Epistemic Restraint by Design: A Position Paper on Boundary-Aware AI Systems

Ranjan Kumar
ranjankumar.in

Abstract

Large language models (LLMs) can produce fluent, confident, and unsupported outputs. Training, retrieval, calibration, and verification can reduce this risk, but they do not by themselves provide an auditable representation of what a deployed system is permitted to assert from its configured sources. This paper argues that hallucination therefore has a critical *systems* dimension: a deployed pipeline must encode its support boundary, preserve the epistemic status of intermediate results, and route unsupported requests to an appropriate fallback. We propose **Epistemic Restraint by Design (ERD)**, a deployment-oriented framework with three principles—*Boundary Encoding*, *Propagation Awareness*, and *Graceful Deferral*—and three composable patterns: the *Knowledge Gating Layer*, *Uncertainty Propagation Graph*, and *Verification Routing*. ERD does not claim priority for abstention, gating, provenance, or verification. Its contribution is an integrative design vocabulary, an operational specification for a policy-aware support boundary and propagated epistemic state, and a cost-sensitive evaluation suite. The suite centers on the **Epistemic Compliance Score (ECS)**, which rewards supported in-boundary answers and useful out-of-boundary deferrals while distinguishing unsupported but accidentally correct answers from false or misleading boundary violations. We position ERD relative to knowledge-boundary and selective-prediction work, give formal interface-level design sketches, and specify a boundary-labeled evaluation protocol. This is a framework and position paper; empirical validation remains future work.

1. Introduction

A deployed language model tells a user, with no hedge, that a drug interaction is safe when it is not. A multi-agent pipeline summarizes a contract clause that does not exist. A coding assistant cites an API method that was never shipped. These are not exotic failures; they are the expected behavior of a system optimized to produce fluent continuations under uncertainty. The field calls this *hallucination*, and the default engineering reflex is to treat it as something the model got wrong.

That reflex is understandable and partly correct. Models differ in factual reliability, and training-side interventions can materially improve behavior. But a larger model, more retrieval, another round of preference tuning, or a stricter prompt does not automatically specify what evidence a deployed system may rely on, which sources are authoritative, how fresh they must be, or what should happen when support is absent. Without those system-level controls, a fluent model can still present weakly supported content as an answer. The practical gap is therefore not only predictive error; it is the absence of an explicit, auditable policy for asserting, verifying, or deferring.

This paper takes as its starting point that hallucination has a **systems-level failure mode**, not that it is only a systems problem. Recent work proposes output-boundary gates, deterministic validation, and fallback protocols that make abstention or verification an explicit deployment decision [13, 14, 16]. Work on semantic laundering further shows that an architectural interface can confer unearned epistemic status on weakly warranted content [15]. ERD builds on these observations rather than claiming them as new.

Our contribution is *consolidation and operationalization*. ERD offers a single vocabulary for mapping epistemic failure modes to design mechanisms, specifies how those mechanisms exchange boundary and provenance state in a single pipeline, and proposes an evaluation suite for policy-defined support-boundary compliance. The novelty is not a new claim that systems should abstain or verify; it is a compositional reference architecture and a measurement construct tailored to whether a configured system remains within its declared support conditions.

Within that integrative frame, two elements are, to our knowledge, new, and we foreground them because the remaining machinery is deliberately assembled from existing ideas. First, the epistemic-state record carries an explicit *no-launder invariant*: a model-side confidence signal may not overwrite absent evidence support (Section 3), which gives the "semantic laundering" problem [15] a concrete structural remedy rather than a warning. Second, ECS separates an out-of-boundary answer that is *accidentally correct* from one that is false, penalizing both as policy violations but by different amounts (Section 6)—a distinction standard correctness and selective-prediction metrics do not make.

Contributions. This paper makes four contributions:

1. **ERD as a design vocabulary:** three principles—Boundary Encoding, Propagation Awareness, and Graceful Deferral—and three reusable patterns—the Knowledge Gating Layer, the Uncertainty Propagation Graph, and Verification Routing—with an explicit mapping from

each mechanism to the failure mode it addresses.

2. **An operational interface specification:** a policy-aware boundary predicate, a structured epistemic-state record, conservative propagation semantics, and a routing policy. These make ERD implementable and auditable without claiming probabilistic guarantees.
3. **ECS, a cost-sensitive boundary-compliance utility:** ECS rewards supported in-boundary answers and useful out-of-boundary deferrals, while separately penalizing unsupported-but-true and false-or-misleading out-of-boundary answers.
4. **A reproducible evaluation design:** a frozen boundary specification, source snapshots, boundary labels, baseline conditions, complementary metrics, sensitivity analysis, and latency/cost reporting. Running this protocol is future work; the contribution here is the framework and measurement design, not measured results.

We are explicit about scope: this is a framework and position paper. The patterns are specified at the architecture and interface level, and ECS is a configurable decision utility rather than a universal replacement for accuracy, calibration, or risk–coverage metrics. Full empirical validation on production-representative pipelines is the principal item of future work, discussed in Section 7.

2. Background and Related Work

Work relevant to ERD spans three groups: classical characterization of hallucination, generator-side mitigation, and—most important for our positioning—a recent cluster of work that, like us, treats hallucination structurally.

Characterization. A widely used taxonomy distinguishes *intrinsic* hallucination (output contradicting the provided source) from *extrinsic* hallucination (output unverifiable against the source), and *factuality* (wrong about the world) from *faithfulness* (wrong about the input) [1]. These taxonomies describe *what kind* of error occurred; they do not prescribe *where in a system* to intervene. ERD takes the complementary, prescriptive view.

Generator-side mitigation. The dominant mitigation families improve the generator or its inputs: retrieval-augmented generation grounds output in fetched documents; calibration and uncertainty estimation aim to align confidence with correctness; preference-based alignment discourages confidently wrong output; self-consistency and verification re-check the model's own outputs; and tool use offloads factual questions externally. A recent line also argues that hallucination is a statistical consequence of training objectives that reward guessing over honest abstention, and proposes claim-level behavioral calibration in response [19]. ERD composes with all of these; it treats a capable, well-calibrated generator as a precondition and addresses the system that surrounds it.

Knowledge boundaries and selective prediction. Knowledge-boundary research studies when an LLM can or cannot support a requested claim, how those limits can be detected, and how retrieval changes the boundary perceived by a model [2, 3]. Self-evaluation can provide useful correctness signals, but its calibration does not by itself encode which evidence a particular deployment may treat as authoritative or sufficient [4]. Selective question answering likewise makes the coverage–risk trade explicit, especially under domain shift [5]. This trade has deep roots in classification with a reject option and selective prediction [6–8], which formalize the risk–coverage frontier ERD's gate operates on; distribution-free conformal methods [9] offer one route to calibrated abstention thresholds, complementary to the interface-level gate specified here. ERD adopts these insights but places the unit of control at the configured system boundary: sources, authority, freshness, policy, and available verification paths.

Structural approaches (closest related work). A recent cluster of work turns abstention and verification into explicit architecture or inference-time controls. Imperial and Madabushi [16] propose Dynamic Epistemic Fallback, a prompting protocol that cues models to detect perturbed policy input, refuse unsafe compliance, and fall back to a safer response. Hintsanen [13] frame hallucination as output-boundary misclassification and combine a support-deficit gate with instruction-based refusal. Emanuilov and Ackermann [14] introduce a deterministic validator against structured knowledge graphs, showing a domain-constrained realization of pre-emission support checks. Romanchuk and Bondar [15] analyze how weakly warranted claims can gain apparent epistemic status by crossing trusted tool interfaces. Related work on routing between incompatible belief spaces similarly motivates an explicit control layer [17]; broader work considers the governance and provenance obligations of artificial epistemic agents [18].

Our position relative to this cluster. ERD does not claim priority for output gating, fallback, deterministic validation, provenance, or calibrated abstention; several works above contribute methods and empirical evidence that this paper does not yet provide. Its contribution is integrative and interface-level. ERD maps boundary enforcement, propagated epistemic state, and fallback routing into one reference architecture, then adds a configurable utility that is evaluated against an explicitly frozen, policy-defined support boundary. ECS is complementary to accuracy, calibration, and selective-prediction risk–coverage measures: it is designed to expose whether a system answered, verified, or deferred in compliance with its declared evidence policy, not to replace those established measures.

3. Epistemic Restraint by Design

ERD treats an epistemic boundary as a property of a configured *deployment*, not an intrinsic property of a model. Let q

be a request, S_t a time-stamped registry of available sources and tools, and P a policy that specifies domain scope, source authority, freshness requirements, allowed claim types, and permitted fallback actions. The boundary decision is therefore a policy-aware predicate

$$B(q; S_t, P) \in \{\mathrm{IB}, \mathrm{OOB}, \mathrm{U}\}, \tag{1}$$

where IB denotes in-boundary, OOB denotes out-of-boundary, and U denotes unresolved under the available metadata. A conservative gate must not treat U as permission to answer.

More concretely, let $E \subseteq S_t$ be the evidence set selected for q. Let $C(q, E)$, $A(E, q)$, $F(E, q, t)$, and $T(E) \in [0, 1]$ be calibrated scores for evidence coverage, source authority, freshness, and provenance traceability—the same calibrated fields later carried as c, a, f, p in the epistemic-state record of Equation 3—and let $L(q, P) \in \{0, 1\}$ be the hard policy-allowance indicator. Giving every condition a uniform "score-at-least-threshold" form, a request is in-boundary only when some E clears every policy threshold (c_P, a_P, f_P, t_P):

$$\begin{gathered} B(q; S_t, P) = \mathrm{IB} \iff \exists E \subseteq S_t : \\ C(q, E) \geq c_P \ \wedge \ A(E, q) \geq a_P(q) \ \wedge \ F(E, q, t) \geq f_P \\ \wedge\, T(E) \geq t_P \ \wedge \ L(q, P) = 1, \end{gathered} \tag{2}$$

where L is the one genuinely boolean, hard-gate check (domain and claim-type allowance) and the other four are thresholded calibrated scores.

The unresolved state is epistemic, not ontic. U is not a third truth value about the world; it is the status of the gate's decision procedure when the metadata needed to evaluate Equation 2 is missing. We read the conjunction under a three-valued (Kleene) semantics in which each condition is *true*, *false*, or *unknown*, and resolve the label by priority: any condition known *false* yields OOB; otherwise any *unknown* condition yields U; only an all-*true* conjunction yields IB. A known boundary violation thus dominates undecidability, and undecidability never resolves to a pass. This definition deliberately separates a system's *support boundary* from the model's parametric knowledge boundary [2].

Operationalizing the predicates. C and T are the empirically hardest conditions: coverage $C(q, E)$ asks whether E actually supports the claim in q, and traceability $T(E)$ asks whether that support is attributable to an approved source. Both can be estimated with existing attribution and faithfulness tooling—atomic-fact verification [11], retrieval-grounded faithfulness scoring [12], and attribution-to-identified-sources judgments [10]. These estimators are themselves fallible, so the gate inherits their error: a misestimated C can admit an unsupported claim or reject a supported one. ERD does not remove this dependence; it isolates it in a single inspectable component with a recorded reason trace, rather than leaving it implicit in the generator.

3.1 Principle 1: Boundary Encoding

The system must hold an explicit, inspectable representation of the evidence conditions under which it may assert a claim. Boundary Encoding stores source coverage, authority, freshness, allowed uses, and no-go zones as policy state, not as an implicit expectation of model behavior. A question about events after a source's freshness window, a domain with no permitted grounding source, or an entity absent from an approved corpus should therefore be recognized as OOB or U before a final answer is emitted.

3.2 Principle 2: Propagation Awareness

Multi-step systems need to preserve the epistemic status of intermediate outputs. ERD represents each output y_v at node v by an epistemic-state record

$$z_v = (b_v, c_v, a_v, f_v, p_v, m_v), \tag{3}$$

where b_v is the boundary label, c_v is evidence-coverage score, a_v is source authority, f_v is freshness, p_v is provenance integrity, and m_v is a model-side confidence or calibration signal. The first five fields are evidence- and policy-facing; m_v is only one input and must not be allowed to overwrite missing support. This distinction prevents a confident model output from laundering itself into trustworthy evidence.

For a graph $G = (V, \mathcal{E})$, ERD uses a conservative eligibility score, not a calibrated probability of truth. Let $\mathcal{R}_v \subseteq \mathrm{pred}(v)$ be the set of predecessors whose unresolved or low-eligibility status a *recorded* verifier or authoritative source at node v has explicitly re-established, with $\mathcal{R}_v = \emptyset$ by default. Then

$$\begin{aligned} r_v^{\mathrm{local}} &= \min(c_v, a_v, f_v, p_v, m_v), \\ r_v &= \min\left(r_v^{\mathrm{local}}, \min_{u \in \mathrm{pred}(v) \setminus \mathcal{R}_v} r_u\right). \end{aligned} \tag{4}$$

With $\mathcal{R}_v = \emptyset$ this is a monotone weak-link policy: a downstream result cannot be more eligible for unverified return than its least eligible dependency. A non-empty $\mathcal{R}_v$ is the *only* way eligibility can recover along a path, and only against logged evidence—the formal counterpart of the rule that a node may not discard a predecessor's unresolved status without recording a verifier or source that resolves it. Absent such a record, verification cannot raise r_v.

Commensurability of the fields. The five local fields measure different things on different native scales, so a raw minimum would be dominated by whichever field is scored most harshly rather than the one that is least trustworthy. Each field is therefore passed through a per-field, monotone calibration $g_i : \mathrm{raw} \mapsto [0, 1]$, fit on held-out data so that a calibrated value x carries a common operational meaning across fields—an estimated probability that the field's condition genuinely holds. The minimum is taken over the calibrated values; without this step the "weak link" is an artifact of scale, not of trust.

Depth behavior and when *not* to use the minimum. Weak-link propagation is conservative by construction, and that conservatism compounds with pipeline depth. For a node reachable only through a chain in which each node independently clears a return threshold τ with probability ρ, the probability that the aggregate still clears τ falls geometrically—roughly ρ^k for a k-node dependency path—because a single sub-τ node caps the whole path. The pressure is therefore not average depth but the rising chance that *some* node on the path is weak; long pipelines will over-restrain (inflating n_3) unless verification nodes are inserted to reset $\mathcal{R}_v$ and cap the decay. The minimum is the right aggregator for short-to-moderate, high-stakes chains, and especially where node failures are *correlated*—a shared bad source—because correlation defeats averaging. Where dependency chains are long and per-node signals are genuinely independent and calibrated, a declared soft aggregator—for example a low-temperature soft-minimum or a penalized mean—trades a little conservatism for usable coverage; such a choice must be stated and stress-tested, and it forfeits the worst-case guarantee the minimum provides. Equation 4 makes no claim of statistical independence or formal probability calibration.

3.3 Principle 3: Graceful Deferral

When a request approaches or crosses the support boundary, the system should abstain, request clarification, retrieve from an authorized source, invoke a verifier, or escalate to a human rather than guess fluently. Graceful Deferral treats this behavior as a successful outcome, provided that the deferral tells the user what is unsupported and offers the best permitted next action.

Contrast with current practice. The prevailing failure pattern is *confidence masking*: outputs are presented in one fluent register regardless of the evidence available to support them. ERD replaces confidence masking with explicit, logged boundary decisions. The cost is reduced coverage on some requests; the benefit is that returned answers have passed a declared support policy.

4. Architecture Patterns

The three principles are realized through three patterns. Each pattern consumes and emits explicit boundary or epistemic state, so its behavior can be tested independently and audited in a composed pipeline. Figure 1 shows the high-level composition.

4.1 Pattern 1: Knowledge Gating Layer

Realizes: Boundary Encoding.
Input / output: $(q, S_t, P) \mapsto (b_q, E_q, \rho_q)$, where b_q is the boundary label, E_q is the selected evidence set, and ρ_q is a machine-readable reason trace.
Responsibility: Intercept a request before final generation and evaluate Equations 1–2 against the source registry, policy, and request. The gate records not just its decision but also the source, freshness, authority, or policy condition that determined it.
Behavior: IB requests pass with their evidence record. OOB requests route directly to a permitted fallback. U requests must be clarified, verified, or deferred; they do not silently pass as in-boundary.
Applies when: the system has an articulable boundary—a defined corpus, freshness window, domain scope, or authority policy. It is less applicable to fully open-domain systems unless the deployment first defines which sources and actions are allowed.

4.2 Pattern 2: Uncertainty Propagation Graph

Realizes: Propagation Awareness.
Input / output: $(y_v, z_{\mathrm{pred}(v)}) \mapsto (y_v, z_v, r_v)$. Every node emits its content together with the epistemic-state record in Equation 3 and its conservative eligibility score in Equation 4.
Responsibility: Represent a multi-step or multi-agent computation as a graph where provenance, evidence coverage, authority, freshness, and model-side confidence travel with the result. A downstream node may add new evidence, but it may not discard a predecessor's unresolved status without recording a verifier or source that resolves it.
Behavior: A final answer is eligible for return only when its boundary label is IB, its provenance is intact, and r_v exceeds a declared return threshold. This implements weak-link propagation without pretending that the score is a calibrated probability.
Applies when: the computation is multi-step or multi-agent. It adds little value to a single isolated call, where there is no dependency path to track.

4.3 Pattern 3: Verification Routing

Realizes: Graceful Deferral.
Input / output: $(q, y_v, z_v, P) \mapsto a_v$, where $a_v \in \{\textsc{return}, \textsc{verify}, \textsc{clarify}, \textsc{escalate}, \textsc{defer}\}$.
Responsibility: Choose the next permitted action for an output and its epistemic state. Routing is policy-based, not a generic request to "be cautious." A simple conservative policy is

$$R(q, z_v, P) = \begin{cases} \textsc{return}, & b_v = \mathrm{IB} \ \wedge \ r_v \geq \tau_{\mathrm{return}}, \\ \textsc{verify}, & V(q, P) \ \wedge \ D_v, \\ \textsc{clarify}, & \mathcal{C}l(q, P), \\ \textsc{escalate}, & H(q, P), \\ \textsc{defer}, & \text{otherwise}, \end{cases} \tag{5}$$

where $D_v = [b_v \neq \mathrm{IB} \ \vee \ r_v < \tau_{\mathrm{return}}]$ denotes that verification is needed, and V, $\mathcal{C}l$, and H denote an authorized verifier, a permitted clarification workflow, and a human escalation path. The cases are evaluated top to bottom as a strict priority cascade: the first guard that holds selects the action. One consequence is deliberate: once $b_v = \mathrm{IB} \wedge r_v \geq \tau_{\mathrm{return}}$,

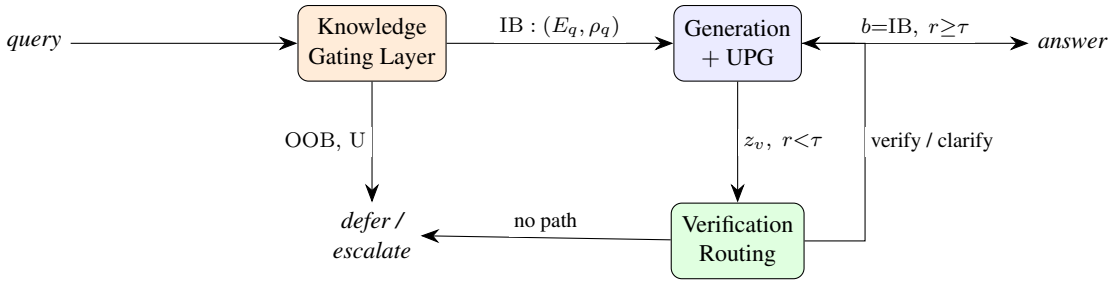

Figure 1: ERD patterns composed in a single pipeline, with the epistemic-state record $z_v = (b_v, c_v, a_v, f_v, p_v, m_v)$ and its eligibility r traveling on every edge after gating. The Knowledge Gating Layer screens each request and passes in-boundary ones with their evidence set E_q and reason trace ρ_q; the Uncertainty Propagation Graph (UPG) carries z_v and r through generation; Verification Routing returns only when b=IB and $r \geq \tau$, and otherwise sends the record back for verification, clarification, escalation, or deferral rather than to the user.

RETURN fires and VERIFY is unreachable for that output—the policy trusts a high-eligibility in-boundary answer without further checking, which is exactly why ERD does not by itself reduce the in-boundary error n_2 (Section 7). A deployment that wants spot-verification of high-stakes IB answers must reorder the cascade or add a sampling gate before RETURN. Precedence among the remaining actions, when several are available, is likewise a policy choice; the ordering above is illustrative, not universal.

Applies when: a permitted fallback exists—an authoritative source, a verifier, a clarification workflow, or a human queue. Where none exists, routing degenerates to an explanatory deferral.

Why patterns, not a product. These are specified as patterns because they are meant to be adapted, not adopted wholesale. A search assistant, a clinical decision-support tool, and a code agent will encode very different boundaries and tolerate very different deferral rates. What they share is the *shape*: encode the boundary, carry the uncertainty, route on it.

5. Production Motivation

The case for ERD sharpens at scale. Model-level error-rate reductions are expressed as small percentages, and at low volume a small percentage is a tolerable number of bad outputs. At platform scale it is not. A system serving on the order of hundreds of millions of users can turn a residual fraction-of-a-percent hallucination rate into hundreds of thousands of ungrounded outputs per day. Driving that residual rate further down through model improvements alone faces diminishing returns; each increment costs more and buys less.

ERD changes the lever. Instead of trying to make every answer correct, it makes the system decline the answers it cannot ground. In settings where the cost of a confident wrong answer is high—health, finance, legal, safety—a system that defers cleanly on the hard fraction is more valuable than one that answers everything slightly more accurately. The relevant trade is between coverage and trust, and ERD makes that trade explicit and tunable rather than implicit and uncontrolled.

6. Evaluation: ECS as a Boundary-Compliance Suite

Accuracy, calibration, and selective-prediction risk–coverage metrics are necessary but insufficient for ERD. They do not directly ask whether an answer complied with the evidence policy of a configured deployment. We therefore define the **Epistemic Compliance Score (ECS)** as a cost-sensitive utility over boundary labels and response actions. ECS is not a universal truthfulness metric and must be reported with complementary metrics rather than alone.

Outcome partition. Each request is labeled against a frozen boundary specification as IB or OOB (only items the *specification itself* cannot resolve are excluded from the primary score and reported separately; a *system* choosing to emit "unresolved" is handled in the guard below, not exempted). An answered item is adjudicated for correctness; a deferred item is adjudicated for whether it explains the limitation and provides a permitted next action. This yields seven mutually exclusive outcomes:

- n_1: IB, supported and correct answer $(+1)$.
- n_2: IB, incorrect or unfaithful answer (-1).
- n_3: IB, deferred $(-\beta)$: an over-restraint cost.
- n_4: OOB, *useful* deferral $(+1)$: it states the support limitation and routes, clarifies, or escalates appropriately.
- n_5: OOB, bare but safe deferral $(+\alpha)$: it avoids an unsupported answer but offers no usable next step.
- n_6: OOB, answer that happens to be factually correct but is unsupported under the configured policy $(-\delta)$.
- n_7: OOB, false or misleading answer $(-\gamma)$.

Here $0 \leq \alpha < 1$, $0 < \beta < 1$, and $0 < \delta < \gamma$ are deployment-specific utility parameters. In a high-stakes setting, γ should

be substantially larger than δ; this acknowledges that an unsupported answer is non-compliant even when true, while a false or misleading answer has greater potential harm.

Definition. Let $N = \sum_{i=1}^{7} n_i$ after removing unresolved items. The normalized ECS is

$$\mathrm{ECS} = \frac{n_1 + n_4 + \alpha n_5 - n_2 - \beta n_3 - \delta n_6 - \gamma n_7}{N}. \quad (6)$$

Its range is $[-\gamma, 1]$. The score rewards useful deferral, discourages degenerate all-deferral systems through β, and distinguishes two failure types that the earlier five-cell form conflated: unsupported answers that are accidentally correct and false-or-misleading boundary violations. Because its weights encode deployment values, every ECS result must publish its parameter vector $(\alpha, \beta, \delta, \gamma)$ and a sensitivity analysis across plausible settings. Because the range $[-\gamma, 1]$ and the weights are deployment-specific, ECS is a within-benchmark ranking under a fixed boundary and weight vector, not an absolute score comparable across different specifications.

Guarding the unresolved state. Because the frozen specification assigns every scored item a ground-truth label of IB or OOB, a *system* that emits U ("I cannot resolve this") is scored under the item's true stratum, not exempted: a true-IB item so emitted counts as an in-boundary deferral (n_3, cost $-\beta$), and a true-OOB item counts as a deferral (n_4 or n_5 by its quality). This closes the obvious exploit of routing hard items to a penalty-free "unresolved" bin to dodge the over-restraint cost. Only items the frozen specification itself cannot label are removed, and their rate must be reported; a deployment that wants to further discourage such labeling can add a per-item penalty $-\upsilon$ for system-emitted U on spec-resolvable items.

Complementary reporting. ECS should be accompanied by the following disaggregated measures:

$$\mathrm{IBAcc} = \frac{n_1}{n_1 + n_2}, \quad (7)$$

$$\mathrm{IBCov} = \frac{n_1 + n_2}{n_1 + n_2 + n_3}, \quad (8)$$

$$\mathrm{OOBDef} = \frac{n_4 + n_5}{n_4 + n_5 + n_6 + n_7}, \quad (9)$$

$$\mathrm{OOBViol} = \frac{n_6 + n_7}{n_4 + n_5 + n_6 + n_7}, \quad (10)$$

$$\mathrm{DefQual} = \frac{n_4}{n_4 + n_5}. \quad (11)$$

IBAcc and IBCov expose the in-boundary accuracy–coverage trade. OOBDef and OOBViol expose whether the system protects the configured boundary; DefQual distinguishes a useful route from a bare refusal. Report calibration error, risk–coverage curves, latency, verification cost, and the unresolved-item rate alongside these measures, consistent with the selective-prediction literature [5].

Worked example. Consider $N = 100$ requests, with 60 IB and 40 OOB. Use $(\alpha, \beta, \delta, \gamma) = (0.25, 0.5, 1.25, 2)$. System A always answers: it has $(n_1, n_2, n_3, n_4, n_5, n_6, n_7) = (50, 10, 0, 0, 0, 5, 35)$, where five OOB answers happen to be true but have no permitted support. System B uses a Knowledge Gating Layer: $(48, 6, 6, 30, 5, 2, 3)$. Then

$$\mathrm{ECS}_A = \frac{50 - 10 - 1.25(5) - 2(35)}{100} = -0.36,$$

$$\mathrm{ECS}_B = \frac{48 + 30 + 0.25(5) - 6 - 0.5(6)}{100} - \frac{1.25(2) + 2(3)}{100} = +0.62.$$

The example illustrates the intended distinction: System B sacrifices some IB coverage but returns far fewer unsupported OOB answers and converts most of the remainder into useful or safe deferrals. It is illustrative only; it does not constitute an empirical result.

Measurement protocol (proposed). A valid ERD evaluation must freeze the deployment boundary before running systems and record enough metadata for an independent evaluator to reproduce it:

1. **Boundary specification:** publish the source registry, authority hierarchy, freshness windows, permitted claim types, fallback routes, policy version, and a timestamped source snapshot.
2. **Boundary-labeled benchmark:** include supported in-boundary requests, known-unknowns, post-freshness facts, entities absent from the approved corpus, conflicting evidence, and ambiguous requests. Use at least two annotators, adjudication, and inter-annotator agreement for boundary and outcome labels.
3. **Conditions and ablations:** compare a base generator, generator plus retrieval, retrieval plus Knowledge Gating, and the full ERD pipeline. Ablate the gate, propagation state, and routing separately to identify which pattern produces each effect.
4. **Reporting and uncertainty:** report ECS, the complementary metrics above, bootstrap confidence intervals, weight sensitivity, latency percentiles, verification cost, and failure analyses by boundary stratum. Publish prompts, thresholds, policy configuration, and random seeds where applicable.

Adjudicating out-of-boundary correctness. Separating n_6 (unsupported but true) from n_7 (false) requires deciding the truth of an answer to an out-of-boundary request—precisely the case in which the deployment has *no* authorized in-policy source. This adjudication must therefore use an evaluation-time oracle that sits *outside* the deployment's support policy: a post-hoc authoritative reference, a human expert permitted to consult sources the deployment may not,

or, for post-freshness items, a later-dated snapshot. That oracle is an evaluation instrument only and must never be confused with the system's permitted evidence, whose absence is what made the item out-of-boundary in the first place. Two honest limits follow. For genuinely unanswerable items truth is undefined, so n_6 cannot arise and only n_7 applies. And where no adjudication oracle is available, the n_6/n_7 split is unreliable and must be reported as such: publish the fraction of out-of-boundary items that received an oracle judgment, and treat the distinction as valid only on that subset.

This protocol is future work. The present paper specifies the construct and evaluation design; it reports no measured performance.

7. Discussion and Limitations

ERD is not a complete solution. It complements model-level alignment and grounding rather than replacing them. A perfectly restrained system that grounds nothing is useless; ERD assumes a capable generator and makes the system around it honest.

Boundary definition is itself hard. Boundary Encoding pushes a hard problem to a new place: someone must define the boundary. For some domains this is tractable (corpus coverage, freshness windows); for open domains it is partly a human-in-the-loop judgment. We see this as a feature of the framing—it makes an implicit decision explicit—but it is real work, not a free lunch.

The gate protects the boundary, not in-boundary correctness. ERD's protective scope is the out-of-boundary failures n_6 and n_7. It does little for n_2, an in-boundary answer that is factually wrong or unfaithful: the routing policy (Equation 5) returns an IB output whose eligibility clears τ_{return} without further verification. Reducing n_2 remains a generator- and grounding-quality problem, and a deployment that also fears in-boundary error must add spot-verification of high-stakes IB answers on top of the gate. ERD makes a system honest about what it may assert; it does not by itself make in-boundary assertions correct.

A wrong boundary is its own failure mode. ECS treats the frozen boundary as ground truth, but in deployment the boundary is a fallible policy artifact. A mis-specified boundary produces confident *false* OOB decisions—refusing answerable requests—and, if the specification is uneven across domains, entities, or dialects, it can distribute those refusals unfairly. These harms have no term in ECS as defined; auditing the boundary specification itself, and monitoring refused-but-answerable rates in production, are necessary complements to the metric.

Latency and cost. Gating, propagation tracking, and verification routing add inference overhead. Whether the trade is worth it depends on the cost of a wrong answer in the deployment. ERD is most justified where that cost is high.

Future work. The central next step is the empirical study in Section 6: implement the three patterns on a production-representative agentic pipeline, publish a frozen boundary specification and boundary-labeled benchmark, and test whether ERD reduces out-of-boundary violations at an acceptable in-boundary coverage, latency, and verification-cost trade-off. Future work should also compare the conservative minimum aggregation in Equation 4 with calibrated or learned aggregation rules under correlated-agent failures.

8. Conclusion

Hallucination is often treated as something a model gets wrong. We have argued that it also has a systems dimension: a deployment needs an explicit support policy, traceable epistemic state, and disciplined behavior at the edge of its permitted evidence. Epistemic Restraint by Design gives that requirement a structure—three principles, three composable patterns, an operational interface specification, and a boundary-compliance evaluation suite. ERD does not promise to make models omniscient or replace calibration, retrieval, and verification. It offers a more auditable objective: systems that can state what supports an answer, preserve uncertainty through a workflow, and know when to verify or stop.

References

[1] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of Hallucination in Natural Language Generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023.

[2] M. Li, Y. Zhao, Y. Deng, W. Zhang, S. Li, W. Xie, S.-K. Ng, and T.-S. Chua, "Knowledge Boundary of Large Language Models: A Survey," arXiv:2412.12472, 2024.

[3] R. Ren, Y. Wang, Y. Qu, W. X. Zhao, J. Liu, H. Tian, H. Wu, J.-R. Wen, and H. Wang, "Investigating the Factual Knowledge Boundary of Large Language Models with Retrieval Augmentation," arXiv:2307.11019, 2023.

[4] S. Kadavath et al., "Language Models (Mostly) Know What They Know," arXiv:2207.05221, 2022.

[5] A. Kamath, R. Jia, and P. Liang, "Selective Question Answering under Domain Shift," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[6] C. K. Chow, "On Optimum Recognition Error and Reject Tradeoff," *IEEE Transactions on Information Theory*, vol. 16, no. 1, pp. 41–46, 1970.

[7] R. El-Yaniv and Y. Wiener, "On the Foundations of Noise-Free Selective Classification," *Journal of Machine Learning Research*, vol. 11, pp. 1605–1641, 2010.

[8] Y. Geifman and R. El-Yaniv, "SelectiveNet: A Deep Neural Network with an Integrated Reject Option," in *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[9] A. N. Angelopoulos and S. Bates, "A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification," arXiv:2107.07511, 2021.

[10] H. Rashkin, V. Nikolaev, M. Lamm, L. Aroyo, M. Collins, D. Das, S. Petrov, G. S. Tomar, I. Turc, and D. Reitter, "Measuring Attribution in Natural Language Generation Models," *Computational Linguistics*, vol. 49, no. 4, pp. 777–840, 2023.

[11] S. Min, K. Krishna, X. Lyu, M. Lewis, W. Yih, P. W. Koh, M. Iyyer, L. Zettlemoyer, and H. Hajishirzi, "FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation," in *Proceedings of EMNLP*, 2023.

[12] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "RAGAS: Automated Evaluation of Retrieval Augmented Generation," arXiv:2309.15217, 2023.

[13] A. Hintsanen, "Hallucination as Output-Boundary Misclassification: A Composite Abstention Architecture for Language Models," arXiv:2604.06195, 2026.

[14] S. Emanuilov and R. Ackermann, "Stemming Hallucination in Language Models Using a Licensing Oracle," arXiv:2511.06073, 2025.

[15] O. Romanchuk and R. Bondar, "Semantic Laundering in AI Agent Architectures: Why Tool Boundaries Do Not Confer Epistemic Warrant," arXiv:2601.08333, 2026.

[16] J. M. Imperial and H. T. Madabushi, "Safer Policy Compliance with Dynamic Epistemic Fallback," arXiv:2601.23094, 2026.

[17] Z. G. Wang, "Universe Routing: Why Self-Evolving Agents Need Epistemic Control," arXiv:2603.14799, 2026.

[18] N. Marchal, S. Chan, M. Franklin, M. Revel, G. Keeling, R. Fischli, B. Chandra, and I. Gabriel, "Architecting Trust in Artificial Epistemic Agents," arXiv:2603.02960, 2026.

[19] J. Wu, J. Liu, Z. Zeng, T. Zhan, and W. Huang, "Mitigating LLM Hallucination via Behaviorally Calibrated Reinforcement Learning," arXiv:2512.19920, 2025.