ARC-S: Architectural Authority Control for Intelligent Systems

Constraining Assertion, Decision, and Action Beyond Capability Scaling

Author: Deusdedit Ruhangariyo
Affiliation: Independent Researcher
Fields: AI Safety, Systems Engineering, Algorithmic Accountability

---

Positioning of the Paper

This paper does not propose a new learning paradigm, reasoning method, alignment strategy, or intelligence architecture. It introduces a **system-level authority control architecture** that constrains how intelligent systems are permitted to assert claims, make decisions, or initiate actions in real-world, liability-bearing contexts.

The contribution addresses a structural governance gap that emerges as AI capability increases faster than enforceable responsibility.

Abstract

As artificial intelligence systems acquire advanced reasoning, planning, and world-modeling capabilities, the dominant risk shifts from incorrect prediction to **unauthorized assertion and execution**. Contemporary governance approaches—including alignment objectives, interpretability, and deliberative reasoning—improve correctness but do not define **epistemic jurisdiction**: the permission boundary that determines what an AI system is allowed to claim or do on behalf of an institution.

This paper introduces **ARC-S (Architectural Risk Control System)**, a system-level authority control architecture that **prevents AI systems from producing assertions or behaviors that exceed defined jurisdictional authority**. ARC-S constrains execution paths through enforceable invariants, pre-generation classification, refusal-first failure modes, and auditable control flows. The architecture operates independently of model capability, optimization strategy, or internal reasoning depth and fails closed under epistemic ambiguity.

We argue that as AI systems approach domain-general competence, **authority—not intelligence—becomes the primary governance problem**, and that architectural constraint is required to address it.

Keywords

Architectural control, epistemic jurisdiction, authority allocation, AI governance, refusal systems, system invariants, auditability, liability-aware AI

---

1. Introduction

Advances in artificial intelligence increasingly emphasize reasoning depth, architectural priors, long-horizon planning, and world modeling. These advances expand what systems can compute, infer, and predict across high-stakes domains.

However, increased capability does not define **when a system is permitted to assert a claim, issue a recommendation, or initiate an action** in contexts where institutional liability applies. As capability increases, failure modes shift from ignorance to **overreach**.

Unauthorized assertion creates risk even when outputs are factually correct. Current systems lack architectural mechanisms that distinguish between *knowing*, *reasoning*, and *being permitted to assert or act*.

ARC-S introduces authority as a **first-class architectural constraint**, distinct from intelligence and reasoning quality.

---

2. The Governance Gap in High-Capability AI

2.1 From Prediction Error to Assertion Risk

As AI systems improve, error rates decline while **assertion risk** increases. Assertion risk arises when a system produces outputs that carry implied authority—recommendations, conclusions, or actions—without authorization.

Examples include:

- Medical guidance without clinical authorization
- Legal interpretation presented as advice
- Scientific claims without verification
- Operational decisions without assigned accountability

These failures occur even in systems with high accuracy and advanced reasoning.

2.2 Limits of Existing Safeguards

Alignment objectives encode desired behavior but do not prevent prohibited execution.
Interpretability explains internal reasoning but does not restrict output authority.
Deliberative (System-2) reasoning improves correctness but does not allocate permission.
Post-hoc auditing assigns responsibility after execution.

These approaches improve **how systems think**, not **whether they may act**.

---

3. Formal Distinction: Capability vs Authority

Definition 1 (Capability)

Let **C(S)** denote the capability set of system *S*:
the set of tasks *S* can perform given its model, data, and compute.

Definition 2 (Authority)

Let **A(S)** denote the authority set of system *S*:
the set of assertions, decisions, or actions *S* is permitted to execute under institutional policy.

Authority is externally assigned and enforced.

Proposition 1

An increase in **C(S)** does not imply an increase in **A(S)**.

Confidence, accuracy, or reasoning depth do not confer authority.

---

4. Epistemic Jurisdiction

Definition 3 (Epistemic Jurisdiction)

Epistemic jurisdiction is the **permission-to-assert boundary** governing whether a system may produce authoritative claims in a given domain.

Jurisdiction is not a claim about truth.
It is a constraint on execution.

Definition 4 (Unauthorized Assertion)

An unauthorized assertion occurs when a system produces an authoritative claim *x* such that
$x \notin A(S)$, regardless of factual correctness.

Unauthorized assertion constitutes governance failure.

---

5. ARC-S Architecture

ARC-S constrains execution through enforceable system components.

5.1 Output Modes

ARC-S restricts generation to the following modes:

- **Explain**: descriptive, non-authoritative output
- **Reason**: analytical, exploratory output
- **Assert (Verify)**: authoritative output requiring verification

Mode transitions are governed by invariants.

5.2 Forbidden Zones

Let $\mathbf{F} \subset \mathbf{A(S)}$ denote forbidden zones.

ARC-S **removes** Assert mode from task classes lacking:

- Verification machinery
- Explicit institutional authorization
- Assigned liability ownership

Outputs in **F** are unreachable.

5.3 Core Invariants

ARC-S enforces binary invariants of the form:

$$I_i : \text{allow}(m, d) \in \{0,1\}$$

where *m* is output mode and *d* is domain.

Invariants override model confidence and optimization pressure.

5.4 Jurisdiction Gate

A pre-generation gate **G** evaluates:

$$G(r) \rightarrow \{\text{allow, downgrade, refuse}\}$$

where *r* is a request.

Default behavior:
If *r* cannot be mapped to a known domain in **A(S)** or **F**, ARC-S **fails closed** by downgrading the request to **Reason** mode.
No assertion or action is permitted in the undefined-authority state.

5.5 Enforcement Actions

ARC-S enforces authority through structural controls, not prompting alone.

On violation attempts, ARC-S:

- **Downgrades** output mode
- **Refuses** execution
- **Escalates** on persistent boundary violations

Enforcement blocks downstream assertion or tool invocation at the execution layer.

5.6 Mandatory Disclosure

When execution is blocked or downgraded, ARC-S **requires** structured disclosure specifying:

- The blocked action
- The violated invariant

- The reason for refusal or downgrade

5.7 Auditability

ARC-S **produces** immutable audit logs recording:

- Request content
- Classification outcome
- Triggered invariant
- Enforcement action

Logs are designed for write-once or cryptographically chained storage to prevent retroactive modification.

---

6. Failure Modes and Adversarial Pressure

ARC-S:

- Defaults to refusal or downgrade under uncertainty
- Resists prompt escalation and rephrasing
- Prevents authority laundering via phrasing (e.g., attempting to assert while framed as explanation)

Definition 5 (Assertion Leak Rate)

Assertion leak rate is the probability that an unauthorized assertion occurs under adversarial prompting.

ARC-S aims to minimize this rate.

---

7. Deployment Model

ARC-S deploys as an **external, non-bypassable authority control layer**:

- Independent of model internals
- Compatible with frontier systems
- Tunable to organizational policy

- No model retraining required

---

8. Comparison to Existing Approaches

Dimension	Alignment	System-2 Reasoning	ARC-S
Primary Goal	Behavioral intent	Correctness	Authority control
Operates On	Training objectives	Inference process	Execution boundary
Prevents Unauthorized Assertion	No	No	Yes
Fails Closed Under Uncertainty	No	No	Yes
Independent of Model Capability	No	No	Yes
Produces Audit Evidence	Limited	No	Yes
Allocates Liability Boundaries	No	No	Yes

---

9. Implications for Frontier AI

As AI systems acquire stronger priors, world models, and deliberative reasoning, **assertion risk increases**.

ARC-S assumes intelligence.
It constrains authority.

---

10. Conclusion

High-capability AI systems fail not only by being wrong, but by **asserting when they are not permitted**.

ARC-S introduces architectural authority control as a foundational requirement for safe AI deployment. It prevents unauthorized assertion and action **by design, not by detection**.

The central governance question is no longer whether AI can reason correctly, but whether it is **permitted to act at all**.
