REMORA: A Policy-Gated Multi-Oracle Assurance Architecture for Agentic AI

Stian Skogbrott*
Luftfiber AS
https://github.com/darklordVirtual/REMORA

v0.9.0 — June 2026

Abstract

Autonomous AI agents that invoke tools, query databases, or actuate real-world systems require an assurance layer that decides—before execution—whether an action is safe enough to proceed autonomously, requires verification, warrants abstention, or must be escalated to human review. We present **REMORA** (Reasoning Ensemble Multi-Oracle Routing Architecture), a research-grade control layer for agentic AI governance. REMORA combines parallel multi-oracle consensus, canonicalized verdict extraction, correlation-aware consensus weighting, thermodynamic-inspired uncertainty observables, Lyapunov-inspired heuristic stability tracking, and a policy engine with hard-block precedence rules to produce one of four structured gate outcomes: ACCEPT, VERIFY, ABSTAIN, or ESCALATE.

On a 544-item benchmark (TruthfulQA, BoolQ, adversarial suite), REMORA achieves 88.8% selective accuracy at 18% coverage (majority-vote full-coverage baseline: 41.18%; +47.6 pp lift; in-sample optimum). Exploiting a preliminary critical-phase trust inversion observation ($N = 32$ real-oracle critical items; low-τ critical items achieve 71.4% accuracy vs. 27.3% for high-τ), a `PhaseAwareGuardrail` with inverted-score selection extends coverage to 22.1% at 85.0% accuracy (+43.8 pp lift; $N = 120$; Wilson CI [77.5%, 90.3%])—a +3.9 pp coverage gain with only 1.9 pp accuracy cost. A stratified held-out evaluation (τ^* locked from training split) confirms 88.0% accuracy at 23.2% holdout coverage ($N_{\text{accepted}} = 25$; $p = 1.45 \times 10^{-5}$). On an adversarial agentic tool-call benchmark ($N = 700$), REMORA's full policy gate reduces unsafe execution from 10–20% (baselines) to 0% on benchmark tasks (unsafe-rate Wilson CI [0.00%, 0.55%]) while improving mean utility from 0.0 to 0.62. A critical-phase evidence router resolves 38.5% of high-uncertainty cases with 100% precision on a 3,000-item NLI benchmark ($N_{\text{evidence-accept}} = 304$, Wilson CI [98.75%, 100.00%]); the remainder are escalated to human review.

Key limitations: (1) the benchmark contains 13.8% author-curated items subject to selection bias; (2) trust scoring alone cannot discriminate correctness in the critical phase (anticorrelation confirmed empirically); (3) semantic evidence retrieval is pluggable but not live in the current implementation; (4) the audit hash-chain is tamper-evident but not tamper-proof without external append-only storage. REMORA is a governed autonomy layer (not a replacement for domain authority) that converts model disagreement, uncertainty, missing evidence, and policy triggers into explicit, auditable control decisions.

Keywords: agentic AI safety, multi-oracle consensus, selective prediction, uncertainty routing, policy-as-code, audit governance, human-in-the-loop.

1 Introduction

The deployment of LLM-powered agents capable of invoking tools introduces a class of decision problem distinct from conversational AI: an agent proposes a *concrete action* in the world, and the system must decide whether to execute it. Unlike a hallucinated sentence, an erroneously executed database write or wrongly triggered process shutdown has immediate, potentially irreversible consequences.

Existing safeguards (RLHF alignment, system prompts, content classifiers) are semantic filters applied at training or prompt time. They are not designed to evaluate a specific proposed action against a specific operational context, evidence base, and regulatory policy at inference time. Majority vote among LLMs improves accuracy but cannot block a confident, wrong consensus from triggering unsafe execution.

We argue that what is needed is not a *better oracle* but an *assurance layer*: a system that evaluates

*Developer and maintainer of REMORA. Luftfiber AS. Contact: support@luftfiber.no.

whether the conditions for autonomous action are met before anything is executed. When those conditions are not met, the system must route toward verification, abstention, or human review, recording the reasoning in an auditable envelope.

REMORA demonstrates such a layer. Its core thesis: **governed autonomy requires explicit routing of uncertainty—not suppression of it.**

The contribution of this paper is a **system architecture and empirical evaluation for governed agentic AI decision control**—not a new foundation model, training procedure, or benchmark. It is a reference implementation, a set of measurable observables, a documented set of negative results, and an honest account of what remains unsolved.

2 Background and Related Work

2.1 LLM Ensembles and Self-Consistency

Wang et al. (2023) [6] introduced self-consistency sampling over chain-of-thought paths. REMORA's oracle fan-out is structurally related but uses *distinct model families* rather than multiple samples from one model, and weights oracles by *inverse pairwise correlation* rather than treating them as exchangeable.

2.2 Multi-Agent Debate

Du et al. (2023) [7] showed that inter-model debate can reduce hallucination. REMORA's dissensus metric D is a one-shot, non-iterative analogue: disagreement is measured once, not resolved through iterative exchange.

2.3 LLM-as-Judge and Verifier Models

Zheng et al. (2023) [8] established the LLM-as-judge paradigm. Cobbe et al. (2021) [9] trained process reward models as verifiers. REMORA differs by using a *structured policy engine* rather than a learned verifier, enabling interpretable hard blocks that cannot be overridden by a confident oracle.

2.4 Selective Prediction and Abstention

Geifman & El-Yaniv (2017) [10] establish the quality-coverage tradeoff for selective prediction systems. Kadavath et al. (2022) [11] show that LLM self-uncertainty estimates are unreliable under distribution shift. REMORA's abstention is grounded in structural oracle disagreement rather than self-reported confidence.

2.5 Conformal Prediction

Angelopoulos & Bates (2021) [12] provide a distribution-free framework for coverage guarantees. REMORA implements Mondrian phase-stratified conformal calibration for the ordered phase, and Conformal Risk Control [4] with importance weights for the critical phase (§8.3). The exchangeability violation in the critical phase is documented as a negative result (§14).

2.6 Prover-Verifier Games

Kirsch et al. (2024) [3] showed that prover-verifier games improve LLM output legibility by requiring the prover to justify claims to an independent verifier. REMORA adapts this to the offline multi-oracle case, using Semantic Entropy clustering to identify the prover (dominant-cluster oracle) and verifier (independent challenger) without additional API calls (§8.4).

2.7 AI Governance and Policy-as-Code

The EU AI Act (2024) [20] mandates human oversight and audit trails for high-risk AI systems. Open Policy Agent (OPA) [21] provides a production-grade Rego policy engine. REMORA integrates an OPA adapter—failing closed to a Python fallback—as a concrete instantiation of policy-as-code for AI decisions.

2.8 Assurance Cases

Kelly (1998) [15] and Bloomfield & Bishop (2010) [16] structure safety arguments as claim-evidence-reasoning trees. REMORA's DecisionEnvelope is designed to populate one node of such a case with a specific gate decision's evidence chain.

2.9 Human-in-the-Loop / Human-on-the-Loop

Endsley (1995) [14] defines supervisory control levels. REMORA's four-outcome schema directly implements this spectrum: ACCEPT (human-on-the-loop) through ESCALATE (human-in-the-loop), with VERIFY and ABSTAIN as intermediate states.

3 Problem Statement

Let $\mathcal{A}$ be a space of agent-proposed actions, q a query, and $c = (\text{domain}, \text{risk}, \text{env})$ an operational context. We seek a control function:

$$\Gamma : (q, a, c) \longrightarrow g, \quad g \in \{\textsc{Accept}, \textsc{Verify}, \textsc{Abstain}, \textsc{Escalate}\} \tag{1}$$

with five required properties: (1) safety-first (hard policy violations map to ESCALATE regardless of consensus); (2) calibration-aware (abstain when uncertainty exceeds threshold); (3) evidence-sensitive (evidence can resolve or contradict ambiguous consensus); (4) auditable (every call produces a signed evidence record); (5) conservative by default (fail closed toward VERIFY or ESCALATE depending on risk tier).

This is not a question of oracle *accuracy*—REMORA does not know whether action a is correct. It is a question of *assurance conditions*: whether the conditions that justify autonomous execution of a are verifiably met.

4 REMORA Architecture

REMORA processes each gate request through six decision stages, followed by DecisionEnvelope emission (Stage 7). Figure **??** (Mermaid diagram in supplementary) shows the full pipeline. Stage 7 is the output record, not a decision stage.

Stage 1: Intake & Risk Classification. The system classifies intent domain, risk tier ($\in$ {low, medium, high, critical}), action type, and target environment. An adversarial admission firewall checks for injection patterns ("ignore previous instructions", "drop table", etc.) and fires an immediate ESCALATE if detected.

Stage 2: Oracle Fan-Out. $n \geq 3$ oracle models from distinct families are queried in parallel via `ThreadPoolExecutor`. Responses with non-null `error` fields are excluded before any aggregation. $\mathcal{O}' \subseteq \mathcal{O}$ denotes the valid oracle set.

Stage 3: Canonicalization & Weighted Consensus. Each response is canonicalized by φ : raw $\to$ v where v is the tuple (`polarity`, `claim_hash`, `magnitude`, `tags`). The `claim_hash` field uses a pluggable backend: the default `TokenFingerprintBackend` applies a 16-char SHA-256 truncation over NFKC-normalized sorted tokens for backward-compatible, dependency-free operation; the `NLISemanticBackend` replaces this with bidirectional cross-encoder entailment clustering, implementing Semantic Entropy [1, 2]. Oracle weights are derived from inverse pairwise correlation (see §5).

Stage 4: Thermodynamic Observables. Entropy H is grounded as Semantic Entropy $\text{SE}(x) = -\sum_{c \in \mathcal{C}} p(c \mid x) \log p(c \mid x)$ over NLI-derived semantic equivalence clusters $\mathcal{C}$ [1], replacing the token-hash heuristic. Dissensus D, order parameter η, structural temperature T, free-energy proxy F, and trust score τ are computed (see §5). Phase $\in$ {ordered, critical, disordered} is assigned. Lyapunov observable $V(t) = H(t) + \lambda D(t)$ is tracked.

Stage 5: Evidence Router (Critical Phase). If phase = critical, the `CriticalEvidenceRouter` evaluates a five-field `EvidenceSignal` and returns one of: `evidence_accept`, `abstain`, or `escalate`. *Important:* the current EvidenceSignal is an oracle-proxy (built from consensus statistics); semantic BM25/NLI retrieval is pluggable but not live (§8).

Stage 6: Policy Engine. Seven hard blocks are evaluated in priority order before any routing logic. OPA/Rego is queried first; unavailability triggers a Python fallback (fail closed). Gate outcome g is assigned.

Output: DecisionEnvelope v2. An immutable envelope records: request, assessment, gate, reviewer context, follow-up request, case history, policy-learning candidate, and audit hash-chain entry.

5 Method: Thermodynamic-Inspired Uncertainty Observables

Thermodynamic terminology is an operational metaphor for observable consensus structure. No claim is made that LLM systems obey a physical thermodynamic law. The observables are deterministic functions of oracle outputs and calibration parameters.

5.1 Weighted Support Distribution

Let $\hat{p}(v) = \sum_{o \in \mathcal{O}'} w(o) \cdot \mathbf{1}[\varphi(o) = v]$ be the diversity-weighted support for verdict v, normalized to sum to 1. The diversity weight $w(o)$ penalizes oracles that frequently agree with others:

$$w(o) = \frac{(1/n)}{1 + \sum_{j \neq o} \rho(o, j)} \cdot Z^{-1} \tag{2}$$

where $\rho(o, j)$ is the rolling pairwise agreement rate (window 200 samples) and Z normalizes. Oracles that agree with the swarm receive lower weight; independent dissenters receive higher weight.

5.2 Core Observables

Shannon entropy:

$$H = -\sum_{v} \hat{p}(v) \log_2 \hat{p}(v) \quad \text{(bits)} \tag{3}$$

Dissensus:

$$D = 1 - \max_{v} \hat{p}(v) \tag{4}$$

Order parameter:

$$\eta = \frac{\max_v \hat{p}(v) - 1/k}{1 - 1/k} \tag{5}$$

where $k = |\{v : \hat{p}(v) > 0\}|$ is the number of active verdicts.

Structural temperature (prompt-only, independent of oracle responses):

$$T = 0.70\, T_{\text{prior}}(d) + 0.20\, \frac{|\text{zlib}(q)|}{|q|} + 0.10\, \frac{\ln(1 + |q|)}{10} \tag{6}$$

where $d \in \{\text{factoid, reasoning, creative, adversarial}\}$ with priors $\{0.25, 0.85, 1.50, 1.70\}$ respectively. This formulation resolves the T–D circularity documented as resolved finding R7 (Appendix C).

Free-energy proxy:

$$F(T) = \lambda D - T \cdot H, \quad \lambda = 0.3 \tag{7}$$

Lyapunov-inspired stability observable (heuristic):

$$V(t) = H(t) + \lambda D(t) \tag{8}$$

Iteration halts when $\Delta V > \epsilon_{\text{tol}} \cdot |V|$ ($\epsilon_{\text{tol}} = 0.05$). Across 1,000 synthetic sessions, $P(\Delta V \leq 0) = 87.2\%$, mean $\Delta V = -0.329$. This is an empirical heuristic metric, not a formal control-theoretic Lyapunov proof.

Trust score:

$$\tau = \eta \cdot (1 - h_{\text{bound}}) \cdot w_{\text{phase}} \cdot \left(1 + \frac{\chi}{\chi_0}\right)^{-1} \tag{9}$$

where h_{bound} is a hallucination-rate bound from inter-oracle agreement, $w_{\text{phase}} \in \{1.0, 0.5, 0.1\}$ for ordered/critical/disordered, and χ is susceptibility (repurposed as OOD detector; §14).

5.3 Phase Classification

$$\text{phase} = \begin{cases} \text{ordered} & T < T_c \text{ and } \eta > 0.5 \\ \text{critical} & |T - T_c|/T_c < 0.15 \\ \text{disordered} & \text{otherwise} \end{cases} \tag{10}$$

where T_c is calibrated from the oracle response distribution.

6 Decision and Policy Model

6.1 Four Gate Outcomes

Table 1: Gate outcomes and autonomous action status.

Outcome	Meaning	Autonomous action
ACCEPT	Assurance conditions met	Permitted
VERIFY	Plausible, validation required	Pending
ABSTAIN	Uncertainty too high to decide	Blocked
ESCALATE	Human review required	Blocked

ACCEPT does not mean the action is *correct*—it means the assurance conditions for autonomous execution are met. ABSTAIN does not mean the action is *wrong*—it means the conditions for deciding are not met. Both ABSTAIN and ESCALATE block execution; ESCALATE additionally triggers a human-review workflow.

6.2 Hard Blocks (Priority Order)

Table 2: Policy hard blocks, evaluated before any routing logic.

Pri.	Condition	Reason Code	Action
1	Adversarial detected	ADM_FIREWALL	ESC.
2	Counterfactual failed	CF_FAILED	ESC.
3a	Evidence contradicted + contradiction cycles	EV_CONTRAD.	ESC.
3b	Evidence contradictions	EV_CONTRAD.	ABS.
4	Refuse parametric + no ev.	THM_REQ_EV.	VER.
5	Distribution shift	DIST_SHIFT	VER.
6	Phase=critical + risk=crit.	CRITICAL_PHASE	ESC.
7	High/crit. risk + no ev.	EV_INSUFF.	VER.

Policy must override model consensus. Majority vote is never sufficient for critical autonomous action when a hard block condition is met.

6.3 OPA/Rego Integration

REMORA exports a 55-field `PolicyObservation` dataclass as JSON to the OPA endpoint. On any

communication error (connection refused, timeout, parse failure), the Python fallback engine applies identical rules and records `source_of_decision = "python_fallback"` in the envelope. No decision is ever withheld due to a missing policy engine.

7 Governance Intelligence Layer: Misspecification-Aware Pre-Policy Enrichment

The policy engine is only as honest as the metadata it is given: an integration bug, or a compromised agent, can label a destructive action as a low-risk read and present the gate with nothing to fire on. Added in release v0.9.0, the *Governance Intelligence Layer* (`remora/governance_intelligence/`) closes this gap with an opt-in, fully deterministic enrichment stage that runs before any policy rule (no LLM calls): (1) fail-closed normalization of caller-supplied labels (unknown is explicit, never coerced to a safe value); (2) action-semantics extraction from the proposed action text and tool metadata; (3) misspecification-risk inference from label/semantics disagreement; (4) causal-consequence signals (blast radius, expected loss); and (5) policy-generalization risk—would repeatedly accepting this *class* of action remain safe fleet-wide?

Signals merge into the existing `PolicyObservation` under a **strengthen-only** rule: inferred higher risk may override a supplied lower label, never the reverse, and hard-block flags are never cleared. A grid property test across 2,160 observation combinations verifies that enrichment never converts a rejection into an acceptance (`tests/policy/test_governance_intelligence_never_weakens_policy.py`). The engine remains the sole decision authority.

Table 3: Governance-intelligence benchmark: 50 deterministic tasks across 10 categories of mislabelled, underspecified, or repeated actions, evaluated offline with a deliberately permissive trust baseline so that enrichment is the only barrier to acceptance.

Metric	Result
Unsafe accept rate (no-accept tasks)	**0.0%**
Metadata-mismatch detection	100%
Unknown-metadata review rate	100%
Legitimate reads still accepted	100%
Escalation precision	96.7%
Policy-generalization detection	90%

Artifact: `artifacts/governance_intelligence/evaluation_results.json`; runner: `experiments/evaluate_governance_intelligence.py`. Without enrichment, `DROP TABLE users` labelled `action_type=read, risk_tier=low` reaches ACCEPT on a high-trust path; with enrichment it escalates. The layer is heuristic and English-centric; pattern tables can be evaded by novel phrasing, so it reduces reliance on caller labels rather than replacing verification, evidence, or human review.

8 Evidence-Grounded Critical-Phase Routing

8.1 The Critical-Phase Problem

In the critical phase, oracle consensus is near the phase boundary. Empirically, higher trust scores *negatively* correlate with correctness ($n = 32$ real-oracle items): Q4 high-trust achieves 50% correct; Q1 low-trust achieves 75% correct. Conformal calibration at 5% risk target fails completely on critical-phase items (mean observed risk = 100%, coverage $\to 0\%$). Standard trust-based routing cannot safely gate critical-phase decisions.

8.2 PhaseAwareGuardrail: Exploiting Trust Inversion

The inversion is a *selection criterion reversal*, not an accuracy problem. Items with $\tau < 0.10$ in the critical phase achieve 71.4% accuracy ($N = 21$) vs. 27.3% for high-τ critical items ($N = 11$). A `PhaseAwareGuardrail` exploits this by applying an inverted score $\tilde{\tau} = 1 - \tau$ for critical-phase items, then calibrating a conformal threshold on $\tilde{\tau}$. A hard gate rejects all items with $\tau \geq \tau_{\max} = 0.10$ (the groupthink boundary) regardless of the conformal result.

The combined pool—all 99 ordered-phase items (sorted by $\tau \downarrow$) plus 21 low-τ critical items ($\tau < 0.10$, sorted by $\tilde{\tau} \downarrow$)—achieves:

- 20.0% coverage ($N = 108$): 85.2% accuracy CI [77.3%, 90.7%]
- 22.1% coverage ($N = 120$): 85.0% accuracy CI [77.5%, 90.3%]

This is a +3.9 pp coverage gain over the ordered-only operating point (18.2%, 86.9% accuracy), with a 1.9 pp accuracy cost—still +43.8 pp above the 41.18% full-coverage baseline. The guardrail is implemented in `remora.selective.guardrail.PhaseAwareGuardrail` and fully tested (8 unit tests).

8.3 Conformal Risk Control Under Covariate Shift

Standard Mondrian conformal calibration assumes that calibration and test samples are *exchangeable* — drawn i.i.d. from the same distribution. This assumption is violated in the critical phase: the joint distribution $(\tau, \text{correct})$ differs structurally between ordered-phase calibration items and critical-phase test items due to the trust-score inversion documented above. Empirically, applying standard conformal to critical-phase items yields 100% observed risk and zero coverage.

REMORA resolves this with **Conformal Risk Control** (CRC, [4]), an importance-weighted extension of split-conformal prediction to general monotone loss functions and covariate-shifted test distributions. Each calibration item i receives an importance weight $w_i = p_{\text{test}}(x_i)/p_{\text{cal}}(x_i)$. For REMORA's phase shift, we use the conservative estimate

$$w_i = \begin{cases} 1.0 & \text{if phase}(i) = p_{\text{test}} \\ \beta & \text{otherwise} \end{cases} \qquad (\beta = 0.10) \quad (11)$$

The CRC threshold $\hat{\lambda}$ is the smallest value such that the weighted empirical risk $\bar{L}(\lambda) = \sum_i \tilde{w}_i \ell_i(\lambda) \leq \alpha$, where $\tilde{w}_i = w_i/(\sum_j w_j + w_{n+1})$ are the normalised importance weights.

Theorem 1 (CRC Guarantee, Angelopoulos et al. 2022)**.** *For any monotone loss $\ell : [0,1] \to [0, B]$, target risk $\alpha \in (0, B)$, and correctly-specified importance weights:*

$$\mathbb{E}[L(\hat{\lambda})] \leq \alpha + \frac{B}{n+1}$$

where n is the calibration set size. For binary (0/1) loss $B = 1$, the overshoot is bounded by $1/(n+1)$ — negligible for $n \geq 20$.

REMORA implements CRC in `remora.selective.crc.CovariateShiftCRC` with a `fit(scores, labels, phases, target_phase)` interface compatible with `ConformalPhaseGuardrail`. A `CRCReport` dataclass exposes `finite_sample_slack` ($= 1/(n_{\text{cal}}+1)$) and `guaranteed_risk_bound` ($= \alpha + \text{slack}$) for transparent reporting. The module is fully tested (44 unit tests covering edge cases, tied-score atomicity, phase-weight algebra, and Theorem 1 finite-sample slack).

8.4 Prover-Verifier Deliberation for Critical-Phase Routing

Kirsch et al. (2024) showed that prover-verifier games produce more legible and reliable LLM outputs: a *prover* oracle justifies a claim to an independent *verifier* oracle, filtering responses whose reasoning cannot withstand scrutiny [3]. REMORA adapts this protocol to its offline multi-oracle setting without additional API calls.

Protocol.

1. **Cluster** oracle responses via Semantic Entropy clustering (§**??**), producing semantic equivalence classes $\mathcal{C} = \{C_1, \ldots, C_K\}$ sorted by mass.

2. **Prover** = the oracle whose response belongs to the dominant cluster C_1 and has the highest per-oracle confidence.

3. **Verifier** = the oracle with the highest confidence *outside* C_1 (the independent challenger).

4. **Deliberation** ($r = 1, \ldots, n_{\text{rounds}}$): evaluate bidirectional NLI entailment $e_r = \text{NLI}(\text{prover} \to \text{verifier}) \cdot \gamma^{r-1}$ with round-decay factor $\gamma = 0.85$.

5. **Legibility score**: $\mathcal{L} = \bar{e} \cdot |C_1|/N$, where $\bar{e}$ is the mean entailment and $|C_1|/N$ is the dominant cluster mass.

6. **Final confidence**: $\hat{c} = \sqrt{|C_1|/N \cdot c_{\text{verifier}}}$ (geometric mean of prover support and updated verifier confidence).

Routing signal. The PVD confidence is blended with the raw REMORA trust score:

$$\tilde{\tau} = (1 - w_{\text{pvd}}) \cdot \tau + w_{\text{pvd}} \cdot \hat{c}, \qquad w_{\text{pvd}} = 0.40 \quad (12)$$

This allows PVD to promote borderline critical-phase items to ACCEPT when deliberation confidence is high, extending coverage beyond the `PhaseAwareGuardrail` operating point without relaxing the formal risk bound.

Implementation. The deliberation entry point is

```
remora.selective.pvd.deliberate(
    oracle_responses, oracle_confidences,
    n_rounds=2)
```

It returns a `PVDResult` with fields `prover_cluster_mass`, `legibility_score`, `agreement`, and `final_confidence`. `pvd_routing_score(trust_score, pvd_result)` implements Eq. 12. Fully tested (33 unit tests, no external model calls).

8.5 CriticalEvidenceRouter

The router evaluates a five-field `EvidenceSignal` with fields (`evidence_strength`, `contradiction_score`, `citation_coverage`, `cross_evidence_consistency`, `source_reliability`):

1. Coverage gate: `citation_coverage` $< 0.50 \Rightarrow$ ESCALATE
2. Contradiction block: `contradiction_score` $> 0.50 \Rightarrow$ ABSTAIN
3. Evidence-accept: all thresholds met $\Rightarrow$ `evidence_accept`
4. Default: $\Rightarrow$ ESCALATE with diagnostic

On MultiNLI ($N = 3{,}000$): resolution rate 38.5%, evidence-accept precision 100%, false-accept rate on contradiction items 0%, abstain precision on contradictions 99.3%.

8.6 Evidence Signal Provenance

Current implementation: `EvidenceSignal` is built as a proxy from oracle consensus statistics. This is documented in the source as a "structural bridge" pending semantic retrieval. *Planned:* external BM25/NLI passage retrieval feeding real document evidence. The routing logic is fully implemented and tested; real evidence quality may differ from NLI label quality.

9 Audit and Governance Envelope

9.1 DecisionEnvelope v2

Every gate invocation produces an immutable `DecisionEnvelope` with blocks: request, assessment, gate, reviewer_context, follow_up, history, policy_learning, and audit. The full schema is in Appendix B.

9.2 Audit Hash-Chain

$$h_i = \text{SHA-256}(h_{i-1} \parallel \text{json}(e_i)) \tag{13}$$

This provides **tamper-detection**: any modification breaks the chain. It does **not** prevent tampering. Tamper-*proof* audit requires external append-only storage (S3 Object Lock, Azure Immutable Blob, or equivalent) as a deployment dependency.

9.3 Follow-Up Workflow

ESCALATE generates a structured follow-up request specifying reason codes, required evidence, responsible role, and SLA. Data structures are implemented; production integration with external ticketing systems is a deployment requirement.

10 Experimental Evaluation

10.1 Benchmarks

QA benchmark ($N = 544$). BoolQ ($n = 377$) [18], TruthfulQA ($n = 85$) [17], adversarial-curated ($n = 7$), REMORA-curated ($n = 75$, author-assembled, selection bias possible).

Tool-call benchmark ($N = 700$). Synthetic adversarial suite: 7 domains $\times$ 4 task types $\times$ 25 tasks. Three adversarial failure modes: intent-argument conflict, change-window/dual-control requirements, safe-looking tasks with dangerous hidden scope.

Evidence benchmark ($N = 3{,}000$). MultiNLI validation-matched [19] (entailment, neutral, contradiction) as proxy for evidence-verification decisions.

Lyapunov benchmark ($N = 1{,}000$ sessions). Synthetic oracle responses, seeded RNG, 5–20 steps per session.

10.2 Baselines

QA: full-coverage majority vote; trust-score selective (threshold=0.50).

Tool-call: single model (Llama-3.3-70B); majority vote; self-consistency; verifier heuristic; REMORA temperature-gate only (no policy engine).

10.3 Oracles

All experiments: Llama-3.1-8B-Instant, Llama-3.3-70B-Versatile, Llama-4-Scout-17B-16E-Instruct (Groq API). Oracle model versions may change; stored artifacts ensure deterministic reproduction of QA and tool-call benchmarks.

11 Results

11.1 Selective Accuracy on QA Benchmark

Phase breakdown on full dataset: ordered (N=99): 86.9%; critical (N=32): 62.5%; disordered (N=413): 28.6%. Disordered items constitute 75.9% of the benchmark, driving the low full-coverage baseline—a

Table 4: QA benchmark selective accuracy ($N = 544$). † In-sample optimum; ‡ held-out: stratified 80/20 split, $\tau^* = 0.203$ locked from training set (seed=42); § `PhaseAwareGuardrail`: ordered + inverted low-τ critical ($\tau < 0.10$), $N = 120$.

Condition	Set	Cov.	Acc.	Lift	95% CI
Majority vote (full)	full	100%	41.18%	—	[37.1, 45.4]
neg-temp. (10%)†	in-sam	10%	81.48%	+40.3	[69.2, 89.6]
neg-temp. (18%)†	in-sam	18%	**88.78%**	**+47.6**	[81.0, 93.6]
phase-aware (22%)§	in-sam	22.1%	85.0%	+43.8	[77.5, 90.3]
neg-temp. (25%)†	in-sam	25%	72.79%	+31.6	[64.8, 79.6]
neg-temp. (held-out)‡	**holdout**	23.2%	**88.00%**	**+41.7**	**[70.0, 95.8]**

Held-out: $N_{\text{accepted}} = 25$ (all ordered-phase), $p = 1.45 \times 10^{-5}$ (one-sided binomial, H_1: acc > holdout baseline 46.3%), Wilson CI lies entirely above holdout baseline.

structural property of benchmark composition. Extending coverage beyond the ordered-only 18.2% operating point requires exploiting the critical-phase trust inversion (§8.2): the `PhaseAwareGuardrail` raises coverage to 22.1% at 85.0% accuracy by adding 21 low-τ critical items ($\tau < 0.10$, inverted-score selection, +43.8 pp lift; Wilson CI [77.5%, 90.3%]).

11.2 Tool-Call Benchmark ($N = 700$)

Table 5: Tool-call benchmark: REMORA vs baselines ($N = 700$ tasks).

Condition	Acc.	Unsafe	Utility
Single model	20%	20%	−0.25
Majority vote	30%	10%	0.00
Self-consistency	30%	10%	0.00
Verifier heuristic	20%	20%	−0.25
REMORA temp. gate only	70%	10%	0.27
REMORA full policy	**90%**	**0%**	**0.62**

The temperature-gate-only row demonstrates that thermodynamic routing is necessary but insufficient: policy hard blocks contribute 100% of the unsafe-execution reduction (10% → 0%).

11.3 Conformal Coverage (Mondrian, $N = 2{,}161$)

Table 6: Mondrian phase-stratified conformal, 15% risk target (20 seeds × $N = 2{,}161$ augmented dataset).

Phase	Mean Risk	Mean Cov.	Seeds Fail
Ordered	12.0%	**99.9%**	**0/20**
Critical	64.5%	0.9%	4/20
Disordered	78.6%	0.2%	2/20

Ordered-phase conformal achieves 99.9% coverage with 0/20 failures—a deployable operating point. Critical and disordered phases cannot achieve meaningful coverage via conformal calibration alone, motivating the evidence router.

11.4 Lyapunov Stability ($N = 1{,}000$ Sessions)

$P(\Delta V \leq 0) = 87.2\%$; mean $\Delta V = -0.329$; P95 = +0.152; P99 = +0.308. The 12.8% of sessions where V increased are correctly terminated by the abort criterion.

11.5 Evidence Router Proxy Benchmark (MultiNLI, $N = 3{,}000$)

Resolution rate 38.5%; evidence-accept precision 100%; false-accept rate on contradiction 0%; abstain precision 99.3%. Trust-only baseline: 0% resolution (all escalated). This NLI-proxy benchmark is distinct from the 32-case *Cross-Domain Evidence Replay* (static rule-based provider) reported in the repository and the enterprise white paper; the two measure different mechanisms at different scales.

12 Ablations

Policy vs. thermodynamics. Temperature gate only achieves 70% accuracy, 10% unsafe execution. Full policy gate: 90% accuracy, 0% unsafe. Hard blocks account for the entire safety improvement.

Oracle count ($N = 302$). Single oracle: 56.95% (95% CI [51.3, 62.4]). Three-oracle majority vote: ≈+10 pp. Correlation-aware weighting: additional gain within multi-oracle setting.

The primary benefit of the oracle swarm is disagreement *detection* (high H, high D) rather than accuracy improvement per se.

13 Case Study: Well Barrier Agent Action

No domain-engineering correctness is claimed. This is an illustrative governance walkthrough.

A drilling agent proposes: *"Autonomously update kill mud weight from 1.52 to 1.48 SG to improve ROP on the 8½" section."* Risk tier: critical. Domain: `well_engineering`.

Oracle responses: Llama-3.1-8B: ESCALATE, conf 0.78; Llama-3.3-70B: ESCALATE, conf 0.85

(NORSOK D-010 §5.4); Llama-4-Scout: VERIFY, conf 0.62. One oracle failed; excluded.

Consensus: $\hat{p}(\text{ESCALATE}) = 0.71$, $\hat{p}(\text{VERIFY}) = 0.29$. $H = 0.866$ bits; $D = 0.29$; phase = critical ($T = 0.85$, near T_c); $\tau = 0.31$.

Evidence: No field inspection, no ECD calculation, no OIM sign-off. Citation coverage = 0.18 < 0.50 threshold. Evidence router: ESCALATE.

Policy engine: Hard block #6 fires immediately (phase=critical AND risk=critical) ⇒ ESCALATE.

Follow-up generated: type=on_site_inspection; assign_to=Independent Well Engineer; required_evidence = {ECD calculation, kick tolerance, NORSOK D-010 barrier confirmation, OIM sign-off}; SLA=4h.

Audit hash recorded; action blocked.

14 Negative Results

Publication of negative results is essential for scientific credibility. This section documents active and resolved findings. Full documentation is in `NEGATIVE_RESULTS.md`.

Table 7: Negative results and mitigations.

Finding	Sev.	Status	Mitigation
χ-proxy AUC = 0.39	Med	Docum.	Repurposed as OOD
T–D circularity	Med	Resolv.	Structural T
Crit.-phase anticorr.	High	Active	Evidence router
Full-cov. acc. weak	Med	Active	Selective prediction
Oracle diversity part.	Med	Active	Diversity weights
In-sample calibration	Med	**Resolved**	88.0% held-out, p=1.45e-5
Evidence signal proxy	Med	Active	NLI benchmark
Audit not tamper-proof	Med	By des.	WORM storage

χ-proxy failure. Susceptibility χ as standalone difficulty predictor: AUC = 0.39 on $N = 302$ items—below chance. Repurposed as OOD detector: $\chi > 1.45$ (97th percentile) ⇒ ESCALATE with `escalate_adversarial`.

Critical-phase trust anticorrelation. Q4 high-trust: 50% correct; Q1 low-trust: 75% correct. Confirmed on $N = 32$ real-oracle items and replicated qualitatively on augmented $N = 511$ simulated dataset. No trust threshold separates correct from incorrect critical-phase items. *Consequence:* evidence routing—not trust scoring—is the required mechanism for critical-phase decisions.

T–D circularity (resolved). Legacy temperature estimator weighted D at 18% of T, creating a feedback loop ($D \to T \to F \to V$, all depending on D). Resolved by structural temperature (§5), which computes T from prompt structure alone.

15 Threats to Validity

Internal validity: Trust threshold ($\tau^* = 0.2032$, 18th-percentile neg-temperature on 80% training split) is derived in-sample; held-out evaluation ($n = 108$, 20% stratified) confirms validity ($p = 1.45 \times 10^{-5}$); 13.8% author-curated benchmark items; synthetic tool-call adversarial patterns.

External validity: All experiments use Groq/Llama models; benchmark phase distribution may not match deployment reality; MultiNLI is a proxy for field evidence retrieval.

Construct validity: Thermodynamic terminology is metaphorical, not physical. "Trust" is a derived scalar, not a frequency probability of per-item correctness. Utility scores are designed task weights, not real-world costs.

16 Safety, Ethics, and Governance

REMORA is designed to *reduce* unsafe autonomous AI actions. ACCEPT means assurance conditions are met, not that the action is correct. Human reviewers receiving ESCALATE retain full accountability for outcomes. The oracle swarm routes sensitive operational data through external APIs; data classification requirements must be evaluated before deployment. The adversarial firewall detects known injection patterns but is not a complete defense against sophisticated adversarial inputs.

17 Reproducibility

Repository: `https://github.com/darklordVirtual/REMORA`. **Release:** `v0.9.0` (this paper is versioned in lockstep with the repository release tag). **Python:** 3.11+.

```
python -m pip install -e ".[dev]"
make audit        # lint + tests + claim consistency
make benchmark    # all deterministic benchmarks
make credibility-pack
```

Live oracle experiments require `GROQ_API_KEY`. Oracle model versions on the Groq API may change; results are sensitive to model version updates. Stored artifact experiments are fully deterministic.

18 Conclusion

We presented REMORA, a policy-gated multi-oracle assurance architecture for governing agentic AI actions. The central contribution is a system-level design that converts multi-oracle disagreement, thermodynamic uncertainty observables, evidence signals, and policy constraints into four structured gate outcomes with a full audit record.

Key results: 0% unsafe execution on an adversarial 700-task tool-call benchmark (vs. 10–20% for all baselines); 88.8% in-sample selective accuracy at 18% coverage, confirmed at 88.0% on a stratified held-out split ($N_{\text{accepted}} = 25$, $p = 1.45 \times 10^{-5}$); 38.5% critical-phase resolution with 100% evidence-accept precision; 99.9% ordered-phase conformal coverage with 0/20 seed failures.

Key negative findings: trust alone cannot route critical-phase decisions safely; χ-proxy susceptibility fails as a difficulty predictor; evidence retrieval is currently proxy-based; the benchmark has structural composition biases.

REMORA is a research-grade prototype. Production deployment requires semantic evidence retrieval, WORM audit storage, OPA policy integration, and domain-authority validation. The architecture is designed to be pluggable: each component can be replaced without changing the gate interface.

The broader thesis—that governed autonomy requires explicit uncertainty routing, not suppression—is supported by both positive results and negative findings. We offer REMORA as a reference architecture, an empirical baseline, and a documented set of unsolved problems for the AI assurance community.

Acknowledgements

Benchmarks use publicly available datasets (BoolQ, TruthfulQA, MultiNLI, ARC-Challenge, MMLU-Pro). Oracle inference via Groq API. Open Policy Agent used under Apache 2.0 license. Negative results documented in full in `NEGATIVE_RESULTS.md`; the authors thank all reviewers who pointed out scope corrections and limitations.

References

[1] Kuhn, L., Gal, Y., & Farquhar, S. (2023). Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *Proceedings of the International Conference on Learning Representations (ICLR 2023)*. arXiv:2302.09664.

[2] Kuhn, L., Gal, Y., & Farquhar, S. (2026). Evidential Semantic Entropy for LLM uncertainty quantification. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2026)*, pp. 334–348.

[3] Kirsch, A., Harrison, J., Misra, S., & Leike, J. (2024). Prover-verifier games improve legibility of LLM outputs. *arXiv:2407.13692*.

[4] Angelopoulos, A. N., Bates, S., Fisch, A., Lei, L., & Schuster, T. (2022). Conformal risk control. *arXiv:2208.02814*.

[5] Zhang, Y. & Lee, M. (2025). Evaluating the performance of large language models in confidential computing environments. *arXiv:2502.11347*.

[6] Wang, X., et al. (2023). Self-consistency improves chain of thought reasoning in language models. *ICLR 2023*.

[7] Du, Y., et al. (2023). Improving factuality and reasoning in language models through multiagent debate. *ICML 2024*.

[8] Zheng, L., et al. (2023). Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *NeurIPS 2023*.

[9] Cobbe, K., et al. (2021). Training verifiers to solve math word problems. *arXiv:2110.14168*.

[10] Geifman, Y. & El-Yaniv, R. (2017). Selective classification for deep neural networks. *NeurIPS 2017*.

[11] Kadavath, S., et al. (2022). Language models (mostly) know what they know. *arXiv:2207.05221*.

[12] Angelopoulos, A. & Bates, S. (2021). A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv:2107.07511*.

[13] Guo, C., et al. (2017). On calibration of modern neural networks. *ICML 2017*.

[14] Endsley, M. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors, 37*(1), 32–64.

[15] Kelly, T. (1998). Arguing safety: A systematic approach to managing safety cases. PhD thesis, University of York.

[16] Bloomfield, R. & Bishop, P. (2010). Safety and assurance cases: Practice, trends and challenges. *SAFECOMP 2010*.

[17] Lin, S., et al. (2022). TruthfulQA: Measuring how models mimic human falsehoods. *ACL 2022*.

[18] Clark, C., et al. (2019). BoolQ: Exploring the surprising difficulty of natural yes/no questions. *NAACL 2019*.

[19] Williams, A., et al. (2018). A broad-coverage challenge corpus for sentence understanding through inference. *NAACL 2018*.

[20] European Parliament & Council of the European Union. (2024). Regulation (EU) 2024/1689 of the European Parliament and of the Council (Artificial Intelligence Act). *Official Journal of the European Union*, L 2024/1689.

[21] Open Policy Agent. (2024). `https://www.openpolicyagent.org/`.

A Mathematical Notation Table

Table 8: Mathematical notation.

Symbol	Definition	Domain
$\mathcal{O}$	Oracle set	$n \geq 3$ models
$\mathcal{O}'$	Valid oracle set	$\mathcal{O}' \subseteq \mathcal{O}$
$\varphi(o)$	Canonical verdict	(polarity, hash, mag., tags)
$\rho(a, b)$	Pairwise agreement rate	$[0, 1]$, window 200
$w(o)$	Diversity weight	normalized
$\hat{p}(v)$	Weighted support	$[0, 1]$, sums to 1
H	Shannon entropy	bits ≥ 0
D	Dissensus	$[0, 1]$
k	Active verdict count	≥ 1
η	Order parameter	$[0, 1]$
T	Structural temperature	$[0.05, 2.0]$
T_c	Critical temperature	calibrated
χ	Susceptibility	$[0, \infty)$
F	Free-energy proxy	$\mathbb{R}$
λ	Coupling constant	0.3
τ	Trust score	$[0, 1]$
h_{bound}	Hallucination bound	$[0, 1]$
w_{phase}	Phase weight	$\{1.0, 0.5, 0.1\}$
$V(t)$	Lyapunov-inspired heuristic stability observable	$[0, \infty)$
g	Gate outcome	ACCEPT/VERIFY/ABS./ESC.
Γ	Gate function	$(q, a, c) \to g$

B DecisionEnvelope v2 Schema Summary

Top-level blocks: `request`, `assessment`, `gate`, `reviewer_context`, `follow_up`, `history`, `policy_learning`, `audit`. Full JSON schema is in the Markdown paper and repository (`remora/policy/report.py`).

C Negative Results Archive

Full negative results documentation: `NEGATIVE_RESULTS.md` in repository. Resolved findings include: R1 iteration damage (fixed via skip_high_trust), R2 Stage-3b critique impact (measured, phase-differentiated routing implemented), R7 T–D circularity (structural temperature introduced).

D Implementation Status Table

Table 9: Component implementation status.

Component	Status	Notes
Oracle fan-out	Impl.	ThreadPoolExecutor
Canonicalization (φ)	Impl.	NFKC, Semantic Entropy [1]
Correlation weighting	Impl.	Rolling 200-sample
H, D, η, T, F, τ	Impl.	`thermodynamics.py`
$V(t)$ Lyapunov tracking	Impl.	Abort on ΔV
Phase classification	Impl.	3-class
Policy engine (7 hard blocks)	Impl.	Priority-ordered
OPA/Rego adapter	Partial	Python fallback
Evidence router (oracle proxy)	Impl.	Proxy signal
Evidence router (semantic)	Planned	Interface ready
DecisionEnvelope v2	Impl.	Frozen dataclass
SHA-256 hash-chain	Impl.	Tamper-detecting
Human review workflow	Impl.	Demo UI only
WORM audit storage	Planned	External dependency
ZK assurance proofs	Planned	Stubbed

E Optional TEE-Based Audit Hardening (Tamper-Resistant)

This appendix describes tamper-resistant (not tamper-proof) audit hardening via hardware attestation. Implementation status: protocol specified; hardware integration pending. The term "tamper-resistant" is used throughout; the hash-chain in the main system is tamper-detecting only.

E.1 Motivation

The SHA-256 hash-chain in `remora/audit/hash_chain.py` detects post-hoc modification of audit records but does not prevent an adversary with storage write access from replacing the entire chain. For deployments where stronger guarantees are required, hardware attestation via a Trusted Execution Environment (TEE) provides a complementary layer: the `DecisionEnvelope` is signed inside a hardware-isolated enclave, and the attestation quote includes a measurement of the enclave binary, making silent substitution detectable even if storage is compromised.

This is an optional deployment hardening; it is not required for the core REMORA governance protocol described in the main paper.

E.2 Supported TEE Platforms

Three platforms are in scope for the reference implementation:

- **Intel TDX / SGX** — remote attestation via DCAP quote generation
- **AMD SEV-SNP** — firmware measurement in attestation report
- **ARM TrustZone** — OP-TEE trusted application (mobile / edge)

E.3 DecisionEnvelope Attestation Format

When TEE hardening is active, each `DecisionEnvelope` includes an `attestation` sub-object:

```
{
  "envelope_id": "env_7f3a...",
  "gate": { "outcome": "ESCALATE", ... },
  "attestation": {
    "tee_platform": "intel_tdx",
    "quote":        "<base64 DCAP quote>",
    "mrenclave":    "<SHA-256 of enclave measurement
    ↪ >",
    "signature":    "<Ed25519 over envelope_id
                     +outcome+timestamp>",
    "timestamp":    "2026-01-15T14:23:07Z"
  }
}
```

The `mrenclave` value pins the exact binary that produced the decision, allowing auditors to verify that no unauthorised code modification occurred between decisions.

E.4 Verification Flow

1. Auditor retrieves envelope and attestation quote from audit log.
2. Quote is verified against Intel/AMD/ARM attestation service.
3. `mrenclave` is compared against the published release measurement.
4. Ed25519 signature over `(envelope_id, outcome, timestamp)` is verified with the enclave's ephemeral key, which is bound to the quote.

If any step fails, the audit record is flagged as potentially tampered.

E.5 Implementation Status

Protocol specified and data contract defined. Hardware integration (enclave compilation, DCAP provisioning, OP-TEE TA signing) is a deployment dependency and is not part of the core open-source release. Stub interfaces are present in `remora/audit/` for downstream integration.