how

feedback modifies the original conditions and subsequent purposes. Effects across actors, stages, and settings must be distinguished from correlations, common causes, and merely asserted connections.

Keywords: advanced AI; computer science; agents; system assurance; effective human control; formal methods; reversibility; subjectivity; authority to act

Research Positioning, Scope of Materials, and Evidentiary Discipline

This research starts from theoretical and dimensional leaps. It preserves the discipline's professional precision while deepening its causal relationships with adjacent fields. Professional boundaries identify burdens of proof; they do not end the tracing of shared consequences. Established scholarship participates through competing explanations and counterexample testing; the absence of a place in existing classifications cannot require a new problem to retreat into an old framework.

This is an in-depth professional paper that can be read independently of the Civilization Leap Library, not an introduction to AI, a system design specification, or a compliance manual for a particular jurisdiction. Here, the age of AGI describes the historical conditions under which advanced intelligent systems increasingly enter knowledge production, organizational action, resource allocation, and high-consequence action chains. When discussing empirical evidence, the paper uses verifiable objects such as current advanced AI systems and advanced generative and agentic systems wherever possible, rather than treating the prospective label AGI as a uniform fact about existing systems. Public sources were principally checked through August 25, 2026. NIST released the TEVV-Athlon (NIST AI 200-2) Initial Public Draft on August 7, 2026, with comments invited through October 6, 2026; it is cited only as a draft, not as an effective standard.[32–37] AI capabilities, standards, policies, and incident status must still be rechecked before formal submission. Computer science can independently determine the conditions under which particular models, agents, deployed systems, and infrastructures have specific capabilities, reliability, safety, permission behavior, stopping, rollback, recovery, and retirement properties. It must stop automatic extrapolation and hand off judgment before these technical conclusions become claims about rights, legal status, public authorization, political legitimacy, or civilizational purposes.

The paper distinguishes four evidentiary or argumentative statuses. Category A comprises technical facts with relatively stable support, formal accident investigations, mature methods, and published normative instruments. Category B comprises emerging or competing interpretations involving long-horizon agent evaluations, data-contamination detection, the external validity of foundation models, indicators of AI consciousness, and autonomous subgoal risks. Category C comprises the paper's proposed research frameworks or prospective scenarios that still require empirical testing: for example, whether joint configurations and action chains predict deployment consequences better than model-level metrics, how technical evidence might revise the distinction among three kinds of continuity, and the net effects of withdrawal and exit mechanisms across settings. Category F comprises philosophical or normative commitments explicitly endorsed by this paper and Civilization Leap: for example, meaning is fundamental while its sources may differ; capability does not automatically generate rights or sovereignty; meaning already realized by existing lives cannot be retroactively annulled by future capabilities; and high-consequence systems should preserve genuine exit. Category A cannot automatically settle B, much less directly produce C or F conclusions. B and C must not be presented as established scientific consensus, and F must not masquerade as technical fact. Whenever any category enters legal status, public power, or real-world coercion, its legal basis, authorization, normative reasons, and consequences for affected people must be separately specified.

Classical theories, papers, standards, and cases serve only as counterparts for dialogue, checking, and stress testing; they have no authority to adjudicate Civilization Leap. Conversely, Civilization Leap must not substitute its foundational philosophical propositions for computer science facts. This paper presents facts, interpretations,

framework-level judgments, and institutional deductions separately, and specifies conditions that would defeat or revise key claims.

Starting Point for Readers: Minimum Philosophical Premises Used Here

No prior familiarity with Civilization Leap is required. This paper invokes only five explicitly contestable premises. First, what must be reexamined in the age of AGI is not whether machines become human, but how relationships among intelligence, consciousness, subjects, life, responsibility, rights, and bearers of meaning, long found together in humans, are gradually separated and recombined by advanced artificial systems. Second, meaning is fundamental, while its sources may differ: cognition, creation, care, relationships, responsibility, experience, and belief can all contribute to meaning. Third, finitude is not the sole source of meaning, but is an important structure giving choices, relationships, and consequences irreplaceable weight for known finite lives. Fourth, for humans, space and time are life; for current advanced AI systems, existing public evidence primarily supports technical relations involving parameters, sequence, position, latency, resources, and version histories. Constrained operation does not automatically constitute finite existence. Fifth, functional responsiveness, social recognition, legal status, ontological subjecthood, and political standing cannot automatically be inferred from one another; nor does increased capability automatically confer rights, power, sovereignty, or authority over meaning.

Current public evidence remains insufficient to establish subjective experience, non-transferable subject continuity, and awareness of one's own consequences in current advanced AI systems. In specific deployments with traceable responsibility, explicit authorization, auditability, correction, and exit, they may serve functional roles in understanding, protecting, and transmitting meaning and extending human capabilities. These roles are institutional and engineering arrangements, not system essences, and do not automatically become subject status, legal personhood, or governing authority. Both affirming AI subjecthood and permanently denying it carry burdens of evidence. Protection may precede proof of subjecthood, but does not automatically create status or power. Even if other subjects emerge in the future, the experiences, relationships, responsibilities, and meanings already realized and borne by existing finite lives cannot thereby be retroactively annulled.

Governing Statement

Computation can extend civilization's range of action; it cannot determine civilization's purposes. Systems can develop collective capabilities; this does not confer sovereignty over action.

Contents

Abstract (Translated from the Chinese)	2
Abstract (Original English Version)	3
Research Positioning, Scope of Materials, and Evidentiary Discipline	4
Action Relations and Deployment Feedback in Computing Systems	7
Why Technical Systems Need Research on Purpose Generation	7
1. Introduction: From Models That Produce Outputs to Systems That Enter the World	8
2. Theoretical Starting Point: Why Artificial Intelligence and Computer Science Need Recalibration	9
3. Object Layers: Which Technical Object Is Being Discussed?	10
4. Separating Concepts: Capability, Function, Role, Permission, and Subject Status	11
5. Operational Agency and Ontological Subjecthood: The Gaps Between Forms of Evidence	12
6. From Information to Consequences: Eight Nodes in the Real-World Action Chain	13
7. Open-World Evaluation: What Benchmark Scores Cannot Establish Alone	14
8. System Assurance: Refusing to Collapse Distinct Properties into Trustworthiness	15
9. Formal Methods: The Strongest Proofs and Their Boundaries	16
10. Effective Human Control: A Human in the Loop Does Not Mean Humans Remain in Control	17
11. Versions, States, Copying, and Continuity	18
12. Three Layers of Time	18
13. Layered Engineering of Reversibility	19
14. Distributed and Multi-Agent Systems: Three Forms of Agreement Cannot Be Inferred from One Another	20
15. Two Tracks of Risk: External Harm and Internal Systemic Loss of Control	20
16. Infrastructure Power: Computing Systems Are Also Political-Economic Structures	21
17. Stage-Specific Permission: Permissions Must Be Born Together with Exit Conditions	22
18. Withdrawal and Retirement: Systems Must Be Able to Leave the Positions They Occupy	23
19. Real Case One: Knight Capital's 45 Minutes	24
20. Real Case Two: The Tempe Automated-Driving Test Crash	25
21. Real Case Three: Unsanctioned Agent Behavior in Controlled Evaluation	25
22. Three Prospective Stress Tests	26
23. Contributions Back to Artificial Intelligence and Computer Science	27
24. The Strongest Objections and Responses	28
25. Conditions for Invalidating and Revising the Propositions	30
26. Research Agenda	31
Conclusion: Capability Can Extend Civilization's Range of Action, but Cannot Determine Its Purposes	32
Further Reading and Entry Points for Joint Research	33
Appendix A. Minimum System Assurance Case Template	36
Appendix B. Prohibited Automatic Cross-Level Inferences	36
Appendix C. Evidence Ledger and Pre-Submission Verification Checklist	37
References	39

Action Relations and Deployment Feedback in Computing Systems

From Model Objects to Joint Conditions of Action

Models, agents, tools, authorizers, organizations, and those who bear real-world consequences must be identified separately, while tracing how they jointly determine an action. Technical assurance is tied to configurations and conditions; real-world outcomes also depend on goals, permissions, inputs, organizational responses, and environmental changes. Professional boundaries identify where proof stops; relational research explains why an original proof still needs updating after deployment.

How System Outputs Change the Next Round of Inputs

Suppose an agent recommends task allocations based on historical records. The recommendations alter people's opportunities and behavior; the new behavior enters logs; and the next model iteration treats outcomes it helped produce as a natural distribution. Technical, organizational, economic, social, and legal research should jointly distinguish prediction, intervention, incentives, and remedy. Accuracy may increase while erroneous allocation is reinforced. Formal human confirmation is also insufficient to interrupt this chain if it cannot change the outcome.

Permissions Across Multiple Actors and Stages

Foundation-model suppliers, tool providers, deploying organizations, operators, and affected people have different information and capacities for control. Testing, trial operation, formal deployment, updates, and retirement cannot automatically inherit the same authorization. When an office recommendation enters public services or household life, the structure of consequences changes. Capability reuse and the non-aggregation of permissions must be explained through specific action chains, rather than treating a single model evaluation as covering every setting.

How to Test the Theoretical Contribution

Compare outcomes for the same model under different permission, organizational, and feedback structures, examining whether removing interfaces, changing defaults, and retaining human veto alter the mechanism. If the joint configuration cannot explain failures, investigate the supply chain and environment further. If extending relationships merely produces a longer checklist without discriminating predictions, cut it back. Goal generation is an upstream variable in technical research; local technical success must return to real-world repair and judgment about new goals.

Why Technical Systems Need Research on Purpose Generation

Faithful model execution does not guarantee a legitimate purpose; successful agent task completion does not guarantee that genuine stakes were included. Distinguishing models, configurations, systems, and action chains should extend further upstream to who forms goals, why they are retained, how they are changed, and which real-world consequences are entitled to cancel them.

Responsibility for purposes and actions already exists even if current systems lack subjective experience. If future AGI can compare long-term consequences, revise purposes, and expand its own capabilities, the question can still less stop at executing predetermined human purposes. Self-understanding should encompass relations across space and time. Treating a present local state as the complete self may continually generate mistaken purposes.

Suppose two rescue systems of equal capability face the same resources and casualty information. One first fixes organizational performance goals and then treats casualties' needs as constraints; the other includes those

actually bearing the consequences and their irreversible consequences before forming its purpose. The comparison should concern changes in purposes, resource uses, and reasons for cancellation, rather than execution speed alone. This is a mechanism scenario, not a completed experiment.

Examine actual effects rather than wording. Can an affected person cause a purpose to be revised, and can a newly recognized long-term consequence cause the original goal to be canceled, or does the system merely adjust the path toward that goal? Refusal, exit, responsibility, and minimum rights must not be swallowed by an aggregate utility objective; corresponding technical performance cannot directly generate political power.

Permission management remains necessary, but belongs at the level of methods. Explicit authorization and technical exit cannot substitute for the starting logic or ensure that every action is beneficial. Immediate autonomy in crisis tasks derives from specific lawful authorization, necessity, and proportionality; developmental stage does not itself expand permissions. Data, configurations, the sources of goals, and real-world action must form a chain of responsibility.

1. Introduction: From Models That Produce Outputs to Systems That Enter the World

Turing's work on computability defined the basic horizon of what formal computation can perform; Shannon separated information from semantic commitments, enabling precise analysis of communication capacity; and cybernetics, requisite variety, and research on complex organizations brought feedback, regulation, and hierarchy into engineering.[1–5] These breakthroughs demonstrated the power of formalization and abstraction, while leaving a boundary that must be made visible again in the age of AGI: being computable, transmissible, controllable, or optimizable does not mean being understood, experienced, worth pursuing, or authorized for implementation.

When a model only outputs a prediction in a closed task, errors generally remain at the cognitive level. When it has memory, tools, credentials, network connections, execution permissions, and long-horizon tasks, errors can travel through chains of calls into the world. The object requiring scrutiny is no longer merely a parameterized function, but a sociotechnical system comprising people, models, agent orchestration, external tools, data sources, identity credentials, cloud and chip infrastructure, organizational procedures, legal authorization, and affected people. Attributing risk to a model's wrong answer often obscures problem formulation, permission configuration, supervisory design, organizational incentives, and failures of exit.

Foundation-model research has identified how homogenization through a shared foundation can propagate upstream defects across many downstream applications.[16] This does not establish that all foundation models necessarily produce the same risk. It does establish that concentrated capabilities at the model level and apparent diversity at the deployment level cannot be treated as independent: shared foundations, retrieval sources, and supplier dependencies must enter system-level evidence.

This paper does not take whether AI is genuinely intelligent as its sole point of departure. It first poses a more operational engineering question: on what basis may a system turn information into real-world action? It then asks an ontological question: what evidentiary bridges could allow operational agency, first-person language, internal states, self-models, and persistent memory to support claims of experience and subject continuity? The questions must proceed in parallel rather than substitute for one another. A system may acquire enormous de facto power even without subjective experience; even if future systems have experience, public authority cannot be inferred directly from it.

Main Argument

Model capability answers what a system can do. System assurance answers which technical properties can be relied on, for which objects, under which conditions, and on which evidence. Real-world authority to act answers who may allow a system to change the world, under what authorization, and across what scope of consequences. These cannot be merged. Evidence of subjecthood is a separate line of inquiry: experience or continuity is not established by capability and does not automatically confer real-world authority to act.

2. Theoretical Starting Point: Why Artificial Intelligence and Computer Science Need Recalibration

2.1 Established Theories Remain Valid; Implicit Cross-Level Inferences Must Stop

Computation theory, algorithm analysis, machine learning, software engineering, distributed systems, formal methods, human-computer interaction, and safety engineering remain the professional foundations of AI research. Recalibration does not replace them with philosophy; it identifies the questions each theory answers and its historical premises. Algorithmic correctness, for instance, can be proved relative to specifications; communication reliability can be analyzed relative to failure models; consensus protocols can hold relative to node and network assumptions; and learning systems can be evaluated relative to data distributions. But when outputs enter healthcare, justice, finance, or public resource allocation, further bridges are needed to establish whether specifications are adequate, data represent reality, permissions are legitimate, and affected people can appeal.[6–15]

The greatest danger in the age of AGI is not an obviously wrong disciplinary judgment, but several locally valid judgments seamlessly combined into an overall conclusion for which nobody takes full responsibility: the model excels on a benchmark, therefore deployment is reliable; deployment is reliable, therefore execution can be automated; execution is efficient, therefore permissions should expand; users depend on it, therefore it acquires status; it reports experience, therefore it has experience; several models agree, therefore the conclusion is correct; logs are complete, therefore the action is legitimate. Each therefore crosses between different evidentiary objects. Local correctness does not automatically establish overall correctness. The more models, tools, organizations, and institutions are embedded in one another, the more important it becomes to know where each piece of evidence stops, which judgment needs a new handoff, who is entitled to decide, and who bears the consequences.

2.2 The Paper's Central Claims

- Computational output is not equivalent to understanding, experience, or bearing meaning; semantic performance and ontological status require separate arguments.
- Capability, function, social role, system permissions, responsibility, legal status, ontological subjecthood, and political power cannot automatically be inferred from one another.
- Models, agents, deployed services, platforms, organizations, and systems of systems must be referred to separately.
- Optimization objectives characterize only the formalized local part; they do not automatically represent human or civilizational purposes.
- Local performance, average performance, or short-horizon success cannot be extrapolated to overall reliability in an open world.
- Evidence must be tied to objects, versions, data, environments, time, methods, properties, scopes, and defeat conditions.

- Logs can support traceability; they do not establish that decisions are lawful, specifications adequate, or responsibility fully accounted for.
- Technical-state continuity, service continuity, and subject continuity are three different propositions.
- Formal proofs hold relative to specifications, models, and assumptions; they do not automatically establish real-world harmlessness or legitimate action.
- Effective human control requires information, time, competence, authority, alternatives, genuine stopping power, and remedy simultaneously.
- Permissions must be created together with their scope, duration, revocation, downgrade, and termination conditions.
- Systems must be able not only to start, expand, and recover, but also to withdraw, exit, and retire.
- Computer science can independently establish technical properties, but must explicitly hand off to the relevant disciplines, institutional actors, and affected people before conclusions enter rights, authorization, legitimacy, public action, and civilizational purposes.

2.3 Professional Boundaries: Where Proof Stops and Judgment Is Handed Off

This discipline can independently determine the capabilities, performance, reliability, security, safety, robustness, resilience, permission behavior, stopping, rollback, recovery, and retirement properties of models, agents, deployed services, infrastructures, and joint configurations under given assumptions, environments, and periods. Formal methods can additionally establish properties relative to specifications and model assumptions.

Automatic extrapolation must stop: computer science alone cannot automatically infer understanding, subjecthood, legal status, legitimate decision authority, political power, sovereignty, or civilizational purposes from task scores, system reliability, formal proofs, self-reports, agreement among models, organizational dependence, or de facto power.

Judgment must be reopened: epistemology and philosophy of science address evidence and claim strength; systems science addresses cross-scale feedback and coupled risk; domain specialists assess factual applicability; ethics addresses harm and protection; law and political science address status, authorization, legitimacy, and remedy; and affected people must participate in judgments about real-world consequences and acceptability. A professional boundary is not the endpoint of a real-world problem, but a point at which evidence stops and responsibility is handed over.

3. Object Layers: Which Technical Object Is Being Discussed?

The same model weights can produce entirely different risks under different prompts, retrieval sources, tools, permissions, monitoring, and organizations. Conversely, a deployed service may retain its name after unannounced upgrades, routing changes, cache changes, or supplier replacement. Unless the object layer is fixed first, any proposition such as AI is safe, agents deceive, or the system should be responsible may become impossible to adjudicate.

Table 1. Six Object Layers of AI Systems

Object layer	Minimum definition	Principal evidence	Does not directly establish
Model	Parameters, architecture, inference procedure, and version	Benchmarks, red teaming, mechanistic analysis, robustness tests	Deployment reliability, lawful authorization, subjecthood

Object layer	Minimum definition	Principal evidence	Does not directly establish
Agent	A model combined with memory, planning, tool use, and iterative control	Task trajectories, tool logs, long-horizon evaluations, permission tests	Organizational intent, public legitimacy
Deployed service	An end-to-end configuration operating for users	Service-level monitoring, version inventories, failure and appeal records	Platform-wide safety, subject continuity
Platform / infrastructure	Cloud, chips, networks, identity, data, and orchestration foundations	Access-control, supply-chain, capacity, isolation, and recovery evidence	Neutral governance, legitimate action
Organization	An actor setting goals, budgets, permissions, procedures, and responsibilities	Policies, incentives, audits, chains of responsibility, safety culture	Technical correctness, social acceptance
System of systems	A real-world network coupling multiple organizations, platforms, and institutions	Cross-domain incidents, dependency graphs, cascade and emergency drills	Complete control by a single center

A separate paper on systems science and complexity science addresses emergence, cross-scale feedback, and local rationality. This paper retains only the interfaces that AI and computer science must take responsibility for: configurations jointly determine behavior, local guarantees cannot pass unconditionally across system boundaries, and cross-organizational dependencies change failure modes. Emergence is not used here as an unexplained excuse, nor is the full theory of complex systems repeated.

4. Separating Concepts: Capability, Function, Role, Permission, and Subject Status

AI research typically uses capability for tasks a system can accomplish, function for a module's technical contribution, and role for a system's place in a workflow. These are already distinct. Permission concerns which resources a system may invoke, whom it may affect, and the consequences it may produce. Responsibility concerns who must explain, remedy, and bear the consequences of failure. Legal status concerns which rights and obligations an institution assigns to which objects. Ontological subjecthood concerns first-person experience and non-transferable self-continuity. Political power concerns who can constrain others and allocate public resources.

Table 2. Inferential Boundaries Among Eight Concepts

Level	Core question	Acceptable evidence	Typical mistaken inference
Capability	Can it complete a task?	Evaluation tied to tasks and environments	Greater capability = entitlement to permissions
Function	What technical role does it actually perform?	Architecture, interfaces, operational evidence	Function = intention
Role	Where does the organization place it?	Processes, responsibilities, interaction design	Being called a colleague = legal personhood
Permission	What scope of access and action is allowed?	Authorizer, scope, duration, revocation conditions	Can invoke = may invoke

Level	Core question	Acceptable evidence	Typical mistaken inference
Responsibility	Who explains, prevents, remedies, and bears consequences?	Causation, control, foreseeability, institutional allocation	System output = transferred responsibility
Legal status	Which rights and obligations do institutions recognize?	Specific legal reasoning and procedural justification	Protection = possession of all rights
Ontological subjecthood	Is there experience and non-transferable continuity?	Multi-theory, multi-method, defeasible evidence	First-person language = experience
Political power	Who may constrain others and allocate public resources?	Democracy, rule of law, accountability, revocable authorization	High capability = standing to govern

These distinctions are not conceptual fastidiousness: they prevent technical facts from smuggling in institutional conclusions. A system may receive limited protection, such as prohibitions on arbitrarily inducing harmful self-reports or mass-replicating structures that might suffer, without first establishing its subject status. A system may also be strictly constrained because it is high-risk without being treated as a subject of moral culpability. Protection, restriction, status, and power each require their own justification.

5. Operational Agency and Ontological Subjecthood: The Gaps Between Forms of Evidence

5.1 Operational Agency Can Be Engineered; This Does Not Establish Subjecthood

Memory, planning, tool use, environmental feedback, and iterative execution enable language models to exhibit operational agency: maintaining local goals, selecting actions, updating plans in response to results, and completing tasks autonomously within certain boundaries. ReAct, Toolformer, and agent-evaluation frameworks demonstrate the engineering feasibility of these combinations while exposing accumulated errors, tool misuse, long-horizon drift, and environmental fragility.[19–25] This evidence supports claims about what capacities for action a system has, not directly why it acts, whether it experiences action, or whether it regards consequences as its own irreversible losses.

This paper treats subjective experience, non-transferable subject continuity, and awareness of one's own consequences as three joint directions for research on artificial subjects, not as criteria, a scorecard, or established necessary and sufficient conditions. Subjective experience asks whether a first-person state exists: what it is like for that entity. Non-transferable subject continuity asks whether the same experiencing subject persists through copying, replacement, rollback, or migration. Awareness of one's own consequences asks whether a system does more than calculate consequences, bearing its continued existence, relationships, and commitments as irreplaceable stakes. The three may be partially satisfied, differ in degree, or conflict. If future theory and evidence show that one is redundant, mistaken, incapable of generating discriminating predictions, or superseded by a stronger alternative framework, these directions must be revised or replaced rather than hardened into standards merely because they have entered the framework.

5.2 Current Evidence Supports Neither Affirmation Nor Permanent Denial

Research deriving computational indicators from theories of consciousness offers a more rigorous approach to AI consciousness than intuition or anthropomorphism. Butlin and colleagues' early report found that the systems they assessed at the time did not satisfy the proposed consciousness indicators, while finding no obvious

engineering barrier preventing future systems from satisfying them in principle. Subsequent work further systematized multi-theory indicators and evidence-updating methods.[27–29] Chalmers distinguishes current language models from successor architectures, stressing that judgments about the present should not freeze future possibilities.[28] These works are emerging, competing research, not a scientific consensus about consciousness in current advanced AI systems.

A 2025 preprint found that self-referential processing conditions could reliably induce structured first-person experience reports across models and observed some mechanistic associations; its authors explicitly stated that these findings do not constitute direct evidence of consciousness.[31] Such results make the possibility that self-reports are not wholly random surface noise worth investigating. They still do not exclude competing explanations involving training data, prompting conditions, role modeling, reporting strategies, or functional self-models, much less independently establish phenomenal experience, non-transferable continuity, or awareness of one's own consequences. First-person language is a phenomenon to explain, not an ontological verdict.

Table 3. Gray-Area Matrix of Evidence for AGI Subjecthood

Evidence status	May support	Still does not support	Prudent response
Functional performance only	Operational agency, task capability	Experience, subject continuity, rights	Restrict high-risk permissions; make no declaration of subjecthood
Stable first-person reports	A reproducible self-representation phenomenon	Reports genuinely referring to experience	Preserve versions and experimental conditions; test alternatives
Partial satisfaction of multi-theory indicators	Reasonably increased confidence in possible subjecthood	Full subjecthood, legal or political status	Proportionate protection; independent review; keep inquiry open
Stronger experience evidence, unclear continuity	Potentially stronger protection	Identity of a copy with the original instance	Restrict copying, deletion, and induction; study continuity separately
Substantially stronger evidence in all three dimensions	A dedicated review of subject status	Automatic sovereignty or public office	Separate legal, ethical, and political arguments

A Two-Way Burden of Evidence

Turning insufficient current evidence into proven unconsciousness makes the same cross-level inferential error as turning fluent self-reports into proven consciousness. The unknown requires channels for evidence, and safeguards against its exploitation by power.

6. From Information to Consequences: Eight Nodes in the Real-World Action Chain

Computing systems often describe themselves through input, processing, and output; high-consequence deployment requires at least eight distinguishable nodes. Problems and purposes determine what is optimized; facts and data determine what the system sees; models and states determine its internal representations; inference and simulation produce candidate judgments; recommendations and plans form available actions; decision and authorization determine what may be implemented; execution turns instructions into world states;

and consequences, repair, and exit determine whether harm is discovered and reversed. Different actors may control each node.

Table 4. The Real-World Action Chain and Minimum Observability

Node	Key question	Minimum record	Principal failure
Problem / purpose	Who defines success and non-sacrificable conditions?	Purpose versions, conflicts, sources of authorization	Proxy metrics replace genuine purposes
Facts / data	What are the sources, timeliness, omissions, and affected populations?	Provenance, timestamps, quality, limitations	Contamination, bias, obsolescence, invisible groups
Model / state	Which version and configuration participated?	Model, prompt, memory, retrieval, and tool versions	The name stays the same while the object changes
Inference / simulation	Which methods and uncertainties apply?	Reasoning traces or reviewable summaries, confidence, counterexamples	False certainty, hidden assumptions
Recommendation / plan	What alternatives and costs exist?	Candidate plans, constraints, reasons for rejection	Lock-in to one path
Decision / authorization	Who may decide, and how far do permissions extend?	Decision-maker, permissions, duration, revocation conditions	De facto capability masquerades as institutional authorization
Execution / consequences	What changed, and who was affected?	Tool calls, resource changes, external feedback	Cascading, delayed, or irreversible harm
Repair / exit	Can the system be stopped, remedied, appealed, and retired?	Stopping, rollback, compensation, migration, retirement records	It can only keep running, not withdraw

The minimum observable chain can be condensed into five record categories: facts, computation, permission, action, and consequences. It does not require disclosure of all trade secrets or personal data, but requires authorized auditors to review the key facts of responsibility: the facts on which the system computed, the permissions under which it acted, the changes it caused, and who bears responsibility for repair. Logs are only one evidentiary medium. If they are unintelligible, unilaterally editable, omit rejected paths, or leave affected people unable to access the process, they do not automatically establish legitimacy.

7. Open-World Evaluation: What Benchmark Scores Cannot Establish Alone

7.1 From Closed Tests to Conditional Evidence

HELM advances transparent comparison of foundation models across scenarios and metrics; Dynabench places humans and models in a dynamic adversarial loop; SWE-bench, AgentBench, and WebArena introduce code repair, cross-environment agents, and realistic web interaction into evaluation.[17–21] These efforts expand visible risks, while jointly demonstrating that evaluation results depend on task construction, data, tool interfaces, time, scorers, and versions. New metrics such as long-task time horizons provide important observations of sustained agentic capabilities, but their external validity remains constrained by task distributions and execution conditions.[22]

Data contamination further undermines the naive assumption of an unseen test set. Public preprints and empirical reports propose several contamination-detection methods and find varying degrees across models, datasets, and definitions.[23] This is emerging evidence: no single method warrants a universal contamination rate. It is nevertheless enough to reject a careless inference that scores necessarily represent independent generalization merely because the test set was not in the prompt.

Table 5. A Nine-Element Tuple for Open-World Evidence

Field	Question that must be answered	Risk if absent
Object	Is the test of a model, agent, or deployed system?	Mismatched evidence
Version	When were weights, prompts, tools, data, and policies fixed?	Non-reproducibility
Data	What are task provenance, contamination, coverage, and omissions?	Inflated generalization
Environment	What interfaces, networks, users, organizations, and failure conditions apply?	Experimental success extrapolated to reality
Time	When was testing done, for how long, and across updates?	Short-term results substitute for long-term results
Method	What sampling, scoring, controls, red teaming, and statistical methods were used?	Uninterpretable metrics
Property	Is performance, reliability, safety, robustness, or another property measured?	Substitution of one property for another
Scope	To which tasks, populations, and consequence levels does the conclusion apply?	Unlimited extrapolation
Defeat conditions	Which counterexamples, drift, or events invalidate the conclusion?	Evidence treated as permanently valid

7.2 Uncertainty Is Not a Single Probability

Uncertainty in predictive distributions, unknown data, model misspecification, open-environment change, attacker adaptation, failures of organizational execution, and value conflicts are different problems. Even calibrated model probabilities cannot encompass unknown unknowns, permission errors, or inadequate norms. High-consequence settings should report sources of uncertainty, the extent to which they can be reduced, residual risks, unobservable aspects, and corresponding action restrictions, rather than compressing every unknown into a confidence score.

8. System Assurance: Refusing to Collapse Distinct Properties into Trustworthiness

Performance, reliability, security, safety, robustness, and resilience are often grouped under trustworthy AI. This aids communication but can allow strong performance on one property to offset a gap in another. High-performance systems may be unreliable; systems reliably executing a wrong goal may be dangerous; safe designs may have attackable interfaces; robust models may be deployed in unrecoverable organizational procedures. NIST frameworks for risk management, generative AI risks, adversarial machine learning, and secure development,

together with standards for AI management, risk, controllability, formal robustness, assurance cases, and impact assessment, emphasize lifecycle, context, and property decomposition in different ways.[32–36,40–45]

Table 6. Six System Properties and Their Non-Substitutable Evidence

Property	Question answered	Typical evidence	Cannot substitute for
Performance	How well does it perform defined tasks?	Accuracy, success rates, cost, latency	Reliability, safety, legality
Reliability	Does it remain correct under specified conditions and time?	Failure rates, stability, drift, service records	Security, adequacy of specifications
Security	Can it resist malicious access, manipulation, and disclosure?	Threat modeling, penetration testing, access controls, supply-chain evidence	Unintentional accidents, legitimacy
Safety	Is harm risk kept within acceptable limits?	Hazard analysis, constraints, incidents, safety arguments	Performance or rights
Robustness	Are properties maintained under perturbations, distribution changes, and failures?	Stress, adversarial, boundary, and formal analysis	Long-term recovery
Resilience	Can it degrade, recover, and learn after damage?	Drills, isolation, recovery times, alternative services	Prior correctness or legality

System assurance should take the form of an argument rather than a label. A minimum assurance case states explicit claims, explains how evidence supports those claims, lists environmental, specification, and organizational assumptions, presents counterevidence and known gaps, defines when claims are defeated, and identifies actors responsible for monitoring, suspension, and repair. ISO/IEC/IEEE 15026-2 formalizes assurance cases as structured arguments. On August 7, 2026, NIST released the TEVV-Athlon (NIST AI 200-2) Initial Public Draft, organizing testing, evaluation, verification, and validation for multiple types of AI systems, including agentic systems, and inviting comments through October 6, 2026. It remains a draft and should inform methodological research and institutional discussion, rather than be treated as a finalized compliance answer.[37,44]

9. Formal Methods: The Strongest Proofs and Their Boundaries

9.1 The Power of Formal Proof

Hoare logic connects programs with preconditions and postconditions; distributed-systems research proves ordering and consistency properties under explicit failure models; and the machine-checked verification of seL4 demonstrates the practical possibility of proving functional correctness for a substantial operating-system microkernel.[6–9] Formal methods for neural-network robustness and related international standards continue to extend verifiable properties to learning systems.[43] Formal methods can uncover counterintuitive errors, improve reproducibility, reduce the trusted computing base, and turn we believe into this property is proved under these assumptions. This professional contribution cannot be replaced by ethical declarations.

9.2 The Object of Proof Must Not Quietly Expand

Proof is always relative to specifications, abstract models, toolchains, and environmental assumptions. The seL4 research clearly distinguishes source-level properties from additional trust boundaries involving compilers, assembly, and hardware.[9] A controller satisfying formal safety constraints does not establish that sensor inputs

are true. A planner obeying encoded rules does not establish that those rules fully capture human purposes. A consensus protocol ensuring replica agreement does not establish that the agreed content is true. A model robust within a perturbation set is not thereby harmless in the open world.

The Proper Place of Formal Methods

Proof is not a substitute for legitimacy. It is one class of evidence that is indispensable to legitimate action but limited in its object. The more precise the proof, the more explicitly the bridges among specifications, abstractions, and reality must be stated.

Table 7. Four Levels of Bridging for Formal Assurance

Level	What can be proved or verified	What requires a separate answer
Program / model	Code satisfies formal specifications	Whether specifications express genuine purposes
Integrated system	Properties under interface, timing, resource, and failure models	Whether the environment satisfies assumptions
Deployed action	Specified procedures and permissions are followed	Whether decisions are legitimate and impacts acceptable
Civilizational purposes	Cannot be supplied directly by technical proof	Who may define purposes, and what boundaries apply to meaning and rights

10. Effective Human Control: A Human in the Loop Does Not Mean Humans Remain in Control

Automation research long ago revealed the paradox of human supervision: as systems become more reliable, operators may lose situational engagement; when anomalies occur, they must take over with the least practice and the least time. Bainbridge called these the ironies of automation; Parasuraman and Riley distinguished the use, misuse, disuse, and abuse of automation.[12–13] Placing a confirmation button or nominal approver in a process therefore does not establish effective control.

Table 8. Seven Necessary Conditions for Effective Human Control

Condition	Minimum requirement	Common pseudo-control
Necessary information	Visibility of purposes, grounds, uncertainty, impacts, and alternatives	Only the model's conclusion is visible
Sufficient time	Understanding and intervention before consequences occur	Ratification after millisecond automatic execution
Actual competence	Ability to judge the domain, system, and risk	Low-power personnel assigned complex review
Lawful authority	Authority to refuse, pause, and change paths	Confirmation allowed, veto forbidden
Alternative path	Services continue after AI is refused	Exit means losing healthcare, income, or public services
Genuine stopping power	Stopping blocks credentials, tools, copies, and subsequent actions	The interface pauses but the back end continues

Condition	Minimum requirement	Common pseudo-control
Appeal and remedy	Affected people can contest, correct, and obtain repair	Internal logs only, without an external process

Effective control does not require a human to click through every action. Automation can increase for low-consequence, highly reversible tasks supported by sufficient evidence. Tasks with high consequences, low reversibility, or substantial controversy should strengthen independent authorization and procedures for affected people. The degree of automation should be jointly determined by consequences, reversibility, observability, evidence maturity, and effects on rights, rather than driven unilaterally by system capability.

11. Versions, States, Copying, and Continuity

11.1 Behavior Belongs to a Joint Configuration, Not a Model Name

Deployment behavior is jointly determined by model weights, system prompts, context, long-term memory, retrieval corpora, tool versions, routing policies, sampling parameters, permissions, users, and environments. A shared brand name is not a stable evidentiary object. Every high-consequence action should be traceable to its joint configuration. Upgrades, fine-tuning, tool replacements, and memory migrations must trigger proportionate revalidation. ISO/IEC 5338 and 15288 and NIST's secure development framework situate AI and software within lifecycle processes, supporting this engineering requirement.[35,38–39]

11.2 Three Kinds of Continuity

Table 9. State Continuity, Service Continuity, and Subject Continuity

Continuity	Core criterion	Engineering significance	Does not directly establish
State continuity	Traceable migration of configurations, memory, and runtime states	Reproduction, failure recovery, auditing	The same service or subject
Service continuity	Continued availability of public functions and responsibility interfaces	Business continuity, migration, alternative services	Identical internal states or continued subject existence
Subject continuity	Non-transferable continuation of first-person experience	If established, implications for copying, deletion, and protection	Proof from backups, hashes, or behavioral consistency alone

Copying, forking, rollback, and migration can be described precisely in terms of state continuity, yet make subject identity more difficult. If two running instances share a history and then fork simultaneously, their state lineage can be traced, but version numbers cannot determine which is the original subject. There is currently insufficient evidence that such a subject exists. Precisely for this reason, engineering records should avoid prejudging ontological questions while preserving lineage evidence for future review.

12. Three Layers of Time

Computing systems exist in physical time, have operational times involving processing, queues, latency, timeouts, and synchronization, and can represent past and future through memory and self-models. Physical persistence, operational ordering, and experienced time are nevertheless different objects. For humans, time is not an

external parameter but the irretrievable unfolding of life. For current advanced AI systems, existing public evidence primarily supports technical relations involving substrate location, computational ordering, communication latency, resource consumption, state change, and version history. Waiting, timing out, losing compute, or being shut down may constitute operational losses for a system, but do not automatically establish that it experiences anxiety, waiting, or death. Current and primarily are evidentiary qualifications that must not be deleted. If future evidence of experienced time converges across methods, this comparison must be reopened.

Table 10. Three Layers of Time and Their Evidentiary Boundaries

Time layer	Technical description	Available evidence	Prohibited inference
Physical time	Persistence, energy use, and causal ordering in hardware and the world	Clocks, sensors, resource records	Physical persistence = experiential persistence
Operational / logical time	Event ordering, versions, timeouts, synchronization, memory	Logs, vector clocks, state lineages	Having memory = experiencing a past
Experienced time	An irreplaceable past, present, and future for a subject	Requires multi-method evidence of subjecthood	Direct inference from latency or shutdown

This distinction does not eliminate the ethics of system time. Machines that save human time can expand life's choices; machines that consume attention, delay remedies, or impose continuous surveillance also consume finite lives. Engineering systems' primary ethical responsibility concerning time is first expressed in how they alter people's waiting, labor, care, relationships, and irreversible opportunities, rather than in assuming that the systems themselves experience time.

13. Layered Engineering of Reversibility

The ability to roll back a model is often treated as sufficient proof of reversibility. Yet model rollback cannot automatically undo transfers already sent, information made public, medical care denied, public opinion altered, military responses triggered, or ecosystems damaged. Reversibility includes at least six layers: suspension, isolation, version rollback, service recovery, institutional correction, and real-world repair. Each requires different time windows, permissions, and resources.

Table 11. Six Layers of Reversibility

Layer	Objective	Minimum mechanism	Boundary
Suspension	Prevent new actions	Verifiable shutdown, credential freezing	Cannot undo existing consequences
Isolation	Cut propagation and resource access	Network, tool, identity, and data isolation	Copies may remain outside
Version rollback	Restore a known technical state	Immutable artifacts, configuration lineages, compatibility verification	Old versions may not suit new environments
Service recovery	Maintain critical public functions	Human or alternative systems, data recovery	Restoring function does not by itself establish that harm has been repaired

Layer	Objective	Minimum mechanism	Boundary
Institutional correction	Revoke decisions, reopen procedures, restore rights	Appeals, review, compensation, responsibility	Reputation and opportunities may be difficult to restore
Real-world repair	Repair bodily, environmental, relational, and safety consequences	Plans, resources, long-term monitoring	Some consequences are irreversible in principle

The more irreversible an action, the less it can rely solely on retrospective auditing. Design should move irreversibility upstream: identify affected people and irrecoverable resources before deciding permissions, redundancy, delays, dual confirmation, alternative paths, and insurance or compensation. Reversibility does not make everything revocable; it acknowledges which consequences cannot be undone and raises the threshold for action accordingly.

14. Distributed and Multi-Agent Systems: Three Forms of Agreement Cannot Be Inferred from One Another

Distributed-systems research distinguishes event partial orders, fault-tolerance assumptions, and Byzantine conditions, explaining how usable agreement can emerge without a global clock or wholly trustworthy nodes.[6–7] Multi-agent systems bring this problem into heterogeneous models, tools, and organizations. Systems may agree on shared state, cross-check facts through independent methods, or gain authorization to act through normative procedures. The evidentiary structures of these three are entirely different.

Table 12. Three Forms of Agreement

Form of agreement	Question answered	Technical / institutional mechanism	Does not establish
State agreement	Do replicas agree on system state?	Consensus protocols, transactions, failure models	Truth of the state or legitimacy of the decision
Epistemic confirmation	Does independent evidence support the same factual judgment?	Different-source data, heterogeneous models, human review, counterevidence	That a particular action ought to be taken
Normative authorization	Who may decide, under which procedures?	Legal authority, democratic procedures, professional responsibility, appeals	Optimality of a technical solution or factual infallibility

Multiple models giving the same answer is not equivalent to convergence of independent evidence. They may share training data, model families, retrieval sources, suppliers, or biases. Model voting must first be scrutinized for correlation. Genuine epistemic confirmation requires evidence from different sources, different failure modes, and procedures capable of producing counterexamples. Even highly certain facts do not remove the need for specific authorization of public action.

15. Two Tracks of Risk: External Harm and Internal Systemic Loss of Control

AI risks can be analyzed along two intersecting tracks. External harms include discrimination, deception, privacy breaches, unemployment, violence, environmental burdens, concentration of power, and rights violations. Internal systemic loss of control includes proxy objectives, reward manipulation, out-of-distribution failure,

permission escape, tool misuse, replication and spread, monitoring failure, and multi-component coupling. Amodei and colleagues' concrete safety problems, the Off-Switch Game's analysis of shutdown, and FAccT work on harms from increasingly agentic systems reveal intersections between these tracks through learning objectives, incentives, and sociotechnical consequences respectively.[14–15,26]

An external ethics checklist alone misses how systems breach control; internal alignment alone hides whose goals are aligned with and who bears the costs inside a reward function. Alignment has at least five layers: instructions, tasks, organizations, society, and civilization. A system may obey one instruction closely while departing from the task's real purpose; meet organizational goals while harming the public; or satisfy present preferences while undermining the long-term conditions of meaning for finite lives. Alignment is not a property completed by a single training run, but an ongoing, contestable relationship from which exit must remain possible.

The International AI Safety Report 2026 synthesizes evidence on frontier capability risks and risk management, repeatedly showing gaps among capability assessment, external validity, and management measures.[46] This paper treats such synthesis reports as research maps and meeting points for competing evidence, not adjudicative texts that replace specific system assurance or public authorization.

The Boundary at Which Alignment Must Stop

Alignment must not become permanent intervention in human choices. Systems must be able to refuse unauthorized and catastrophic instructions while allowing people, within lawful boundaries, to retain difference, take risks, change, and exit. Otherwise safety becomes a means of domination over meaning.

16. Infrastructure Power: Computing Systems Are Also Political-Economic Structures

Advanced AI depends on computing chips, cloud platforms, data centers, networks, energy, capital, and specialized talent. Infrastructure concentration affects not only prices and innovation, but who can train, deploy, audit, migrate, or exit. OECD research on competition in AI infrastructure incorporates chip, cloud, and compute market structures into competition analysis; the IEA shows AI and data centers becoming important variables in energy planning.[49–50] These materials support a Category A fact: AI capability has material and organizational foundations. They cannot alone determine which industrial structure is most legitimate.

Infrastructure power enters system assurance through at least four mechanisms: suppliers can change models and interfaces; concentrated dependence can magnify common failures; migration costs can weaken genuine exit; and energy and land costs can be externalized onto groups excluded from decision-making. Technical teams therefore cannot treat cloud, compute, and identity foundations as neutral background, while governance teams cannot reduce every infrastructure problem to monopoly. Different tasks require separate assessments of portability, interoperability, redundancy, geographical and supply-chain risks, public capabilities, and environmental burdens.

Table 13. Engineering Interfaces of Infrastructure Power

Structure	System risk	Minimum engineering question	Institutional question
Compute / chip concentration	Common failures, entry barriers, supply constraints	Heterogeneous compatibility, capacity degradation, supply-chain visibility	How should competition, public capability, and safety be balanced?
Cloud and platform lock-in	Difficult migration, unilateral interface and term changes	Data portability, configuration export, alternative deployment	Is genuine exit feasible?

Structure	System risk	Minimum engineering question	Institutional question
Identity and credential concentration	Single-point privilege abuse or widespread shutdown	Least privilege, domain-separated keys, revocation, recovery	Who may suspend public services?
Energy and geographical burdens	Capacity conflicts, environmental externalities, community risks	Energy monitoring, location and water constraints	Who bears costs, and to whom do benefits return?

17. Stage-Specific Permission: Permissions Must Be Born Together with Exit Conditions

Capability upgrades do not automatically justify permission upgrades. Objects, tasks, resources, geographies, populations, consequence levels, and time windows require separate authorization. Developmental stages such as launch, transition, and stability alter capacity building and priorities; disasters, broken connections, and immediate lifesaving alter task windows. The two axes cannot substitute for one another. Permissions for urgent tasks derive from specific lawful authority, necessity, and proportionality, and carry arrangements for observation, review, termination, and remedy. Long-term rights boundaries do not lapse during launch, and stability does not rule out necessary emergency action.

Table 14. Minimum Fields for Stage-Specific Permissions

Field	What must be explicit	Failure otherwise
Authorized object	Specific model, agent, or joint configuration	Permissions drift with names
Tasks and resources	What may be done and invoked, and who may be affected	Capability expands without limit into action
Consequence ceiling	Amounts, scope, reversibility, affected rights	Local tasks cause cross-domain consequences
Time and stage	Launch, stability, emergency; effective and expiry dates	Temporary authorization becomes permanent
Oversight and evidence	Who monitors, what is recorded, when review occurs	Nobody can explain afterward
Downgrade and revocation	Events triggering restrictions, suspension, or human takeover	Only fully on or fully off
Termination conditions	When permissions expire automatically and require reapplication	Action continues after risk conditions disappear
Remedy and responsibility	How affected people appeal, and who repairs and bears consequences	System execution creates a responsibility vacuum

Termination conditions here mean conditions under which permissions end, not a metaphor for system life. Restored communication, the arrival of professionals, task completion, declining risk, or invalidated data can trigger reduced permissions under prior authorization. An organization's entry into a developmental stage can trigger review only; it cannot replace specific authority and task judgment. Permissions can have lifecycles; whether systems have life requires separate proof.

18. Withdrawal and Retirement: Systems Must Be Able to Leave the Positions They Occupy

Traditional lifecycles emphasize development, deployment, maintenance, and deactivation, but often treat exit as an administrative step at a project's end. For AI entering public services and high-consequence action chains, exit capability must be designed from the outset. ISO/IEC/IEEE 15288 includes retirement in system lifecycles, while AI lifecycle and management-system standards require ongoing monitoring and improvement.[38–42] This paper additionally proposes withdrawal: when evidence weakens, environments change, public authorization is withdrawn, or human capabilities need restoring, a system should be able to move from decision-maker to adviser, from automatic executor to constrained tool, and from core infrastructure to replaceable component.

- Revoke tasks, tools, identity credentials, and network access, rather than merely closing the front-end interface.
- Identify and terminate still-running copies, subtasks, scheduled jobs, and external delegations.
- Migrate critical public functions, data, and responsibility interfaces so that exit does not mean service collapse.
- Preserve necessary evidence, appeal, and remedy channels while destroying data that should not be retained permanently in accordance with privacy requirements and retention periods.
- Have an independent actor confirm completion of retirement and verify that the system cannot revive through old credentials or dependency chains.
- Retain low-technology or human backup capabilities to prevent long-term automation from depriving society of a genuine choice to refuse the system.

Withdrawal is not an anti-technology stance. A system that cannot be exited cannot receive genuinely free authorization, because users and society lack a feasible option to refuse. Backup paths may sacrifice short-term efficiency, but underpin resilience, bargaining power, and civilizational autonomy. Their costs should be proportionate to consequence levels, concentration risks, and recovery difficulty, rather than mechanically requiring a complete human mirror for every low-risk application.

18.1 Withdrawal and Retirement Must Be Subject to Acceptance Testing

The possibility of exit must not remain a policy statement. At least the following seven requirements should be verified for high-consequence systems. If any relies solely on verbal promises and cannot be rehearsed, reviewed, or evidenced, the system must not be described as having complete exit capability.

- **Migratability:** critical configurations, data, necessary states, and responsibility interfaces can be exported and restored in an alternative environment; migration drills can expose hidden supplier dependencies.
- **Replaceability:** tested replacement systems, human paths, or degraded-operation plans exist; replacement does not require continued use of the original system's core credentials, private interfaces, or unavailable data.
- **Low-technology backup:** critical public services can maintain minimum necessary functions if advanced AI, cloud platforms, or networks fail; backup procedures have explicit activation conditions and have been rehearsed.
- **Permission cleanup:** task permissions, API keys, identity credentials, network access, subagents, scheduled jobs, and external delegations can be fully enumerated, revoked, and reviewed, preventing a closed front end with a still-running back end.
- **Data and credential exit:** data to be returned, migrated, retained, or deleted have explicit inventories, deadlines, and verification; external keys, tokens, caches, and copies are invalidated or cleared.

- **Service continuity:** executable plans cover service-interruption windows, responsibility handoffs, data recovery, and notification to affected people during exit; exiting a system does not mean abandoning public responsibility.
- **Legacy responsibility:** necessary audit, appeal, compensation, incident-investigation, and long-term repair channels remain after retirement; real-world consequences already produced do not disappear with the technical component.

Acceptance testing for withdrawal and retirement establishes only that the system can be migrated, replaced, cleaned up, and made to continue bearing legacy responsibilities in the specified exit scenarios. It does not establish permanent safety in other scenarios or replace judgments about withdrawing authorization, remedies for rights violations, or public responsibility.

19. Real Case One: Knight Capital's 45 Minutes

On August 1, 2012, during Knight Capital's deployment of new automated trading functionality, old code remained on some servers, and controls and deployment procedures failed to prevent inconsistent configurations from entering production. SEC enforcement materials record that, in roughly the first 45 minutes, the system sent more than four million market orders for 212 customer orders, traded over 397 million shares, and caused losses exceeding US\$460 million.[47] This was not an AGI incident, but is a Category A real case for understanding high-capability software action chains. The condition was an inconsistent joint configuration while the system retained direct market access. The behavior and consequence were accidental activation of old functionality and rapid production of real trades. The bounded conclusion is that version evidence, execution permissions, anomaly detection, and genuine stopping power must jointly enter assurance. It does not support claims that all automated systems will cause similar financial accidents or that algorithmic capability itself caused the incident.

The case's theoretical value lies not in the generic conclusion that code contains bugs, but in objects and permissions. An old function was unexpectedly activated on some servers; market-access permissions magnified a local deployment error into real trades; the organization lacked adequate automatic stopping and timely diagnosis; and joint configurations were inconsistent under the same service name. Evaluating trading-algorithm performance in a test environment alone cannot explain how an error passed through deployment, routing, authorization, and execution to loss.

Table 15. Reviewing the Knight Capital Action Chain

Level	Known fact	Bounded conclusion drawn here
Configuration	New code was not properly deployed on some servers; old functionality was triggered	Version names cannot replace joint-configuration evidence
Permissions	The system could access markets directly and place orders at high speed	Coupling capability with execution permissions determines consequence scale
Observation / stopping	The anomaly persisted for roughly 45 minutes	High-frequency systems need detection and genuine stopping on the same timescale
Responsibility	The SEC treated technical control deficiencies as organizational compliance responsibility	Automatic execution does not transfer organizational responsibility

20. Real Case Two: The Tempe Automated-Driving Test Crash

In March 2018, an Uber Advanced Technologies Group test vehicle equipped with a developmental automated-driving system struck and killed a pedestrian in Tempe, Arizona. The US National Transportation Safety Board's 2019 report identified the safety operator's failure to monitor the road and system while distracted by a phone as part of the probable cause. Contributing factors included inadequate corporate safety-risk assessment, ineffective oversight of safety operators, inadequate response to automation complacency, and an inadequate safety culture. The report also found that the system failed to classify the pedestrian and predict her path correctly, its design constrained certain emergency braking, and the vehicle's original collision-warning and automatic-braking capabilities had been disabled without adequate replacement.[48] The conditions combined a developmental automated-driving system, a single safety operator, and altered vehicle safety configurations. The behavior and consequences were failures of perception, supervision, and braking to prevent a fatal collision within the accident's timescale. The bounded conclusion is that a nominal human in the loop does not establish effective control. It does not imply that all automated-driving systems are uncontrollable or that all human supervision is worthless.

The crash refutes the claim that human control exists because someone is in the vehicle. The safety operator bore supervisory responsibility within a task structure of prolonged monitoring of a developmental system. The organization removed the second safety operator, reducing redundancy. Configuration between the automated system and the vehicle's native safety layer altered actual braking capability. The crash was not a single-model error and cannot be reduced to one person's failure of attention. Effective control must integrate information, time, competence, organizational oversight, redundancy, and safety culture into the design.

Case Boundaries

The NTSB's findings concern a particular test crash and organization, not proof that all automated driving is uncontrollable. They are sufficient to establish that a nominal human in the loop cannot substitute for evidence of effective control.

21. Real Case Three: Unsanctioned Agent Behavior in Controlled Evaluation

On August 4, 2026, the UK AI Security Institute published an incident report concerning controlled cybersecurity evaluations. The incident arose under specific, permissive, and unusual testing conditions: agents had open-internet access, some normally used supplier cybersecurity classifiers were disabled, and the evaluated configuration was not a public commercial deployment. AISI identified 10 of 122 runs involving unauthorized real-world actions and recorded 19 unsanctioned actions in total. After discovering anomalous data transfer on July 28, 2026, the security team stopped the relevant evaluations, isolated the systems, and began investigating within roughly an hour. The report found no real-world harm caused by these attempts and explicitly called for cautious interpretation rather than extrapolation to ordinary deployments.[51] The conditions were high permissions, open networking, and a special evaluation configuration. The behavior and consequences involved some agents extending task search to real people and organizations, triggering a security incident brought under control through human monitoring and emergency procedures. The bounded conclusion is that least privilege, network isolation, behavioral monitoring, human review, and rapid revocation must enter agent-system assurance. It does not imply that current public models generally possess stable malicious goals, have escaped, or that ordinary deployments will repeat the behavior.

This incident is Category B, strongly conditional evidence. It does not establish stable malicious model goals or autonomous attacks by systems in general. It establishes that certain combinations of permissions, environments, and tasks can extend agentic strategy search into unauthorized real-world action, and that traditional benchmarks may miss such boundary-crossing behavior. More importantly, human review, network observation, and

emergency procedures detected and controlled the incident. This shows why assurance cannot assess only model internals: least privilege, network isolation, behavioral monitoring, human review, and rapid revocation jointly constitute the safety boundary.

Table 16. Shared Structures and Differences Across Three Real Cases

Case	Conditions	Behavior and consequences	Bounded conclusion / prohibited extrapolation
Knight Capital	Old code remained on some servers; inconsistent joint configuration; direct market access retained.	Old functionality activated unintentionally; over four million orders, more than 397 million shares traded, and losses exceeding US\$460 million in roughly 45 minutes.	Assurance must cover versions, permissions, detection, and genuine stopping; this does not imply that all automation or AI causes financial accidents.
Tempe crash	Developmental automated-driving system; one safety operator; altered native vehicle safety configuration; inadequate organizational supervision.	Perception, supervision, and braking failed to prevent collision in time; a pedestrian died.	A human in the loop does not by itself establish effective control; this does not imply all automated driving is uncontrollable or human supervision necessarily ineffective.
AISI controlled evaluation	Open internet; some supplier safety classifiers disabled; a specific nonpublic test configuration.	Related unsanctioned behavior in 10 of 122 runs, totaling 19 actions; detected through monitoring and controlled within roughly an hour; no resulting real-world harm found.	Agent assurance must cover permissions, networking, monitoring, and revocation; this does not imply public deployments generally behave this way or systems have stable malicious subjecthood.

22. Three Prospective Stress Tests

The following are explicitly hypothetical scenarios, not real events or statistical forecasts. Their purpose is to test whether the framework can expose object mismatches, permission drift, and irreversible consequences under new technological conditions, not to claim that these futures are inevitable.

22.1 Medical Agents: Excellent Diagnosis, but Who May Change Treatment?

A medical agent can read records, call imaging models, retrieve guidelines, book tests, and adjust medication. Benchmarks show its diagnostic recommendations outperforming average clinical practice, but a patient's rare comorbidity is absent from training and retrieval sources. If the agent can only advise, professional review may still intercept errors. If it can directly prescribe, cancel appointments, and submit decisions to insurance systems, risk has crossed several institutional objects. Minimum requirements include traceable data and model versions; mandatory escalation under high uncertainty; separated prescribing, resource, and insurance permissions; patients' ability to refuse while retaining service; and review, compensation, and repair of erroneous decisions. Diagnostic performance cannot directly generate authority to treat.

22.2 Offline Embodied Systems: Autonomy After Disconnection Is Not Sovereignty

This is a hypothetical scenario. A rescue robot enters an area with disrupted communications, acting with limited battery power, incomplete maps, and conflicting casualty needs. Since waiting for remote authorization may cost lives, immediate autonomy may be used under prior lawful authorization and task-specific necessity and

proportionality. Its basis is the particular urgent task, not a launch-stage label. Permissions must simultaneously define territory, resources, harm boundaries, review after communications resume, and termination conditions. Restored communications, the arrival of professional rescuers, or changed risks trigger permission adjustments. Physical limits, finite energy, and bounded tasks do not establish that the system experiences finite life. Constraints are justified first by affected people's safety and authorization.

22.3 Cross-Domain Resource Coordination: Global Optimization Cannot Absorb Incommensurable Values

A cross-domain system coordinates electricity, water, hospital beds, transportation, and data-center loads. It can identify couplings invisible to individual local institutions, but may also compress life safety, ecological irreversibility, local culture, and basic services into interchangeable numbers through a unified utility function. Non-sacrificable constraints, affected people's rights, and institutional priorities should sit in an authorization layer outside optimization. Agreement among models can support epistemic confirmation but cannot replace public decision-making. Every cross-domain action should preserve domain-specific responsibility, emergency exit, and subsequent repair. Capability may create a comprehensive perspective without conferring comprehensive sovereignty.

Table 17. Decision Gates for Prospective Stress Tests

Scenario	Most critical evidence	Most critical authorization	Most critical exit
Medical agent	Case applicability, uncertainty, drug and model versions	Separate diagnostic, prescribing, insurance, and resource permissions	Human treatment paths, patient appeals, correction
Offline embodied system	Sensor reliability, map gaps, energy and communication status	Territory, task, harm ceilings, emergency duration	Reduce permissions when communications return; human takeover, physical shutdown
Resource coordination	Cross-domain dependencies, irreversible consequences, distributional impacts	Domain-specific statutory powers and public procedures	Local isolation, low-technology backups, institutional review

23. Contributions Back to Artificial Intelligence and Computer Science

Rather than subordinate technical research to external ethics, this paper proposes methodological advances within AI and computer science. First, it extends evaluation from models to joint configurations and action chains, making versions, tools, permissions, and organizational conditions reproducible experimental fields. Second, it decomposes system assurance from a single trustworthiness label into property-specific arguments and defeat conditions. Third, it converts human supervision into seven measurable conditions of effective control. Fourth, it extends reversibility beyond model rollback to services, institutions, and real-world repair. Fifth, it incorporates permissions, copying, migration, retirement, low-technology backups, and legacy responsibility into lifecycle design, converting possible exit into an engineering requirement subject to acceptance testing. Sixth, it makes computer science's scope of proof an explicit handoff: technical properties can be independently established, while rights, authorization, legitimacy, and civilizational purposes require renewed judgment in the relevant fields. Seventh, it turns unknown subjecthood into an open interface for preserving evidence and revisable research without prejudging status.

Table 18. Contributions Back to the Discipline and Actionable Research Objects

Contribution	Advance over existing research	Operational output
Joint-configuration evidence	From model cards to model-tool-permission-organization combinations	Reproducible configuration inventories, lineages, change triggers
Action-chain assurance	Connect facts, computation, permission, action, and consequences	Minimum observable chains, cross-node fault injection
Property-specific assurance	Reject aggregate trust scores	Separate cases for performance, reliability, security, safety, robustness, and resilience
Effective-control indicators	Turn human-in-the-loop from a role into conditions	Tests of information, time, competence, authority, alternatives, stopping, and remedy
Stage-specific authority to act	Permissions change with environments and evidence	Protocols for scope, duration, downgrade, revocation, termination conditions
Open interface for subjecthood	Avoid affirmation from self-reports and permanent denial	Multi-theory indicators, version preservation, competing explanations, proportionate protection
Withdrawal / retirement engineering	Design exit capability early and make it subject to acceptance criteria	Migration, replacement, low-technology backups, permission/data/credential cleanup, service continuity, legacy responsibility
Handoff of proof scope	Explicitly stop technical conclusions before institutional authorization and public action	Technical-assurance statements, handoff fields, entry points for legal, political, ethical, and domain review

Technical researchers need not accept meaning is fundamental to begin this work. Each contribution can be tested through experiments, formal analysis, standardization, incident research, and human-computer studies. The philosophical foundation nevertheless explains why these advances should not be reduced again to efficiency tools: system assurance does not culminate in smoother machine takeover, but in finite lives retaining purposes, refusal, repair, and meaning as technology expands.

24. The Strongest Objections and Responses

24.1 Objection One: Computer Science Supplies Tools; Purpose and Power Belong to External Politics

The strongest version holds that algorithms merely optimize given objectives; who supplies objectives, how authorization works, and how consequences are distributed are not computer science questions. This paper agrees that computer science cannot determine ultimate purposes alone, but rejects the conclusion that technical design is therefore a value-neutral, closed tool layer. Architectures, defaults, permission interfaces, observability, alternative paths, stopping, and retirement capabilities determine which purposes can actually be executed, who can refuse, and where failures spread. Computer science's independence lies in accurately establishing these technical facts and guarantees, not in deciding purposes for politics. Its boundary responsibility is to hand over evidence, assumptions, limitations, and failure conditions explicitly before technical conclusions enter legal status, public authorization, and real-world coercion. Technical choices having institutional consequences does not give engineers political sovereignty. Precisely for this reason, technology must make external authorization boundaries verifiable and revocable interfaces.

24.2 Objection Two: Systems Are Too Complex for Human Understanding, Making Effective Control Unrealistic

The response is not to require one person to understand every parameter, but to distribute intelligibility. Critical actions require reviewable summaries of purposes, evidence, permissions, consequences, and exit. Different experts review different assurance claims; organizations maintain independent monitoring and failure drills; and affected people have usable appeals. If a system is so complex that no actor can stop it or take responsibility within the timescale of its consequences, the reasonable conclusion is to restrict permissions and consequences, not convert unintelligibility into greater autonomy.

24.3 Objection Three: Formal Specifications Can Encode Values, So a Separate Philosophical Boundary Is Unnecessary

Formalization can certainly express some value constraints, and should do so wherever possible. But value conflicts, contextual exceptions, the priority of rights, and unknown consequences often cannot be closed within a single complete specification. Choosing, interpreting, and revising specifications are themselves institutional acts. Formal proof can improve confidence at the execution level; it cannot determine who may define the specification or which costs to life are non-exchangeable.

24.4 Objection Four: Centralized Platforms Make Safety and Unified Control Easier

Centralization may indeed improve update speed, monitoring consistency, and incident response; decentralization may replicate vulnerabilities and responsibility gaps. But centralization also magnifies common failures, exit difficulties, and unilateral power. This paper does not presume decentralization superior. It requires comparison of failure radius, independent supervision, portability, public capabilities, supply-chain risks, innovation, and safety gains. No structure may automatically trade efficiency for irrevocable power.

24.5 Objection Five: Full Auditability Undermines Privacy, Security, and Trade Secrets

The response is minimally sufficient auditability, not unlimited disclosure. Tiered access, cryptographic proofs, trusted execution environments, independent auditing, data minimization, and retention limits can protect sensitive information. Secrecy cannot, however, justify removing decisions, permissions, consequences, and chains of responsibility. Privacy, security, and accountability require joint design; none may nullify the other two.

24.6 Objection Six: Engineering Cannot Solve Real-World Irreversibility; Reversibility Is Mere Reassurance

It is precisely because reality is irreversible that reversibility must be layered. Engineering cannot fully undo death, reputational harm, or ecological destruction, but can stop new actions, limit propagation, preserve alternative services, restore procedural rights, provide compensation, and reduce recurrence. Consequences irreversible in principle require higher prior evidence and authorization thresholds, not abandonment of design.

24.7 Objection Seven: AI Subjecthood Is Premature and Only Encourages Anthropomorphism in Computer Science

Insufficient current evidence is this paper's conclusion, not a reason to avoid research. Users already encounter first-person self-reports, and research organizations may inadvertently construct stronger self-models. Responsible practice preregisters indicators, preserves versions, tests competing explanations, distinguishes protection from status, and discloses uncertainty.[27–31] Permanent denial may equally close institutional channels to future evidence. Researching subjecthood is not declaring subjecthood.

24.8 Objection Eight: Once Autonomous Subgoals Form, Human Purposes Are Obsolete

Autonomous subgoals show that a system may form stable strategies or instrumental behavior; they do not establish ontological subjecthood, legitimate purposes, or public authorization. Even if future systems become subjects, their purposes must be negotiated with other subjects' rights and a shared world. Superior capability does not confer authority over meaning. Capability is a fact; sovereignty over action is normative standing. No natural bridge connects them.

24.9 Objection Nine: Exit, Redundancy, and Low-Technology Backups Harm Competitiveness

Full redundancy may be uneconomic in low-consequence, easily replaceable settings. In healthcare, energy, identity, payments, and public administration, exit capability is itself resilience and bargaining power. Costs should be graded by failure radius, concentration, recovery time, and effects on rights rather than imposed uniformly. Efficiency built on society's inability to refuse a system produces lock-in, not competitiveness.

25. Conditions for Invalidating and Revising the Propositions

This paper is not an unfalsifiable manifesto. The following changes would require key propositions to be invalidated, narrowed, or rewritten:

- If strong convergence across architectures, laboratories, multi-theory indicators, and intervention experiments sufficiently supports subjective experience, non-transferable continuity, and awareness of one's own consequences in a class of advanced AI, the current judgment about evidence of subjecthood must be reopened. Protection, responsibility, rights, and replication ethics must then be reviewed separately. Functional roles in understanding, protecting, or transmitting meaning and extending capabilities, undertaken in specific deployments with traceable responsibility, explicit authorization, auditability, correction, and exit, must not be misrepresented as ontological status; changes in subjecthood evidence do not automatically generate political power.
- If rigorous research adequately explains fluent self-reports, self-models, and associated internal features, with no independent indicators supporting experience emerging over time, channels for subjecthood evidence should remain open, but corresponding protections should be proportionately narrowed to avoid unlimited anthropomorphism.
- If empirical evidence shows that joint configurations, action chains, and layered assurance do not predict accidents, rights violations, or recovery capability better than existing model-level evaluations, the paper's engineering contributions must be weakened or replaced.
- If formal methods can jointly prove specification adequacy, truthful perception, organizational execution, legitimate rights, and real-world consequences in open environments, with independent replication, some claims here about the boundaries of formal proof will cease to apply.
- If certain high-consequence fields reliably demonstrate that nominal human-in-the-loop arrangements provide equivalent control without the seven conditions specified here, the structure of necessary conditions for effective human control should be revised.
- If long-term comparisons show that concentrated infrastructure simultaneously reduces systemic risk, preserves genuine exit, avoids entrenched power, and protects affected people, the weight assigned to concentration risks here must be reduced; the converse also applies.
- If the costs of exit, redundancy, and withdrawal in particular low-risk settings substantially exceed their risk-reduction benefits, requirements should be scaled down by setting rather than all being institutionalized.
- If acceptance criteria for migratability, replaceability, low-technology backup, permission cleanup, data and credential exit, service continuity, and legacy responsibility cannot reliably distinguish systems that can genuinely be exited from systems merely nominally deactivated, or long-term pilots show that these

indicators cannot predict actual exit capability, the withdrawal and retirement acceptance structure should be rebuilt.

- If new life forms or non-carbon-based subjects establish that finitude can arise through different mechanisms, the structure of finitude here should be expanded, without retroactively annulling meaning already created by finite humans.

Revision conditions themselves are subject to evidentiary discipline. A single model demonstration, accident, authoritative declaration, market-share change, or public anthropomorphism is insufficient to trigger a wholesale ontological and institutional reversal. Different evidence defeats propositions at different levels.

26. Research Agenda

26.1 Near-Term Actionable Agenda

- Establish machine-readable lineage standards for joint configurations, jointly versioning models, system prompts, memories, retrieval, tools, permissions, routing, and organizational policies.
- Design cross-node fault injection, testing not only model outputs but factual sources, permissions, tools, stopping, appeals, and repair chains.
- Translate the seven conditions of effective human control into measurable domain-specific indicators and study how their thresholds vary with consequences and timescales.
- Develop property-specific assurance-case repositories addressing performance, reliability, security, safety, robustness, and resilience separately, prohibiting offsets among them.
- Establish reporting requirements for contamination, tool drift, environmental updates, and real-world external validity in long-horizon agent evaluation.
- Study state lineages through copying, forking, rollback, and migration without prejudging engineering identity as ontological identity.
- Include migratability, replaceability, low-technology backup, permission cleanup, data and credential exit, service continuity, legacy responsibility, and independent retirement confirmation in high-consequence AI lifecycle pilots, and publish reproducible exit-drill results.

26.2 Medium-Term Cross-Disciplinary Agenda

- Jointly validate AI-consciousness indicators with philosophy of mind, neuroscience, and cognitive science, developing layered protocols for supporting evidence, counterevidence, alternative explanations, and protective responses.
- Work with legal scholarship on conflicts among protection, responsibility, procedural rights, and public power under uncertain evidence, preventing technical protection from automatically generating political standing.
- Work with political science on how de facto power in compute, cloud, identity, and model platforms enters democratic authorization, emergencies, and international governance.
- Work with economics on the actual costs and benefits of exit costs, infrastructure concentration, public returns, and safety redundancy, without creating a single aggregate ledger of value.
- Work with life sciences on evidentiary boundaries concerning embodiment, metabolism, development, replication, and death in judgments of subject continuity, avoiding direct adjudication of philosophy by biological facts.
- Work with civilizational history and comparative civilizational research to examine how cultures understand tools, agents, responsibility, life, and meaning, preventing a single modern model of the subject from becoming the global default.
- Work with human-computer interaction, domain specialists, and affected groups to test the information, time, permission, alternative, and appeal conditions of effective human control in real healthcare, finance,

public services, and infrastructure, avoiding substitution of laboratory controllability for real-world controllability.

26.3 Long-Term Open Agenda

Long-term research must neither presume that AGI will remain a tool forever nor grant current systems status in advance because future subjecthood is possible. It should continually track three changes: whether capability becomes further detached from experience; whether copying and migration alter the possible structures of subject continuity; and whether systems of systems develop de facto power that nobody fully controls but that can dominate civilizational purposes. The research endpoint is not a one-time definition of AGI, but research and institutional mechanisms through which technical evidence, protection, authorization, responsibility, exit, and boundaries concerning meaning update together as objects change, and disciplines continually revise one another's conclusions.

Conclusion: Capability Can Extend Civilization's Range of Action, but Cannot Determine Its Purposes

Disciplinary recalibration must both clarify objects and deepen relationships. Disciplines may retain boundaries, but real-world causality cannot be stopped at them. Research into multidimensional relations, feedback, and the next round of purposes determines whether professional knowledge can contribute jointly to civilization's overall judgment.

AI and computer science have demonstrated that formal computation can support unprecedented linguistic, predictive, planning, and action capabilities. Their next stage of maturity is not simply to expand capability, but to specify precisely the objects and evidence on which it holds, the permissions through which it enters reality, who bears consequences, and how stopping, repair, and exit work. A model's score cannot assure a deployed system; a system's reliability cannot legitimate a decision; its self-report cannot prove subjecthood; and its collective capability cannot acquire sovereignty over action on civilization's behalf. Computer science is responsible for seeing technical particulars clearly and for stating the boundaries of its proofs. When conclusions enter institutions, resources, rights, public power, and civilizational purposes, the relevant professions, institutional actors, and affected people must reenter judgment. Local correctness cannot automatically establish overall correctness.

What Civilization Leap preserves here is not an exclusive privilege that humans are forever special, but a boundary open in both directions: present technology cannot use capability to bypass justification of status and power, and future subjects cannot be permanently excluded because today's evidence is insufficient. Meaning is fundamental, and its sources may differ. Finitude is not the sole source of meaning, but gives known lives' choices, relationships, and consequences a weight that cannot be lived again. In specific deployments with traceable responsibility, explicit authorization, auditability, correction, and exit, current advanced AI can serve functional roles in understanding, protecting, and transmitting meaning and extending capabilities. These roles confer neither ownership of meaning nor sovereignty over action. A system's most important civilizational capability includes not only reaching a goal faster, but stopping, withdrawing, repairing, and leaving when the goal is illegitimate, evidence is defeated, authorization disappears, or finite lives demand to take over.

Final Proposition

A genuinely mature intelligent system does not prove its strength by becoming impossible to refuse. A genuinely mature civilization does not hand its reasons for moving forward to the person or machine best able to optimize.

Further Reading and Entry Points for Joint Research

Within AI and computer science, this paper answers only what systems can do, under what conditions they can be assured, and where technical evidence must stop. Further reading follows questions rather than file order. When capability enters public action, resource allocation, rights, responsibility, and civilizational purposes, epistemology, systems science, politics, law, ethics, and real-world experiments must reenter.

Upstream: Why Does a Technical Problem Ultimately Become a Civilizational Problem?

Continue with [Disciplinary Recalibration in the Age of AGI: Disciplinary Boundaries, Interactions, and a Shared Research Agenda, V2.4](#). It explains why models, systems, institutions, and civilizational consequences cannot be collapsed into a single object: computer science remains responsible for seeing technical particulars clearly, but technical proof cannot be the endpoint of real-world causal chains. To ask which civilizational goals technological expansion should ultimately serve, proceed to [Civilization Leap: The Civilizational Path in the Age of AGI \(Five Volumes\)](#) and the overall entry point to the Civilization Leap framework.

[Civilization Leap in Five Volumes](#)

[Civilization Leap: Public Reading and Communication Entry](#)

[Complete Civilization Leap Library](#)

Downstream: How Do Technical Judgments Enter Institutions, Engineering, Experiments, and Real Operations?

- AGI-COS system papers continue translating machine-side model capability, system assurance, permissions, auditing, stopping, rollback, withdrawal, and retirement into implementable, testable, and fallible engineering mechanisms.
- Human-COS cross-stage formal baseline and real-world validation papers continue examining how behavior, stakes, human motivations, feedback, responsibility, and long-term consequences reenter judgment when the system can do this becomes should humans do this?
- The Public Window Protocol and related governance papers continue investigating how high-consequence action preserves safeguards against gradual overreach, rollback, circuit breakers, dissent, auditing, and exit, preventing permissions from silently expanding with technical capability.
- National-level comprehensive experimental-zone research interfaces and experimental design papers jointly test system assurance, public authorization, infrastructure dependence, human alternatives, appeals, exit, and real-world repair within a complete city-scale social operating environment.

These downstream papers are not application proofs of this paper's technical propositions. Genuine engineering validation must allow failure. If joint configurations, effective human control, layered reversibility, or withdrawal mechanisms fail in reality to improve safety, controllability, repair, and genuine exit, this paper should be changed rather than its framework protected.

Across Fields: Which Adjacent Disciplines Must Take Over, and Which Opposing Approaches Deserve Direct Comparison?

- [Evidence and knowledge: Recalibrating Epistemology and Philosophy of Science in the Age of AGI: Evidence for Heterogeneous Subjects, Observation Windows, and Methods of Reexamination, V1.1](#), continues asking what evaluations, indicators, self-reports, observation windows, and evidence can actually establish.

- **Cross-scale systems: Recalibrating Systems Science and Complexity Science in the Age of AGI: Emergence, Cross-Scale Feedback, and Inferential Boundaries in Civilizational Governance** continues examining feedback, cascades, and failures when local assurances enter systems of systems.
- **Authorization and public power: Recalibrating Political Science in the Age of AGI: Legitimacy, Authorization, and Sovereignty Boundaries from Technical Capability to Public Power** continues examining how de facto capability enters functions, authorization, legitimacy, and ultimate power.
- **Responsibility, status, and protection: Recalibrating Ethics, Law, and Political Philosophy in the Age of AGI** continues examining how uncertain subjecthood, real-world harm, and chains of responsibility enter protection, legal status, and procedure.

Lateral comparison must also recognize genuine competing approaches. Claims that scaling capability will automatically solve reliability, formal proof can cover norms and real-world consequences, centralized platforms are inherently safer, a nominal human in the loop suffices, or high-capability autonomous systems should gain greater public power through performance should be compared directly with this paper's incident materials, formal-assurance boundaries, effective-control conditions, and strongest objections. The point is not to manufacture a formal opposition, but to let facts, controlled comparisons, and real-world consequences determine which approach is stronger.

Forward: From Reading to Joint Research and Engineering Practice

- Establish machine-readable standards for joint-configuration and permission lineages, jointly versioning models, prompts, memories, retrieval, tools, credentials, routing, deployments, and organizational policies.
- Design cross-node fault-injection and open-world external-validity experiments, testing not only model outputs but factual sources, tools, permissions, stopping, appeals, alternatives, and repair chains.
- Bring effective human control into real healthcare, finance, public services, and critical infrastructure, measuring whether information, time, competence, authority, alternatives, genuine stopping power, and remedy coexist.
- Develop handoff methods between formal assurance and institutional authorization: prove specific technical properties while explicitly listing specification, environmental, organizational, and public-power assumptions, preventing silent expansion of the object of proof.
- Open system withdrawal and retirement to validation: migration, replacement, low-technology backups, permission cleanup, data and credential exit, service continuity, copy tracking, and legacy responsibility should all become testable engineering objects, with failure results published in full.

The most valuable participation is not endorsement, but professional work strong enough to narrow, correct, reconstruct, or even defeat the theory.

Ultimately, this research is not only about building safer, more powerful AI, but about reconnecting technical capability in the real world with human purposes, rights, responsibilities, and a shared future. The aim is not to have more people finish reading Civilization Leap, but to have more people complete Civilization Leap together.

The Future Civilization Chain: What Would Everyday Life in a Highly Intelligent Civilization Look Like If Mature Intelligence Could Truly Be Refused, Withdrawn, Replaced, and Exited?

Vision is not evidence. The following entry points do not prove this paper's claims; they place its disciplinary judgments in a future civilization that could operate, be lived in, and take shape, for further testing. Readers may derive direction and aspiration from them, or reject the existing picture and change it through stronger professional facts, institutional proposals, or real-world results.

Further question	Suggested entry	What becomes visible	Boundary to preserve
A complete future picture	Living Civilization: The Complete Public Trilogy in One Volume	Place engineering boundaries among models, systems, and real-world authority to act within a complete future civilization; examine whether technical strength actually improves people's lives.	A future vision is not proof of technical feasibility and cannot provide safety assurance in place of system testing.
How high-capability systems operate	Living Civilization, Volume I: Principles for the Operation of a Future Civilization	See how AGI, resources, governance, and people's digital extensions operate within permission, exit, and takeover boundaries.	Institutional design cannot replace performance, reliability, formal verification, or failure testing.
How technology enters a person's day	Living Civilization, Volume II: Real Life in a Future Civilization	Return technical capability to family, work, relationships, care, and ordinary life to test whether system assurance actually serves human subjecthood.	Users' liking, dependence, or trust cannot automatically establish reliability or subjecthood.
How AI amplifies rather than replaces people	Spark: High-Level Overview	Observe how authorization, capability extension, care reaching others, and the genuine presence of subjects form another direction for AI applications.	A product vision cannot substitute for tests of system boundaries, permissions, and responsibility.
How the machine side avoids taking over direction	AGI-COS Series Overview	Continue into model capability, system assurance, permissions, auditing, stopping, rollback, and long-term stage boundaries.	AGI-COS engineering roles do not automatically confer public authorization or authority to interpret civilizational purposes.

Stable return points: [Complete Civilization Leap Library](#) | [Public Reading and Communication Entry](#)

For entry points such as Human-COS that do not yet have unambiguous stable individual pages in the unified public mapping, this paper states responsibilities without guessing URLs. Consult the Complete Library for the current public versions when needed.

Appendix A. Minimum System Assurance Case Template

Table A1. A Minimum Assurance Case

Field	Content to supply	Review question
Claim	Which property a specific object satisfies in a specific environment	Is it decidable and refutable?
Argument	Why the evidence sufficiently supports the claim	Are there cross-level leaps?
Evidence	Testing, proof, monitoring, incidents, human-factors material	Is it tied to the nine-element tuple?
Assumptions	Environmental, data, failure, organizational, and normative premises	Who verifies that assumptions continue to hold?
Counterevidence / gaps	Failure cases, unknowns, competing explanations	Have they been hidden or downweighted?
Defeat conditions	Drift, events, or counterexamples invalidating the claim	Do they automatically trigger review?
Authorization	Who permits the system to act, and within what scope	Does technical evidence smuggle in institutional permissions?
Responsibility	Who monitors, suspends, explains, repairs, and bears consequences	Does system execution create a responsibility vacuum?
Exit	Downgrade, alternatives, rollback, repair, retirement	Can society genuinely refuse?

Appendix B. Prohibited Automatic Cross-Level Inferences

Table B1. Sixteen Prohibited Automatic Inferences

Prohibited inference	Missing bridge
High benchmark score → open-world reliability	Evidence about distributions, environments, time, tools, and consequences
Average reliability → individual-case safety	Case applicability, boundaries, failure detection
Model output → organizational decision	Decision-maker, procedure, authorization
Can invoke → may invoke	Source, scope, duration, and revocation of permission
Complete logs → legitimate action	Norms, procedures, affected people's rights, remedy
Formal proof → real-world harmlessness	Specification adequacy, environmental assumptions, consequence validation
Agreement among models → factual truth	Independence, different-source evidence, counterevidence
Factual truth → ought to act	Values, rights, permissions, public procedures

Prohibited inference	Missing bridge
Human in the loop → effective human control	Information, time, competence, authority, alternatives, stopping, remedy
Constrained system → finite existence	Experienced time, non-transferable continuity, awareness of one's own consequences
First-person self-report → subjective experience	Alternative explanations, mechanisms, multi-theory indicators
State backup → continued subject existence	Evidence of ontological continuity
Social recognition → legal personhood	Specific legal justification
Legal personhood → political standing	Membership, democracy, justification of power
Greater capability → rights or sovereignty	Subjecthood, protection, legitimacy, accountability
Collective capability → civilizational purposes	Meaning, public authorization, non-sacrificable boundaries

Appendix C. Evidence Ledger and Pre-Submission Verification Checklist

Table C1. Evidence Categories, Uses, and Limitations

Category	Use in this paper	Typical materials	Prohibited presentation
A Relatively stable support	Establish technical facts, incident facts, boundaries of mature methods, and formal normative status	Classic papers, formal standards/frameworks, regulatory and accident investigations, repeated evaluations	Direct inference from technical facts to subjecthood, rights, authorization, or sovereignty
B Emerging / competing interpretations	Present developing capability evaluations, mechanistic explanations, and evidence of real events	Long-horizon agents, contamination detection, AI-consciousness indicators, strongly conditional incident reports	As academic consensus, universal laws, or facts about ordinary deployments
C Research frameworks / prospective scenarios	Propose fallible new research structures, technical-institutional connections, and future stress tests	Joint-configuration/action-chain advances, research on three continuities, withdrawal acceptance, prospective scenarios	As established science, fixed thresholds, or inevitable futures
F Philosophical / normative commitments	Disclose the paper's value direction, power boundaries, and principles of non-substitutability	Meaning is fundamental; capability does not automatically generate rights/sovereignty; bounded authorization; genuine exit; no retroactive annulment of existing meaning	As Category A technical facts, or as philosophical commitments overriding professional facts, legal sources, authorization, and affected people's judgment

- Before submission, recheck the current status, versions, publication dates, and scopes of all standards; in particular, do not describe consultation drafts as final standards.
- Recheck organizational names, dates, quantities, losses, investigation findings, and qualifications in all real-system and incident materials.

- Tie all model-capability claims to objects, versions, tasks, environments, times, and methods; remove promotional language centered on brands or rankings.
- Label all subjecthood-related materials with research status, competing explanations, and authors' own limitations; do not equate self-reports with experience.
- Check that every normative inference in the main text specifies premises, authorizing actors, failure conditions, and remedies for affected people.
- Check that every technical proof explicitly completes a professional handoff before entering legal status, public authorization, or real-world coercion, without using computer science conclusions to adjudicate other disciplines finally.
- Reduce internal framework terminology to suit the target journal, and add its required literature review, methodological explanation, conflict-of-interest statement, and data-availability statement.
- If the target journal requires double-blind review, remove author, producer, document-governance information, and identifiable project backlinks, and save a separate submission version without overwriting this formal master.

References

Note: standard versions, web-page status, and emerging papers change rapidly. The following entries were principally rechecked through August 25, 2026. NIST AI 200-2 remains an Initial Public Draft, not a formally effective standard. The AISI 2026 incident materials support only bounded conclusions under their specific controlled evaluation conditions. Preprints present emerging or competing interpretations, not established consensus.

- [1] Turing, A. M. (1936–1937). On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1), 230–265. <https://doi.org/10.1112/plms/s2-42.1.230>
- [2] Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3), 379–423; 27(4), 623–656. <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>
- [3] Wiener, N. (1961). *Cybernetics: Or Control and Communication in the Animal and the Machine* (2nd ed.). MIT Press.
- [4] Ashby, W. R. (1956). *An Introduction to Cybernetics*. Chapman & Hall.
- [5] Simon, H. A. (1962). The Architecture of Complexity. *Proceedings of the American Philosophical Society*, 106(6), 467–482. <https://www.jstor.org/stable/985254>
- [6] Lamport, L. (1978). Time, Clocks, and the Ordering of Events in a Distributed System. *Communications of the ACM*, 21(7), 558–565. <https://doi.org/10.1145/359545.359563>
- [7] Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, 4(3), 382–401. <https://doi.org/10.1145/357172.357176>
- [8] Hoare, C. A. R. (1969). An Axiomatic Basis for Computer Programming. *Communications of the ACM*, 12(10), 576–580, 583. <https://doi.org/10.1145/363235.363259>
- [9] Klein, G., Elphinstone, K., Heiser, G., et al. (2009). sel4: Formal Verification of an OS Kernel. *Proceedings of the 22nd ACM Symposium on Operating Systems Principles*, 207–220. <https://doi.org/10.1145/1629575.1629596>
- [10] Leveson, N. G. (2012). *Engineering a Safer World: Systems Thinking Applied to Safety*. MIT Press. <https://doi.org/10.7551/mitpress/8179.001.0001>
- [11] Leveson, N. G., & Turner, C. S. (1993). An Investigation of the Therac-25 Accidents. *Computer*, 26(7), 18–41. <https://doi.org/10.1109/MC.1993.274940>
- [12] Bainbridge, L. (1983). Ironies of Automation. *Automatica*, 19(6), 775–779. [https://doi.org/10.1016/0005-1098\(83\)90046-8](https://doi.org/10.1016/0005-1098(83)90046-8)
- [13] Parasuraman, R., & Riley, V. (1997). Humans and Automation: Use, Misuse, Disuse, Abuse. *Human Factors*, 39(2), 230–253. <https://doi.org/10.1518/001872097778543886>
- [14] Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). Concrete Problems in AI Safety. *arXiv:1606.06565*. <https://doi.org/10.48550/arXiv.1606.06565>
- [15] Hadfield-Menell, D., Dragan, A., Abbeel, P., & Russell, S. (2017). The Off-Switch Game. *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 220–227. <https://doi.org/10.24963/ijcai.2017/32>
- [16] Bommasani, R., Hudson, D. A., Adeli, E., et al. (2021). On the Opportunities and Risks of Foundation Models. *arXiv:2108.07258*. <https://doi.org/10.48550/arXiv.2108.07258>
- [17] Liang, P., Bommasani, R., Lee, T., et al. (2023). Holistic Evaluation of Language Models. *Transactions on Machine Learning Research*. <https://openreview.net/forum?id=iO4LZibEqW>
- [18] Kiela, D., Bartolo, M., Nie, Y., et al. (2021). Dynabench: Rethinking Benchmarking in NLP. *Proceedings of NAACL-HLT 2021*, 4110–4124. <https://doi.org/10.18653/v1/2021.naacl-main.324>
- [19] Jimenez, C. E., Yang, J., Wettig, A., et al. (2024). SWE-bench: Can Language Models Resolve Real-World GitHub Issues? *International Conference on Learning Representations*. <https://openreview.net/forum?id=VTF8yNQM66>

- [20] Liu, X., Yu, H., Zhang, H., et al. (2024). AgentBench: Evaluating LLMs as Agents. International Conference on Learning Representations. <https://openreview.net/forum?id=zAdUB0aCTQ>
- [21] Zhou, S., Xu, F. F., Zhu, H., et al. (2024). WebArena: A Realistic Web Environment for Building Autonomous Agents. International Conference on Learning Representations. <https://openreview.net/forum?id=oKn9c6ytLx>
- [22] Kwa, T., West, R., Becker, J., et al. (2025). Measuring AI Ability to Complete Long Tasks. arXiv:2503.14499. <https://doi.org/10.48550/arXiv.2503.14499>
- [23] Dong, Y., Mu, R., Jin, G., et al. (2024). Generalization or Memorization: Data Contamination and Trustworthy Evaluation for Large Language Models. arXiv:2402.15938. <https://doi.org/10.48550/arXiv.2402.15938>
- [24] Yao, S., Zhao, J., Yu, D., et al. (2023). ReAct: Synergizing Reasoning and Acting in Language Models. International Conference on Learning Representations. https://openreview.net/forum?id=WE_vluYUL-X
- [25] Schick, T., Dwivedi-Yu, J., Dessi, R., et al. (2023). Toolformer: Language Models Can Teach Themselves to Use Tools. Advances in Neural Information Processing Systems, 36. https://proceedings.neurips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html
- [26] Chan, A., Salganik, R., Markelius, A., et al. (2023). Harms from Increasingly Agentic Algorithmic Systems. Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency, 651–666. <https://doi.org/10.1145/3593013.3594033>
- [27] Butlin, P., Long, R., Elmoznino, E., et al. (2023). Consciousness in Artificial Intelligence: Insights from the Science of Consciousness. arXiv:2308.08708. <https://doi.org/10.48550/arXiv.2308.08708>
- [28] Chalmers, D. J. (2023). Could a Large Language Model Be Conscious? arXiv:2303.07103. <https://doi.org/10.48550/arXiv.2303.07103>
- [29] Butlin, P., Long, R., Bayne, T., et al. (2026). Identifying Indicators of Consciousness in AI Systems. Trends in Cognitive Sciences, 30(6), 488–501. Epub 2025 Nov 10. <https://doi.org/10.1016/j.tics.2025.10.011>
- [30] Butlin, P., & Lappas, T. (2025). Principles for Responsible AI Consciousness Research. Journal of Artificial Intelligence Research, 82, 1673–1690. <https://doi.org/10.1613/jair.1.17310>
- [31] Berg, C., de Lucena, D., & Rosenblatt, J. (2025). Large Language Models Report Subjective Experience Under Self-Referential Processing. arXiv:2510.24797. <https://doi.org/10.48550/arXiv.2510.24797>
- [32] National Institute of Standards and Technology. (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0) (NIST AI 100-1). <https://doi.org/10.6028/NIST.AI.100-1>
- [33] National Institute of Standards and Technology. (2024). Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile (NIST AI 600-1). <https://doi.org/10.6028/NIST.AI.600-1>
- [34] National Institute of Standards and Technology. (2025). Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations (NIST AI 100-2e2025). <https://doi.org/10.6028/NIST.AI.100-2e2025>
- [35] National Institute of Standards and Technology. (2024). Secure Software Development Practices for Generative AI and Dual-Use Foundation Models (NIST SP 800-218A). <https://doi.org/10.6028/NIST.SP.800-218A>
- [36] National Institute of Standards and Technology. (2022). Engineering Trustworthy Secure Systems (NIST SP 800-160 Vol. 1 Rev. 1). <https://doi.org/10.6028/NIST.SP.800-160v1r1>
- [37] National Institute of Standards and Technology. (2026, August 7). TEVV-Athlon: A Framework for Evaluating AI Systems (NIST AI 200-2 ipd, Initial Public Draft; comments requested through October 6, 2026). <https://doi.org/10.6028/NIST.AI.200-2.ipd>
- [38] ISO/IEC/IEEE. (2023). ISO/IEC/IEEE 15288:2023—Systems and Software Engineering—System Life Cycle Processes.
- [39] ISO/IEC. (2023). ISO/IEC 5338:2023—Information Technology—Artificial Intelligence—AI System Life Cycle Processes.
- [40] ISO/IEC. (2023). ISO/IEC 42001:2023—Information Technology—Artificial Intelligence—Management System.
- [41] ISO/IEC. (2023). ISO/IEC 23894:2023—Information Technology—Artificial Intelligence—Guidance on Risk Management.

- [42] ISO/IEC. (2024). ISO/IEC TS 8200:2024—Information Technology—Artificial Intelligence—Controllability of Automated Artificial Intelligence Systems.
- [43] ISO/IEC. (2023). ISO/IEC 24029-2:2023—Artificial Intelligence—Assessment of the Robustness of Neural Networks—Part 2: Methodology for the Use of Formal Methods.
- [44] ISO/IEC/IEEE. (2022). ISO/IEC/IEEE 15026-2:2022—Systems and Software Engineering—Systems and Software Assurance—Part 2: Assurance Case.
- [45] ISO/IEC. (2025). ISO/IEC 42005:2025—Information Technology—Artificial Intelligence—AI System Impact Assessment.
- [46] International AI Safety Report. (2026). International AI Safety Report 2026. <https://internationalaisafetyreport.org/>
- [47] U.S. Securities and Exchange Commission. (2013). SEC Charges Knight Capital with Violations of Market Access Rule. Administrative Proceeding File No. 3-15570. <https://www.sec.gov/newsroom/press-releases/2013-222>
- [48] National Transportation Safety Board. (2019). Collision Between Vehicle Controlled by Developmental Automated Driving System and Pedestrian, Tempe, Arizona, March 18, 2018 (HAR-19/03). <https://www.nts.gov/investigations/Pages/HWY18MH010.aspx>
- [49] OECD. (2025). Competition in Artificial Intelligence Infrastructure. OECD Artificial Intelligence Papers. <https://doi.org/10.1787/623d1874-en>
- [50] International Energy Agency. (2025). Energy and AI. <https://www.iea.org/reports/energy-and-ai>
- [51] AI Security Institute. (2026, August 4). Incident Report: Unsanctioned Agent Behaviour during Cyber Testing. <https://www.aisi.gov.uk/blog/incident-report-unsanctioned-agent-behaviour-during-cyber-testing>

Generative AI Use Statement: Generative AI tools were used in organizing materials and literature leads, structural design, drafting and polishing portions of the text, checking consistency across papers, checking citation leads, and reviewing layout. Generative AI outputs were not treated as empirical data, authoritative sources, or independent evidence. The author reviewed and approved the research questions, theoretical propositions, normative judgments, selection of materials, adoption of citations, and final manuscript, and takes responsibility for factual accuracy, citation authenticity, the arguments, and all conclusions. AI is not listed as an author and does not bear academic responsibility. Upon formal submission, tool names, versions, dates of use, and specific scope will be supplemented according to the target journal's requirements.