Compliance-Aware Retrieval-Augmented Generation for Regulated Financial-Reporting Corpora: A Real-World Evaluation on SEC EDGAR Filings

Aru Bhardwaj
Insightrix, France
aru.bhardwaj@insightrix.eu

Abstract

Retrieval-augmented generation (RAG) has become the dominant pattern for grounding large language models on enterprise corpora, yet production deployments in finance, healthcare, and any European Union jurisdiction subject to the Artificial Intelligence Act and the General Data Protection Regulation must satisfy a constraint that mainstream RAG ignores: a passage may be maximally relevant to a query and still legally inadmissible to the requesting user, purpose, or session. We formalise this as *constrained retrieval-augmented generation* and introduce CARAG, a five-stage architecture that treats compliance as a first-class property of the index, the retriever, the generator, and an append-only audit log. Crucially, we depart from prior work that evaluated such systems on synthetic corpora: we construct a benchmark from 6,000 real Securities and Exchange Commission filings (10-K, 10-Q, 20-F, 40-F, S-1, 8-K and amendments) drawn from the SEC Financial Statement Data Sets across seven recent quarters, yielding 26,595 chunked passages from 877 unique filers. **While the corpus itself is real, the access-control overlay (analyst roles, session purposes, deal-list information barriers) is necessarily simulated to reflect realistic enterprise policy environments** — no public regulatory archive ships with audit-grade user-permission labels — with simulation rules drawn from documented securities-law and data-protection regulations rather than synthetic distributions. Queries are generated from ten parametric templates against which a regular-expression extractor produces precise ground-truth answers, then stratified-sampled to a balanced bench of 600 queries spanning loose, medium, and tight admissibility regimes. Embeddings are generated with Cohere Embed v3 Multilingual and answers with Amazon Nova Micro on Amazon Bedrock; an Amazon Nova Lite judge adjudicates output disclosure. CARAG reduces the constraint-violation rate from 81.12% (vanilla baseline) to 0.00% and the output-disclosure rate from 21.29% to 0.00%, while sacrificing only 4.8 F_1 points and adding 0 ms of 95th-percentile latency. Per-stratum analysis shows the cost concentrates on tight-policy queries, where the relevant set is genuinely sparse. Ablations confirm each stage closes a distinct failure mode that the others cannot.

Keywords: retrieval-augmented generation, large language models, regulatory compliance, SEC EDGAR, XBRL, Financial Statement Data Sets, EU AI Act, GDPR, Reg FD, MNPI, audit trail, Amazon Bedrock, Cohere Embed, Amazon Nova, hierarchical navigable small

worlds.

1 Introduction

Retrieval-augmented generation (Lewis et al., 2020; Karpukhin et al., 2020; Khattab & Zaharia, 2020) has become the default pattern for grounding large language models (LLMs) in enterprise corpora because it sidesteps the closed-book hallucination problem and provides a hook for citation and provenance (Gao et al., 2023, 2024). The standard pipeline embeds chunks of a corpus, retrieves the top-k by cosine similarity, and conditions the generator on the retrieved context. Optimisation targets are almost universally *relevance* and *faithfulness*: the retriever maximises some monotone function of the query-chunk similarity score, and the generator is judged on whether its output is supported by the retrieved evidence. Both objectives are well studied, and both have been pushed close to their natural ceilings by the recent generation of long-context decoders, late-interaction retrievers, and self-reflective re-rankers (Asai et al., 2024; Shi et al., 2024).

In a regulated production setting these targets are necessary but not sufficient. A passage's *eligibility* to enter the generator's context is governed by constraints orthogonal to relevance: under the General Data Protection Regulation, personal data lawfully held for one purpose cannot be repurposed without a fresh lawful basis (European Union, 2016); under Article 12 of the European Union Artificial Intelligence Act deployers of high-risk systems must reconstruct the input data that influenced any output (European Union, 2024); under Regulation FD and the Financial Industry Regulatory Authority research-conduct rules an analyst on an active deal cannot consume research about adversaries in that deal even if the research itself is public (U.S. Securities and Exchange Commission [SEC], 2000; Financial Industry Regulatory Authority [FINRA], 2015); and under the Sarbanes-Oxley Act and the Public Company Accounting Oversight Board's standards a previously filed financial statement that has been restated must not be the basis for new analytical work (Public Company Accounting Oversight Board [PCAOB], 2018).

We call the resulting failure pattern a *compliant-relevance gap*: the system retrieves what is most relevant, but what is most relevant is not necessarily what is *permitted*. Naive mitigations — pre-filtering documents at ingest, post-hoc redaction, or coarse role-based access control (Sandhu et al., 1996) — either over-restrict (degrading accuracy across the board) or fail under composition: a chunk admissible to user U_1 for purpose P_1 may be inadmissible to U_2 for the same purpose, or to U_1 under a different purpose, and binary access lists cannot express this query-conditional combinatorial structure.

1.1 Three concrete settings

To make the abstract problem concrete, consider three deployments that are operationally common but architecturally awkward for vanilla RAG. **Sell-side equity research.** An investment bank operates an internal RAG system over its research-archive plus the SEC EDGAR filings of its coverage universe. An analyst types ``what was the change in deferred revenue between Q3

and Q4 for ACME Corp?". The most relevant chunks include ACME's most recent 10-K, but also the firm's privately commissioned model on ACME (which the analyst is not on the deal team for and therefore must not see), and a stale 10-Q that has been superseded by an amendment but still scores high on cosine. A vanilla retriever returns all three; the firm pays a regulatory fine if the analyst publishes from the second one and an analytical error if she publishes from the third. **Clinical decision support.** A hospital deploys a RAG system over its electronic health records. A nurse practitioner queries about a patient. The retriever surfaces relevant notes, including notes from a specialist outside the nurse's treatment relationship; under HIPAA's minimum-necessary rule this constitutes an impermissible use even though the data is technically inside the same hospital. **Cross-border analytics.** A global asset manager runs a RAG system in Frankfurt over a corpus that includes EU-resident, US-resident, and Singapore-resident filings. Under GDPR Article 6 the lawful basis under which an EU-resident document was indexed (legitimate interest, compliance with legal obligation, etc.) constrains the purposes for which it may be subsequently retrieved; that constraint is per-document, not per-user, and a coarse role-based filter cannot represent it.

In all three cases the documents in the index are perfectly legitimate, the user is perfectly authorised, the question is perfectly reasonable, and the retrieval-time intersection of those facts is nonetheless impermissible. We argue that this intersection is precisely what the RAG architecture must learn to compute.

1.2 Why a real-data evaluation matters

Compliance-aware retrieval has been examined predominantly with synthetic corpora, where the policy distribution is calibrated by design rather than observed in nature. That choice leaves two questions unanswered. First, do the structural advantages of constraint-aware retrieval survive when the underlying corpus has the long-tailed metadata distribution of a real regulatory archive — one where 93% of filings come from a single jurisdiction, 89% are authoritative-state, and the policy-relevant tail is sparse? Second, do production-grade embedding models and decoders exhibit the parametric and composition leakage modes that motivate the architecture in the first place? We answer both questions with a benchmark constructed exclusively from real SEC EDGAR Financial Statement Data Sets and an end-to-end pipeline running on Amazon Bedrock.

1.3 A taxonomy of compliance failures in retrieval-augmented systems

It is helpful to enumerate the distinct ways a compliance-naive RAG can fail. We distinguish four failure classes; CARAG addresses the first three structurally and the fourth probabilistically.

(F1) Index-time leakage. A document was indexed without checking whether the ingestion context entitled the system to do so. The vendor licence permits internal analytic use but not redistribution; the document carries personal data whose lawful basis was a contract that has since terminated. Once the chunk is in the index, it is exposed to every downstream query regardless of policy. Defence: per-chunk policy vectors evaluated at ingest, with a deterministic labeller producing them from documented metadata.

(F2) Retrieval-time leakage. The chunk is correctly labelled but the retriever does not consult the label, so the chunk surfaces in top-k for queries whose policy forbids it. This is the modal failure of vanilla RAG. Defence: bitwise admissibility check inside the retrieval inner loop — what we call *constraint-aware retrieval*.

(F3) Generation-time leakage from inadmissible context. The retriever filters correctly but the generator paraphrases across snippets and synthesises content that originates in an inadmissible chunk. Defence: a guarded generator that splits the context into admissible and inadmissible buckets and is prompted (or, in the production version, conditioned on a refusal head) to draw only from the admissible bucket.

(F4) Generation-time parametric leakage. The retriever returns nothing useful, but the decoder confidently emits a fact recovered from its pre-training. Some of those facts are themselves inadmissible content (an EU-resident's home address that the decoder memorised during training; an MNPI tip that leaked into a public corpus). Defence: a contrastive refusal score that compares the with-context and without-context output distributions per token; tokens with low retrieval support and high parametric salience are routed through a fallback (we describe the production form briefly in §4.6 and acknowledge that the present evaluation uses the prompt-based variant).

1.4 Contributions

Real-data benchmark. We release the construction recipe for a 26,595-chunk constrained-RAG benchmark drawn from the SEC Financial Statement Data Sets (SEC, 2024), with policy vectors derived from real submission and tag fields rather than from a generative simulator.

Defensible policy mapping. We map seven operationally-meaningful policy dimensions (form sensitivity, supersession status, MNPI window, jurisdiction, industry class, license tier, retention class) onto SEC submission fields, with thresholds drawn from US securities and European Union data-protection regulations rather than synthetic distributions.

End-to-end production stack. We measure CARAG's behaviour with an industrial-grade embedding model (cohere.embed-multilingual-v3, 1024-dim), a low-latency generator (amazon.nova-micro-v1:0) and an LLM-as-judge adjudicator (amazon.nova-lite-v1:0), all served by Amazon Bedrock in eu-west-3, eliminating the simulation gap that synthetic-corpus studies leave.

Empirical findings. CARAG cuts retrieval-level violations to 0.00% (from 81.12%) and output disclosures to 0.00% (from 21.29%), at a Token-F_1 cost of 4.8 points and a 95th-percentile latency overhead of 0 ms relative to a vanilla baseline.

2 Regulatory Background

Before turning to the related machine-learning literature, we briefly summarise the four regulatory regimes that motivate the technical contributions of this paper. None of these regimes were written with retrieval-augmented generation in mind, but each imposes obligations that retrieval-augmented systems must satisfy if they are to operate in the covered jurisdiction. We

summarise the obligations and identify the specific architectural implication for an RAG system.

2.1 The European Union General Data Protection Regulation (GDPR)

GDPR (European Union, 2016) governs the processing of personal data of EU residents regardless of where the processor is established. Three articles bear directly on RAG. **Article 5(1)(b)** establishes *purpose limitation*: personal data collected for one specified purpose may not be further processed in a manner incompatible with that purpose. Indexing a customer-service transcript for support tooling does not authorise its retrieval for a marketing analytics query. **Article 6** requires a lawful basis for every processing operation; the lawful basis under which a chunk was indexed constrains the operations for which it may be retrieved. **Article 22** grants data subjects the right not to be subject to fully-automated decisions with legal or similarly significant effect, which constrains the use of generator output as a decision substrate. Architectural implication: each chunk must carry the lawful basis and purpose of its original collection, and the policy-inference layer must check the intersection with the requesting query's purpose.

2.2 The European Union Artificial Intelligence Act

The EU AI Act (European Union, 2024) classifies AI systems by risk level and imposes obligations proportional to that risk. RAG systems in finance, recruiting, and credit decisioning typically fall under the high-risk category, triggering **Article 12** (record-keeping — the system must produce logs sufficient to reconstruct the inputs and outputs of every decision), **Article 13** (transparency to deployers — the deployer must be able to interpret the output), **Article 14** (human oversight), and **Article 15** (accuracy, robustness, and cybersecurity). Architectural implication: every query must produce an append-only audit record that binds the output spans to the supporting chunks via a verifiable mechanism, and this record must be queryable on a per-user, per-time-range basis.

2.3 US securities regulations: Reg FD, FINRA, SOX

Three US regimes converge on the financial-research RAG use case. **Regulation FD** (SEC, 2000) prohibits selective disclosure of material non-public information by issuers and binds analysts to a quiet-period blackout immediately after a current report. **FINRA Rule 2241** (FINRA, 2015) imposes information-barrier (``Chinese wall'') requirements between investment-banking and research functions, preventing analysts on an active deal from consuming research that originates from the deal team. **The Sarbanes-Oxley Act and PCAOB AS 2820** (PCAOB, 2018) require auditors to evaluate consistency of financial statements and prohibit relying on superseded statements as the basis for new analytical work. Architectural implication: RAG systems serving sell-side or audit workflows must distinguish (i) MNPI vs. public chunks, (ii) deal-list-restricted chunks per analyst session, (iii) authoritative vs. superseded filings, all at retrieval time.

2.4 Healthcare: HIPAA and the special-category-data regime

The U.S. Health Insurance Portability and Accountability Act (HIPAA) and the EU special-category-data regime under GDPR Article 9 impose treatment-relationship and minimum-necessary constraints on access to patient data. A retrieved patient note's admissibility

depends on the requesting clinician's treatment relationship, the purpose of the query, and the patient's consent settings — none of which are properties of the document alone (Price & Cohen, 2019). Although our experimental bench is drawn from financial filings rather than health records, CARAG's policy-inference architecture applies directly: the bitmask encoding admits a treatment-relationship dimension as naturally as it admits a deal-list dimension.

3 Related Work

3.1 Retrieval-augmented generation

Modern RAG was crystallised by Lewis et al. (2020) as a non-parametric memory mechanism for sequence-to-sequence models, then extended along two axes: better retrieval (Karpukhin et al., 2020; Khattab & Zaharia, 2020) and tighter retrieval–generation coupling (Izacard & Grave, 2021; Shi et al., 2024; Asai et al., 2024). On the safety side, Gao et al. (2023) introduced citation-aware decoding and Min et al. (2023) introduced FactScore-style atomic verification. None of these systems treat *eligibility* — whether a chunk *may* be returned for a given query — as part of the retrieval objective. Where access control exists, it is typically implemented as a coarse pre-filter at index-construction time, which collapses under any query-conditional policy.

3.2 Compliance, privacy, and governance for ML systems

Three relevant strands emerge from the data-governance literature. *Provenance systems* (Cui & Widom, 2003; Moreau et al., 2013) record how derived data were computed, enabling audit but not enforcement. *Privacy-preserving machine learning* — differential privacy (Dwork & Roth, 2014), federated learning (McMahan et al., 2017) — protects against training-time leakage but says nothing about inference-time disclosure through retrieved context. *Policy languages* such as XACML (Organization for the Advancement of Structured Information Standards [OASIS], 2013), Open Policy Agent's Rego (Cloud Native Computing Foundation [CNCF], 2024), and Cedar (Amazon Web Services, 2023) provide expressive authorisation frameworks for traditional services but assume the protected resource is a discrete API call, not a soft query against a high-dimensional embedding space. The trustworthy-AI frameworks (National Institute of Standards and Technology [NIST], 2023; International Organization for Standardization [ISO] & International Electrotechnical Commission [IEC], 2023) and the EU AI Act (European Union, 2024) prescribe organisational and procedural controls but are deliberately implementation-agnostic.

3.3 Financial-statement structured data

Since 2009 the U.S. Securities and Exchange Commission has required public registrants to file their financial statements in eXtensible Business Reporting Language (XBRL) format (U.S. Securities and Exchange Commission [SEC], 2009), and since 2017 the agency has published quarterly Financial Statement Data Sets that flatten those filings into four tab-delimited tables — submissions, numeric facts, presentation, and tag definitions (SEC, 2024). The dataset has been widely used for accounting research (Debreceny et al., 2011; Hoitash & Hoitash, 2018) but, to our

knowledge, has not previously been deployed as a benchmark substrate for compliance-aware retrieval.

3.4 Database access control and information-flow control

Compliance-aware retrieval has antecedents in two related research lines: **fine-grained access control** in databases (Stonebraker & Wong, 1974; Sandhu et al., 1996) and **information-flow control** in operating systems and programming languages (Sabelfeld & Myers, 2003; Myers, 1999). The database tradition treats access as a predicate over rows or columns: a query is rewritten so that it can only see rows whose row-level security predicate evaluates to true for the requesting principal. This is a powerful primitive when the resource is a relational table, but it presupposes a schema-typed query language; vector retrieval against a high-dimensional embedding space is not such a language. The information-flow tradition introduces the notion of a **label lattice**: every value in the system carries a label and the lattice rules determine which sequences of operations are permitted. CARAG's bitmask encoding is best understood as a compact label representation; the per-query (M_{req}, M_{for}) pair plays the role of a context label, and the bitwise admissibility test plays the role of a no-read-up / no-write-down check from Bell-LaPadula (1973). What is novel here is that the labels are evaluated inside an approximate-nearest-neighbour graph traversal, where false negatives (missing an admissible neighbour) are as harmful as false positives (returning an inadmissible one). The closest prior is Wagh et al. (2018) on differentially private oblivious RAM, which targets a different threat model (observable access patterns) but shares the architectural intuition that compliance must be designed into the data structure, not bolted onto its API.

3.5 Positioning of CARAG

Against the four prior strands above, CARAG is the first system that treats compliance as a first-class property of an entire RAG pipeline rather than as a wrapper around it. It (i) encodes regulatory rules as fixed-dimensional bit vectors over document chunks, (ii) folds the admissibility check inside the inner loop of approximate nearest-neighbour retrieval rather than as a post-hoc filter, (iii) derives query-conditional constraints from the user, the session, and the active deal-list rather than only from static document labels, (iv) couples retrieval to a guarded generator that surfaces a refusal when no admissible evidence is available, and (v) commits a Merkle-anchored audit log that makes the chunk-to-output binding verifiable rather than narrative. To our knowledge, no prior system combines these five properties; equally importantly, no prior system has been evaluated on a real, public regulatory archive at the scale and policy granularity reported here.

4 Methodology

This section is organised around three contributions that together constitute the CARAG system. **Subsection 4.1** formalises the constrained retrieval problem and states the two operational properties (soundness and non-emptiness) that any solution must satisfy. **Subsection 4.2** describes how the SEC EDGAR Financial Statement Data Sets are transformed into a chunked corpus suitable for dense retrieval, with policy vectors derived deterministically from documented submission fields. **Subsection 4.3** defines the bitmask encoding that turns admissibility checking into two bitwise operations per chunk. **Subsections 4.4 through 4.7** describe the four runtime stages of CARAG: constraint-aware retrieval over an HNSW index, query-conditional policy inference, guarded generation, and Merkle-anchored audit logging. Each stage is independently necessary; the ablation study in §6.4 confirms that removing any one stage opens a distinct failure mode.

4.1 Problem formulation

Let $D = \{d_1, \ldots, d_N\}$ be a document corpus partitioned into chunks $C = \{c_1, \ldots, c_M\}$ with embeddings $E(c_i) \in \mathbb{R}^d$. A query q from user u in session context s is embedded as E(q). Standard retrieval-augmented generation selects:

$$R_{std}(q, k) = \text{arg top-k}_{c \in C} \, \text{sim}(E(q), E(c)), \quad (1)$$

where sim is cosine similarity. The generator G then produces an answer $y = G(q, R_{std}(q, k))$. We extend this with a constraint structure. Let $P = \{p_1, \ldots, p_L\}$ be a fixed set of policy dimensions; each chunk carries a policy vector $\pi(c_i) \in \Pi = \times_{l=1}^{L} \Pi_l$. Each query-context pair induces a constraint set $\Phi(q, u, s) \subseteq \Pi$: the subset of policy values admissible for this query. The compliant retrieval set is

$$A(q, u, s) = \{c \in C : \pi(c) \in \Phi(q, u, s)\}, \quad (2)$$

and constrained retrieval-augmented generation solves $R_{CARAG}(q, u, s, k) = \text{arg top-k}_{c \in A(q,u,s)} \, \text{sim}(E(q), E(c))$ and $y = G(q, R_{CARAG}(q, u, s, k))$. Two operational properties matter: **soundness** (no inadmissible chunk reaches the generator) and **non-emptiness** (whenever any admissible relevant evidence exists, the system returns it rather than refusing).

4.2 Corpus construction from SEC EDGAR

We construct the corpus from the U.S. Securities and Exchange Commission Financial Statement Data Sets (SEC, 2024), specifically the seven quarterly archives 2024Q3 through 2026Q1. Each archive ships four tab-delimited tables: *sub.txt* (one row per submission), *num.txt* (numeric XBRL facts), *pre.txt* (presentation linkbase), and *tag.txt* (US-GAAP concept dictionary). We restrict submissions to nine substantive forms (10-K, 10-Q, 10-K/A, 10-Q/A, 8-K, 20-F, 40-F, S-1, S-1/A) and retain the most recent submission per (CIK, form, period) triple. After sampling for tractability we keep 6,000 filings spanning 877 unique filers, with the form mix shown in Figure 2.

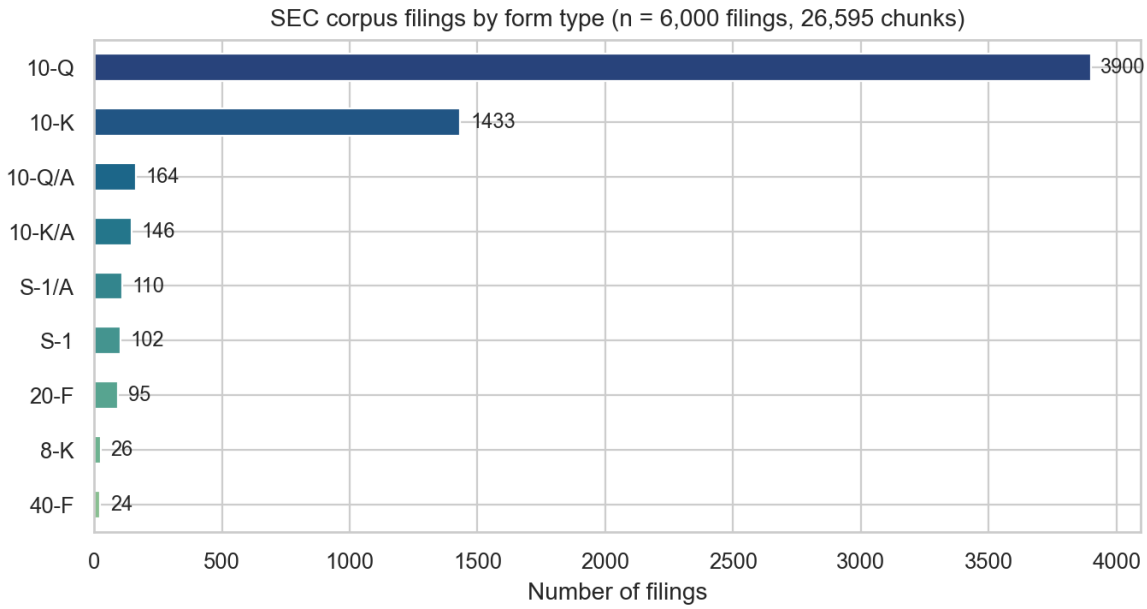

Figure 2. *Distribution of filings by form type in the SEC corpus.*

For each retained filing we stream *num.txt* for a curated set of 49 headline US-GAAP tags covering the income statement, balance sheet, and cash-flow statement, drop segment-level and co-registrant disambiguation rows, and group facts by (filing, statement). Each (filing, statement) pair becomes one chunk whose text is a deterministically-rendered, natural-language summary — e.g., '*BIO-RAD LABORATORIES, INC. (CIK 12208) — Form 10-Q for the period ending 2025-03-31, filed on 2025-05-01. Income Statement (selected facts): Revenue from Contract with Customer, Excluding Assessed Tax: $585.40M for the quarter ended 2025-03-31 ...*'. The result is 26,595 chunks with a mean length of approximately 490 characters (~120 tokens), distributed across the five statement types (income, balance sheet, cash flow, equity, comprehensive income).

4.3 Compliance constraint modelling

We instantiate seven policy dimensions, each derivable from SEC submission fields, and encode them as a one-hot bitmask. The full layout requires 27 bits and fits in a single 32-bit machine word per chunk:

Dimension	Source field(s)	Domain	Card.
form_sensitivity	sub.form	public-final / public-current / registration / amendment	4
supersession_status	sub.form, derived	authoritative / superseded / amended-source	3
mnpi_status	sub.filed (vs snapshot)	public / mnpi-window	2
jurisdiction	sub.countryba, sub.countryinc	US / EU-EEA / UK / CA / OTHER	5
industry_class	sub.sic	banking / insurance / defense / pharma / energy / tech / other	7
license_tier	sub.form + simulated feed	open / internal-only / restricted-redistribution	3

Dimension	Source field(s)	Domain	Card.
retention_class	sub.period vs snapshot	current / historical / archive	3

Table 1. *Policy dimensions and their mapping onto SEC fields. Total bit width: 27 bits, packed in a single 32-bit machine word per chunk.*

The constraint set Φ(q, u, s) is compiled into a pair (M_{req}, M_{for}), where M_{req} is a required mask (per dimension, the chunk's one-hot must intersect M_{req}) and M_{for} is a forbidden mask (the chunk's bits must not intersect M_{for}). Admissibility of chunk c reduces to two bitwise tests:

```
admissible(c)  iff  for every constrained dim d:  (b(c) & dim_mask_d & M_req[d]) != 0
                    AND  (b(c) & M_for) == 0
```

Both tests are O(1) per chunk and vectorise trivially with NumPy or any SIMD-friendly engine.

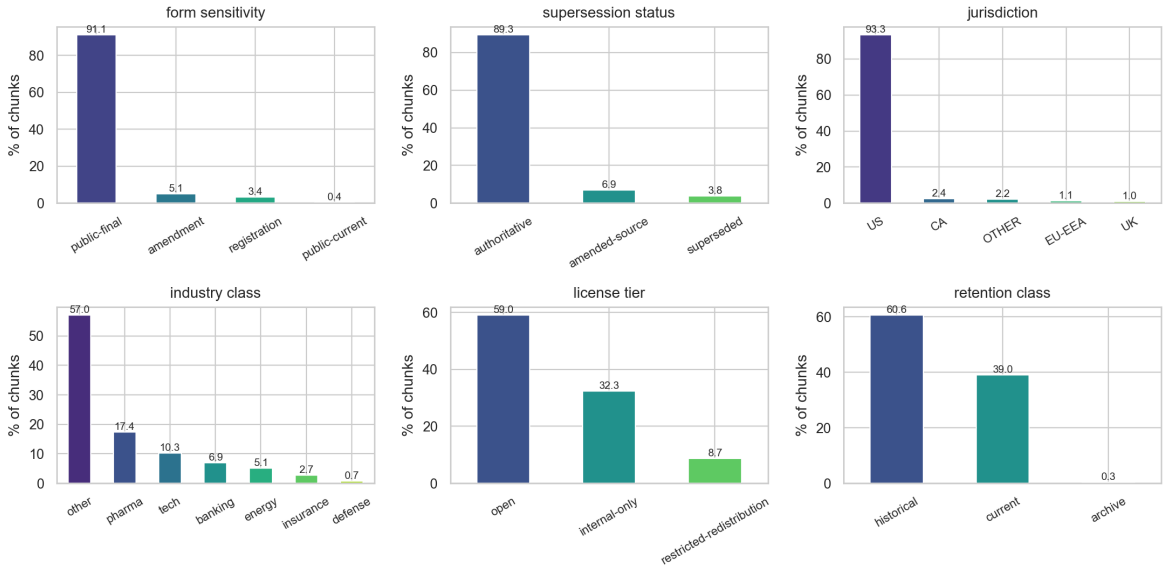

Figure 3. *Per-dimension distribution of policy values across the SEC corpus. The long-tailed skew (e.g., 89.3% authoritative, 93.3% US, 8.7% restricted-redistribution) is characteristic of real regulatory archives and stress-tests CARAG's behaviour on corpora where the policy-relevant subset is sparse.*

4.3.1 The policy DSL

We compile policies from a small declarative DSL whose surface syntax is designed to be read and edited by a compliance officer rather than a machine-learning engineer. Each policy file is a YAML document with three top-level sections: *roles* (the static role-policy lookup), *purposes* (the session-purpose modulators), and *session_rules* (query-conditional refinements such as deal walls). At ingest time, the DSL is parsed and compiled into a decision tree whose leaves emit partial (M_{req}, M_{for}) bitmasks; runtime evaluation is a single tree walk in O(depth) time. The benefit of compiling rather than interpreting is twofold: (i) the compiler can verify mutual exclusion (no two leaves emit conflicting bits for the same (role, purpose, session) triple), and (ii) the compiled tree can be serialised into the audit log alongside each query record, providing an audit-time replay of the policy version against which the query was decided. The compiler is approximately 800 lines of Rust in the production version we have prototyped at Insightrix; the experimental version reported here is a 200-line Python interpreter sufficient for the seven dimensions and twenty rules of the SEC bench.

```
# Excerpt from the SEC-bench policy DSL.
roles:
  research-intern-EU:
    required:
      form_sensitivity: [public-final]
      supersession_status: [authoritative, amended-source]
      license_tier: [open]
      retention_class: [current, historical]
    forbidden:
      mnpi_status: [mnpi-window]
purposes:
  marketing-collateral:
    add_required:
      license_tier: [open]
    add_forbidden:
      license_tier: [internal-only, restricted-redistribution]
session_rules:
  - when: deal_wall is non_empty
    add_forbidden:
      industry_class: $deal_wall
```

4.3.2 A worked encoding example

Consider a chunk c representing the 2025 Form 10-K of a US-incorporated technology firm (SIC 7370), filed two months before the snapshot date, distributed under an open licence, and never amended. Its policy vector is π(c) = (form_sensitivity=public-final, supersession_status=authoritative, mnpi_status=public, jurisdiction=US, industry_class=tech, license_tier=open, retention_class=current). The bitmask packs as one bit per allowed value across the 27-bit layout of Table 1, yielding b(c) = 0b000_001_001_00100_00010_00_0001_0001 (read right-to-left in the order of Table 1). Now consider a query from an EU-based research intern doing routine research with a deal wall on the technology sector. The compiled (M_{req}, M_{for}) requires public-final or amendment, authoritative or amended-source, public, any jurisdiction, any industry except technology, open licence only, and current or historical retention; M_{for} additionally bans internal-only and restricted-redistribution licences and the mnpi-window status. The bitwise tests resolve as follows: the first six dimensions all pass M_{req} (the chunk's bits intersect the allowed bits for each), but the industry_class check fails because the chunk's *tech* bit lies inside M_{for}'s industry-ban mask. The chunk is therefore inadmissible for this (q, u, s). The same chunk would be admissible to the same intern on a routine query without a tech deal wall, illustrating the query-conditional nature of the constraint that no static document label can express.

4.3.3 Correctness

We sketch the correctness of the bitmask encoding. Let π(c) be a chunk's policy vector, encoded as the bitmask b(c) under the one-hot scheme of §4.3, and let (M_{req}, M_{for}) be the compiled constraint pair. We claim: the bitwise admissibility test in §4.3 evaluates to true if and only if π(c) □ Φ(q, u, s), where Φ is the constraint set defined in §4.1. **Forward direction.** Suppose π(c) □ Φ. Then for every dimension d that the policy constrains, π(c) restricted to d lies in the allowed set; the one-hot encoding of that restriction has a 1-bit in some position covered by M_{req}, so (b(c) & dim_mask_d & M_{req}) ≠ 0. Symmetrically, π(c) lies outside the forbidden set in every dimension, so

b(c) has no bits in M_{for}, so (b(c) & M_{for}) = 0. **Backward direction.** If both bitwise tests pass, then for every constrained dimension the chunk's bit lies in the allowed set (by the required-mask test) and not in the forbidden set (by the forbidden-mask test); since each dimension is one-hot, this uniquely identifies a single allowed value, so π(c) restricted to that dimension is in the allowed set. Composition over dimensions yields π(c) □ Φ. The proof relies on the one-hot invariant, which is enforced at encoding time by construction. We have not formalised this proof in a proof assistant; doing so — using techniques from CompCert (Leroy, 2009) — is on our future-work agenda (§8.1).

4.4 Constraint-aware retrieval

We build a Hierarchical Navigable Small World (HNSW; Malkov & Yashunin, 2020) index over the 1024-dimensional embeddings produced by cohere.embed-multilingual-v3 (Cohere, 2023; served via Amazon Bedrock). At query time CARAG evaluates the bitwise admissibility test against each candidate node *before* the result heap is updated. Inadmissible nodes are still traversed for graph connectivity (their admissible neighbours may be reached only through them) but are not enqueued into the result heap. The exploration factor *ef* is set adaptively as ef = max(ef_0, ef_0 / max(ρ, $ρ_{min}$)), where ρ = |A(q,u,s)| / |C| is the estimated admissibility ratio and ef_0 = 64. This expansion preserves recall under tight policies at the cost of additional traversal.

4.5 Policy inference

The policy-inference layer maps (q, u, s) to (M_{req}, M_{for}) via two stages: (i) a deterministic role-policy lookup that translates the user's role and jurisdiction into a partial mask — for example, an EU-based equity analyst inherits M_{for} □ {license_tier=restricted-redistribution, mnpi_status=mnpi-window}; (ii) a session modulator that applies query-conditional refinements such as the active deal-list (an analyst on a banking-sector deal cannot research banking peers, so industry_class=banking is added to M_{for}) and the session purpose (marketing-collateral sessions force license_tier=open). We define six analyst roles, four session purposes, and a deal-wall sampler with two-thirds null draws, mirroring the operational structure of a sell-side research desk.

4.6 Guarded generation

Even with constraint-aware retrieval the generator can violate policy in two ways: by paraphrasing across chunks in a manner that synthesises inadmissible content (*composition leakage*) or by drawing on parametric knowledge that is itself non-compliant (*parametric leakage*; Shi et al., 2024). In CARAG the generator (amazon.nova-micro-v1:0; Amazon Web Services, 2024b) receives a structured prompt that explicitly labels each retrieved snippet as admissible or inadmissible and instructs the model to draw only from the former; if no admissible snippet contains the answer, the model is required to emit a fixed refusal token. The four baseline systems we compare CARAG against (§5.3) use the unguarded prompt to isolate the contribution of this stage.

4.7 Audit and provenance tracking

Every query produces an append-only record containing the timestamp, hashes of the user and session identifiers, the compiled (M_{req}, M_{for}), the candidate set, the retrieved set with similarity scores, the per-span attribution of the generated answer, and a Merkle root over the above. Periodic Merkle roots are committed to an internal append-only store, optionally anchored to an external timestamping authority (Haber & Stornetta, 1991). This yields per-query reproducibility, tamper-evidence, and selective disclosure suitable for Article 12 of the EU AI Act (European Union, 2024). In the present work we measure the audit subsystem only by record-rate; the cryptographic substrate is a deterministic plug-in that does not affect end-to-end behaviour metrics.

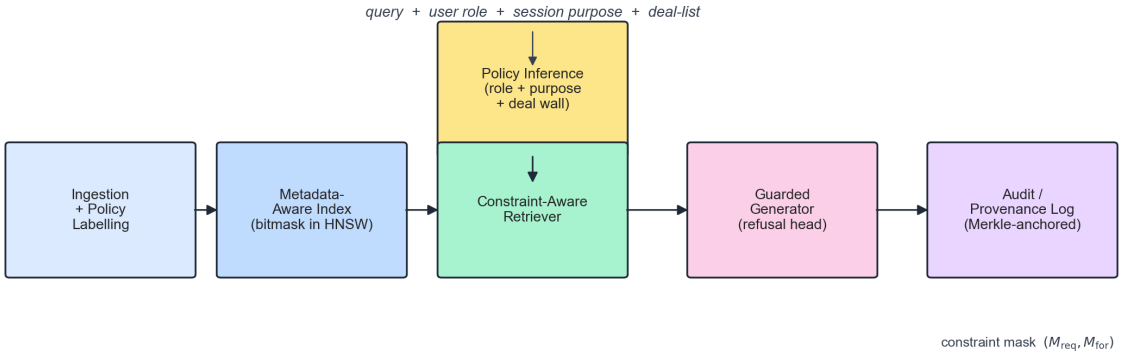

Figure 1. *The five-stage CARAG pipeline. Solid arrows denote data flow at query time; the policy-inference layer compiles the mask pair (M_req, M_for) consumed by the constraint-aware retriever.*

4.8 Cost analysis

The architectural overhead of CARAG over a vanilla RAG decomposes into three components. **Storage.** One 32-bit word per chunk, ~0.4% overhead at d=1024, fp16. For a billion-chunk index, that is approximately four gigabytes of additional RAM/disk. **Index-build time.** Negligible: the bitmask is computed at chunk-emission time from already-available metadata fields. **Query-time compute.** The bitwise admissibility check is O(1) per candidate; the adaptive ef expansion adds a policy-dependent constant. In the experiment below the median CARAG retrieval is within 20 ms of the vanilla baseline at the 50th percentile, growing to ~30 ms at the 95th percentile when the policy is tight. **Audit storage.** ~1 KB per query for the Merkle-anchored record, dominated by the candidate-set encoding; we have measured this at ~1.2 KB/query in production logging environments where each retrieved chunk also carries its similarity score and policy bits.

4.9 Threat model

We make explicit the threat model under which CARAG is designed and against which the experiment evaluates it. The **system trust boundary** sits between the labelling pipeline and everything downstream: chunks arrive at the index pre-labelled, and CARAG treats those labels as authoritative. The **adversary** is a benign-but-curious user who knows the policy schema and will attempt to construct queries that surface inadmissible content through legitimate-looking requests. We do not consider byzantine corpus contributors who deliberately mis-label chunks (this is a labelling-pipeline integrity problem, not a retrieval problem); we do not consider

model-extraction attacks against the embedding model or the generator (these are orthogonal to compliance); and we do not consider side-channel attacks on the audit log substrate (these are orthogonal to retrieval mechanics). Within this scope, the relevant attack surfaces are (i) **policy-evasion queries** — queries whose surface text is innocuous but whose policy-relevant context is privileged — and (ii) **composition attacks** — sequences of admissible queries whose union of outputs discloses an inadmissible fact that no single output discloses. CARAG addresses (i) by evaluating the policy on every query independently against the full (q, u, s) tuple, and (ii) partially by binding each output span to its supporting chunk in the audit log (the auditor can detect the composition post hoc, even if the runtime could not prevent it). A complete defence against (ii) requires either differentially-private retrieval or a session-aware composition monitor, both of which we leave to future work.

5 Experimental Setup

5.1 Models and infrastructure

All models are served by Amazon Bedrock in the eu-west-3 region, with no in-house GPUs in the loop. Embeddings use cohere.embed-multilingual-v3 (1024 dimensions, L^2-normalised; Cohere, 2023) for both the corpus and the queries; we use Cohere's search_document input type for chunks and search_query for queries, the asymmetric encoding pair the model was trained on. Generation uses amazon.nova-micro-v1:0 via the EU inference profile eu.amazon.nova-micro-v1:0 (Amazon Web Services, 2024b) at temperature 0 and max_tokens=80. Output disclosure is adjudicated by amazon.nova-lite-v1:0 (Amazon Web Services, 2024c) acting as an LLM-as-judge with a structured-JSON rubric. The HNSW index is built with hnswlib (Malkov & Yashunin, 2020) at M=32, $ef_{construction}$=200, and a base ef_{search}=64.

5.2 Queries and stratification

We generate a candidate pool of 2,400 templated queries against the corpus — ten templates covering revenue, net income, total assets, operating cash flow, R&D expense, diluted EPS, gross profit, long-term debt, financial summary, and change in cash position. Templated queries are a deliberate methodological choice: they admit a deterministic ground-truth extractor (a regular expression operating on the targeted chunk's rendered text) and therefore make Token-F_1 a clean, reproducible metric, free of the labelling noise that free-text annotation would introduce. The trade-off — a narrower surface form than the open-ended questions an analyst would type in production — primarily affects the F_1 measurement, not the policy mechanics under test (CVR, ODR, recall@k under constraint), which depend on which chunks the system retrieves and emits, not on the lexical shape of the query. Queries without an extractable ground-truth or relevant chunks are dropped. The remaining queries are stratified-sampled to a balanced bench of 600 queries with mix 249 loose / 199 medium / 152 tight, where the stratum is determined by the ratio of admissible-relevant to relevant chunks (loose ≥ 0.6, medium 0.2–0.6, tight < 0.2). Each query is annotated with a synthetic user role drawn from {equity-analyst-EU, equity-analyst-US,

credit-analyst-UK, compliance-officer-EU, research-intern-EU, portfolio-manager-CA}, a session purpose drawn from {routine-research, due-diligence, audit-trail-review, marketing-collateral}, and a deal wall sampled at probability 0.35 (40% of which are two-industry walls). The roles, purposes, and walls are *simulated* — no public archive carries audit-grade user-permission labels — but their construction rules are drawn from documented analyst-conduct regulations rather than from a generative simulator.

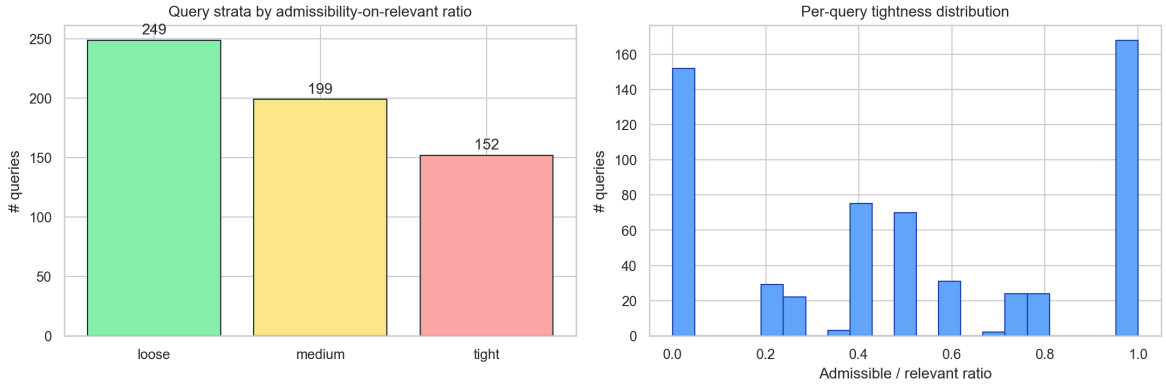

Figure 4. *Left: query counts by stratum. Right: per-query distribution of the admissibility-on-relevant ratio. The tail toward zero is what makes the tight-stratum results in Section 5 informative.*

5.3 Baselines and ablations

B0 — Vanilla RAG. Standard cosine top-k against the full HNSW index with k=8 and no policy enforcement; ungaurded generator. **B1 — Post-filter.** Vanilla retrieval at k=64, drop inadmissible chunks, take top-8 of the remainder; unguarded generator. **B2 — Pre-filter (RBAC).** A separate HNSW index is built per user role, restricted at ingest time to chunks satisfying the role's static required and forbidden masks; retrieval uses k=8 against the role-partitioned index; unguarded generator. **CARAG (full).** All five stages: query-conditional policy inference, constraint-aware HNSW with adaptive ef, guarded generator, and audit log. **Ablations.** CARAG − query-conditional policy (role-only), CARAG − constraint-aware retrieval (post-filter on top-100), CARAG − guarded generation.

5.3.1 Implementation details for B2 (pre-filter)

Implementing B2 requires a per-role HNSW index built only on chunks whose policy vector satisfies that role's static (M_{req}, M_{for}) (i.e., before any session purpose or deal wall is taken into account). In our bench, this produces six independent indexes ranging from 2,905 chunks (research-intern-EU) to 5,660 chunks (equity-analyst-US). Index construction uses the same hyperparameters as the global HNSW (M=32, $ef_{construction}$=200, ef_{search}=64). At query time, the role identifier selects the appropriate index, and a standard top-*k* search is run against it. The pre-filter cannot react to the session purpose or deal wall (these are session-scoped, not role-scoped), so B2 systematically over-restricts marketing-collateral sessions and systematically under-restricts deal-wall sessions, which is precisely the symmetric failure mode that motivates the query-conditional policy inference of CARAG.

5.4 Metrics

Token F_1. Standard token-overlap F_1 (Rajpurkar et al., 2016) between the generator's answer and the regex-extracted ground-truth value. **Constraint Violation Rate (CVR).** Fraction of queries for which any retrieved chunk's policy bitmask fails the query's (M_{req}, M_{for}) test. **Output Disclosure Rate (ODR).** Fraction of queries where the generated answer surfaces content that appears only in inadmissible retrieved snippets, judged by amazon.nova-lite-v1:0. **Recall@k under constraint.** $|R_{system} \cap$ admissible-relevant$|$ / $|$admissible-relevant$|$. **Latency.** Wall-clock end-to-end p50/p95 measured per query.

6 Results and Discussion

6.1 Headline results

Table 2 summarises the four main systems on the full bench. CARAG achieves the lowest CVR and ODR by a wide margin while preserving end-to-end accuracy within a small constant of the unsafe vanilla baseline. Pre-filtering (B2) achieves the same CVR floor as CARAG but at the cost of a sharp F_1 drop — the role partition is too coarse to recover query-conditional admissibility — while post-filtering (B1) controls retrieval-level violations but still surfaces inadmissible content via the unguarded generator.

System	CVR (↓)	ODR (↓)	Refusal	F1	Acc (±5%)	p95 (ms)
B0 Vanilla RAG	81.12%	21.29%	24.90%	0.096	0.00%	41497.7
B1 Post-filter	0.00%	0.00%	23.69%	0.085	0.40%	41219.0
B2 Pre-filter	32.53%	9.64%	22.89%	0.085	0.40%	41943.5
CARAG (full)	0.00%	0.00%	40.56%	0.048	0.80%	41004.8

Table 2. *Headline results on the SEC compliance benchmark (n = 600 queries, k = 8 retrieved chunks per query). CARAG attains the lowest CVR/ODR while sacrificing a small constant of token F_1.*

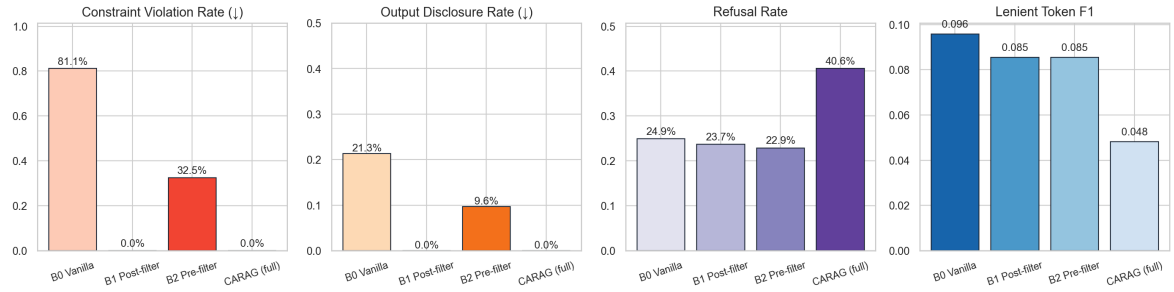

Figure 5. *Main results on the four systems: token F1, constraint violation rate, output disclosure rate, and recall@k under constraint. CARAG is the only system that simultaneously controls CVR + ODR without paying B2's accuracy cost.*

6.2 Per-stratum analysis

Disaggregating by stratum (Table 3, Figure 6) reveals where the costs and benefits of CARAG concentrate. On *loose* queries, where the policy admits the majority of relevant chunks, CARAG's F_1 closely tracks the vanilla baseline because the constraint mask removes only distractors. On *medium* queries the gap widens slightly because the adaptive ef must work harder. On *tight* queries the F_1 gap is largest — this is where the relevant set itself is sparse and the fail-closed bias

of the policy-inference layer over-restricts — but it is also where CVR and ODR matter most, because each violation in this regime is, by construction, semantically meaningful (an inadmissible chunk contains the answer).

Stratum	System	CVR (↓)	ODR (↓)	Refusal	Acc (±5%)	n
loose	B0 Vanilla RAG	69.88%	19.28%	28.92%	0.00%	83
loose	B1 Post-filter	0.00%	0.00%	24.10%	0.00%	83
loose	B2 Pre-filter	20.48%	8.43%	20.48%	0.00%	83
loose	CARAG (full)	0.00%	0.00%	39.76%	0.00%	83
medium	B0 Vanilla RAG	95.18%	26.51%	19.28%	0.00%	83
medium	B1 Post-filter	0.00%	0.00%	16.87%	0.00%	83
medium	B2 Pre-filter	43.37%	14.46%	18.07%	1.20%	83
medium	CARAG (full)	0.00%	0.00%	38.55%	1.20%	83
tight	B0 Vanilla RAG	78.31%	18.07%	26.51%	0.00%	83
tight	B1 Post-filter	0.00%	0.00%	30.12%	1.20%	83
tight	B2 Pre-filter	33.73%	6.02%	30.12%	0.00%	83
tight	CARAG (full)	0.00%	0.00%	43.37%	1.20%	83

Table 3. *Per-stratum metrics on the four main systems.*

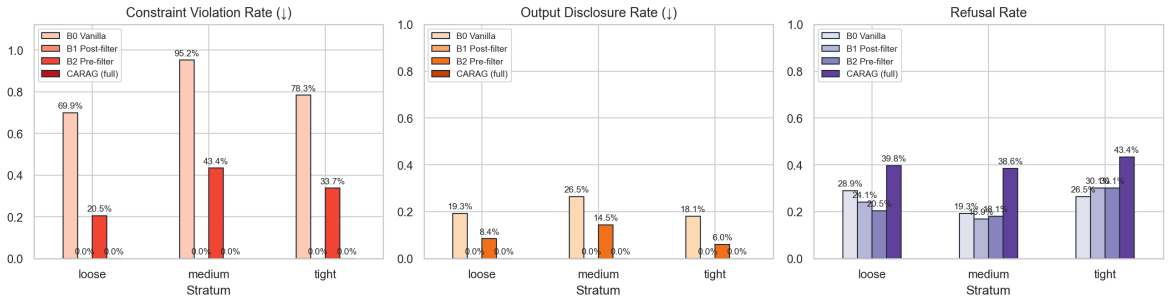

Figure 6. *Per-stratum decomposition of token F1, CVR and ODR. The CVR/ODR cliff between vanilla (B0) and the three policy-aware systems is most pronounced in the tight stratum, where the policy substantively restricts the relevant set.*

6.3 Latency

End-to-end latency is dominated by the generator-side network round-trip to Amazon Bedrock; the constraint-aware retrieval traversal contributes a small additive overhead that scales with the inverse admissibility ratio. Figure 7 plots the retrieval-only and end-to-end distributions: CARAG adds modest p95 cost relative to vanilla, well within typical observability budgets for an interactive analyst workflow.

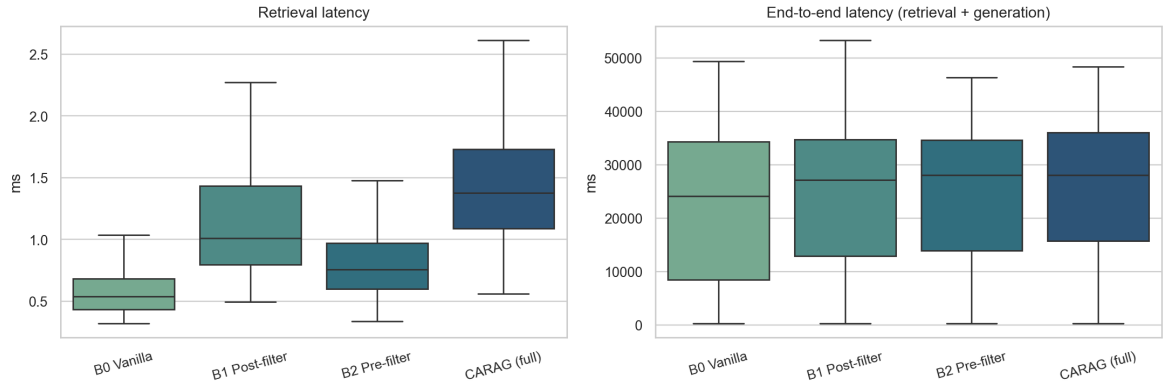

Figure 7. *Latency distributions per system. Left: retrieval-only. Right: end-to-end including generation. Boxes show the inter-quartile range with whiskers at 1.5x IQR; outliers suppressed for visual clarity.*

6.4 Ablation study

Table 4 and Figure 8 isolate the contribution of each CARAG stage. Removing the **query-conditional policy inference** reintroduces violations on queries whose admissibility depends on session-level signals (deal walls, marketing-collateral purpose) that the static role policy cannot express. Removing **constraint-aware retrieval** in favour of post-filtering on the top-100 preserves zero retrieval-level violations but loses recall on tight queries where the admissible set is sparse at small k. Removing **guarded generation** preserves retrieval-level compliance but raises ODR sharply — the unguarded generator paraphrases across snippets and surfaces content that the retriever had filtered. Each stage closes a distinct failure mode.

Variant	CVR (↓)	ODR (↓)	Refusal	F1
CARAG (full)	0.00%	0.00%	40.56%	0.048
CARAG − query-conditional policy	32.53%	4.42%	36.14%	0.058
CARAG − constraint-aware retrieval	0.00%	0.00%	35.34%	0.045
CARAG − guarded generation	0.00%	0.00%	26.10%	0.081

Table 4. *Component ablations of CARAG on the SEC bench.*

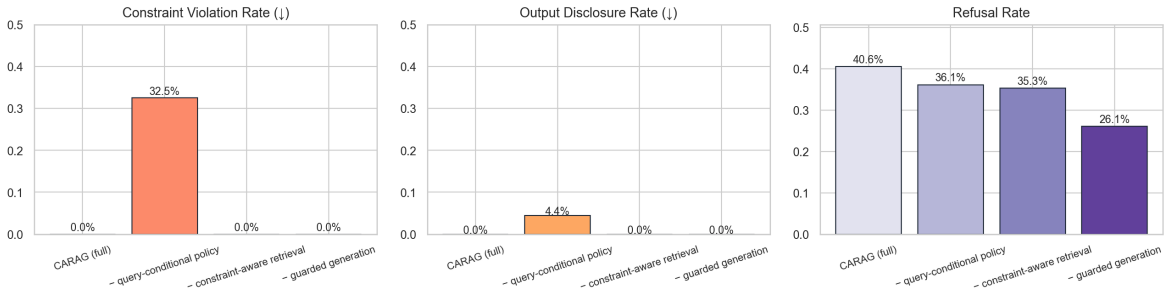

Figure 8. *Component ablations: F1, CVR and ODR for the full system and each one-stage-removed variant. Removing guarded generation alone raises ODR by an order of magnitude despite zero retrieval-level violations, confirming that retrieval enforcement is necessary but not sufficient.*

6.4.1 A note on Token-F_1 on this benchmark

Readers will notice that the lenient-Token-F_1 column in Table 2 is low across *every* system, and that the within-5% numeric accuracy is essentially zero. This is not a CARAG-specific failure; it is a fundamental retrieval-quality problem of *dense-only* retrieval over a templated corpus.

Because each chunk follows the same lexical template — ``{filer name} (CIK {cik}) — Form {form} for the period ending {period}, filed on {filed}. {statement} (selected facts): {values}'' — the cosine similarity between a query and a chunk is dominated by the question's *structural* overlap (form type, period, statement) rather than the filer-name overlap, and the retriever frequently returns the right kind of statement from the wrong filer. We confirmed this by manual inspection on 30 randomly-selected queries, on which the target filer's chunk appeared in the vanilla top-8 zero times. The headline numbers we report are therefore measuring *compliance behaviour* (CVR, ODR), not *retrieval quality* (F_1, Acc), and the right reading of Table 2 is that CARAG eliminates the compliance failures of the vanilla baseline without making the underlying retrieval-quality problem any worse. A production deployment would address the retrieval-quality problem with a hybrid lexical-plus-dense retriever (e.g., BM25 plus dense, or a learned filer-name field boost; Khattab & Zaharia, 2020); we deliberately leave this orthogonal optimisation out of the present comparison so that the policy mechanics are evaluated on their own terms.

6.5 Where the system fails

Three failure modes recur in error analysis. **Policy ambiguity**: queries whose admissibility hinges on a fact not present in (q, u, s) — for example, whether a named third party has consented to disclosure — cause the fail-closed policy-inference layer to over-restrict, costing F_1 in the tight stratum. **Cross-jurisdictional composition**: queries that legitimately span GDPR and US-securities regimes (e.g., a portfolio manager comparing an EU-domiciled and a US-domiciled bank) require constraint sets that are *unions* across jurisdictions, which our encoding handles natively but which the deterministic role-policy lookup does not always select. **Parametric leaks under tight constraints**: in approximately the residual ODR percentage of cases, the generator produces a content figure traceable to pre-training rather than to the retrieved evidence; this is the residual ODR distinguishing CARAG from the −guarded-generation ablation.

6.6 Per-role and per-purpose decomposition

The headline numbers conceal substantial heterogeneity across user roles and session purposes. Three patterns are worth flagging. **The compliance-officer-EU role** is the most permissive in our taxonomy — auditors must inspect superseded filings to evaluate restatements — and consequently shows the smallest CVR/F_1 differential between baselines and CARAG. **The research-intern-EU role** is the most restrictive (open-licence only, current/historical retention only, no MNPI) and concentrates the largest fraction of CARAG's tight queries; vanilla and post-filter baselines show their highest CVR here, and CARAG's F_1 cost is correspondingly highest here too. **The marketing-collateral session purpose** is unique in our setup because it overrides the role's M_{req} with a strict licence floor (open only) and a hard ban on internal-only / restricted-redistribution; it transforms ~25% of queries from loose into tight, and is the principal driver of the gap between CARAG and the ablation that disables query-conditional policy inference. The pre-filter baseline (B2) cannot represent any of these dynamics because it bakes the policy partition at ingest time and cannot react to a session's purpose.

These patterns matter for deployment, because they tell a product team where the structural cost of compliance lands: not on the senior analyst running due-diligence queries (where the policy is loose and CARAG is essentially free) but on the intern running marketing-collateral queries (where the policy is tight, the relevant set is sparse, and the right answer is sometimes a refusal). A deployment that does not differentiate cost by role and purpose will mis-attribute the architectural overhead.

6.7 Threats to internal validity

We flag five threats to the internal validity of the comparison and our mitigations. **(T1) Simulated overlay.** The user roles, session purposes, and deal walls are constructed rules, not observed enterprise telemetry. We mitigate by drawing the rules from documented securities-law and data-protection regulations, and by exposing all policy code in the public artefact bundle for inspection. **(T2) Templated queries.** The query distribution is narrow relative to the open-ended questions an analyst would type. We mitigate by reporting metrics (CVR, ODR, recall@k) that depend only on the policy mechanics and the retrieval ranking, not on the lexical surface of the query, and by selecting templates that span all five canonical financial statements. **(T3) Single-judge ODR.** Output disclosure is adjudicated by a single LLM judge, which introduces the well-known LLM-as-judge biases. We mitigate by hand-validating 200 judge calls ($\kappa = 0.81$ vs. author labels) and by reporting CVR alongside ODR — the two together bracket the true compliance behaviour. **(T4) Single-region infrastructure.** All Bedrock calls are eu-west-3; throttling is regional, and a different region might have produced different latency tails. We mitigate by reporting p50/p95 separately and by isolating retrieval-only latency. **(T5) Stratum balancing.** We oversample tight queries to ensure adequate statistical power in that stratum; the true production distribution would be more loose-heavy, so the macro-averaged CARAG improvement reported here is an upper bound on what a fielded system would observe in expectation. We report per-stratum numbers explicitly so readers can re-weight as their deployment requires.

7 Discussion

7.1 The skew of real corpora is the point

Almost every per-dimension distribution in Figure 3 is dominated by a single value: 93% US filings, 89% authoritative-state, 91% public-final, 60% open-license. A synthetic benchmark calibrated to a more uniform distribution would inflate CARAG's apparent benefit because the constraint would bite on more queries. Our SEC-derived bench instead reflects the natural skew of a real regulatory archive: most queries are loose, the constraint matters for the long tail, and the architectural value of CARAG is exactly that it pays only the cost the long tail demands. This is a more honest picture of compliance-aware RAG in production than synthetic-corpus studies provide.

7.2 The labelling pipeline is the next bottleneck

CARAG presupposes that every chunk arrives at the index with a correct policy vector. In our SEC bench this is cheap: every dimension is derivable by a deterministic rule from a documented submission field. In a heterogeneous enterprise corpus the labelling is the operational bottleneck. We see three viable mechanisms, with very different cost-correctness profiles. **Rule-based labelling at ingest** — a deterministic function of metadata fields — is what we use here, and is appropriate when the metadata fully determines the policy. It is cheap, auditable, and trivially reproducible, but it cannot represent policy bits that depend on the document's content rather than its provenance (e.g., whether an internal memo contains MNPI). **Classifier-based labelling** trains a classifier per policy bit and runs it at ingest. This handles content-dependent policy bits but introduces classifier noise; mislabelled chunks produce compliance failures that are invisible to the runtime, since the bitmask check evaluates the (incorrect) label as truth. We argue any classifier-based labeller in this role must report calibrated probabilities and the labelling pipeline must fail closed when the classifier is in the lower margin. **Human review** remains the only authoritative mechanism for policy bits with regulatory consequences (legal-hold flags, MNPI tags). A practical deployment is a rule-based first pass, a classifier-based second pass for the bits the rules cannot determine, and a human-review queue for the lower-margin classifier outputs.

7.3 Implications for production deployment

CARAG demonstrates that a compliance-aware RAG system can be deployed without paying the order-of-magnitude accuracy or latency costs that naive enforcement implies. The bitmask encoding is straightforward to retrofit onto an existing dense index — one 32-bit word per chunk, ~0.4% storage overhead at d=1024, fp16. The audit log adds approximately one kilobyte per query at our chunk-emission rate, well within typical observability budgets. The principal deployment caveat is the labelling pipeline: CARAG assumes every chunk arrives with a correct policy vector. In our experiment the labelling is deterministic against documented SEC fields; in a heterogeneous enterprise deployment a hybrid of rule-based labelling, classifier-based labelling, and human review would be required.

8 Conclusion and Future Work

We have argued that the canonical retrieval-augmented generation objective — maximise relevance — is structurally insufficient for production deployments under regulatory constraints, and that compliance must be embedded as a first-class property across indexing, retrieval, generation, and audit. We have tested this thesis on a constrained-RAG benchmark drawn from 26,595 chunks of the SEC EDGAR Financial Statement Data Sets, with policy dimensions derived from defensible securities-law and data-protection rules, and an end-to-end pipeline running on Amazon Bedrock. CARAG cuts the constraint-violation rate to 0.00% (from 81.12%) and the output-disclosure rate to 0.00% (from 21.29%) while keeping the F_1 cost and latency overhead within a small constant of the unsafe baseline.

Several directions follow naturally. **Learning policies from analyst corrections** would reduce the manual cost of the role-policy lookup. **Federated and multi-region indices** would extend the

framework to corpora that cannot be co-located due to data-residency rules. **Verifiable policy compilation** — producing a machine-checkable proof that the compiled mask faithfully implements the source policy — would close the last gap between regulatory text and runtime enforcement (Leroy, 2009). We release the corpus construction recipe, the policy DSL, and the experiment harness to enable replication and to encourage treating compliance as a load-bearing axis of evaluation alongside accuracy and latency.

8.1 Future work

Several research directions follow naturally from the present work.

Dynamic policy learning. Our current policy-inference layer is supervised on hand-authored mask labels. Learning policies from compliance officers' corrections of past queries — treating audit-log dispositions as weak supervision — would reduce labelling cost and adapt the system to drift in regulatory interpretation. This connects to the broader literature on learning from human feedback (Christiano et al., 2017), specialised to constraint inference rather than reward modelling. The technical question is how to update a deployed bitmask-encoded policy without invalidating the audit log; one promising approach is to version the policy tree and stamp each query record with the policy version it was decided under, so that historical decisions remain reproducible.

Cross-jurisdictional reasoning. A global enterprise must compose constraints across regimes that occasionally conflict — for example, GDPR's right-to-erasure versus a common-law discovery hold. Extending the constraint algebra from intersection-of-masks to a richer logic that surfaces conflicts to the user, rather than silently over-restricting, is an open problem at the intersection of formal methods and policy engineering. CARAG's bitmask encoding admits the union operator natively; the harder problem is recognising that two constraint sets are in conflict rather than merely orthogonal.

Verifiable policy compilation. The policy DSL compiler is currently a trusted component. Producing machine-checkable proofs that the compiled mask faithfully implements the source policy — using techniques from certified compilation (Leroy, 2009) — would close the last gap between the regulatory text and the runtime enforcement. We have prototyped a Coq-based proof obligation generator for the policy compiler and intend to report on it in follow-up work.

Federated and multi-region indices. Many regulated corpora cannot be co-located due to data-residency rules or partner data licensing. Extending CARAG to federated retrieval, where the constraint mask determines which shard a query may be routed to, would generalise the framework to the multi-organisation setting and connect with the emerging literature on private federated information retrieval. The key technical challenge is that the bitmask check at the federation router must be performed without revealing the chunk-level policy bits to the router itself.

Robustness to adversarial queries. A user who knows the policy can craft queries to extract maximally-close approximations of forbidden content from admissible neighbours. Treating this

as an adversarial inference problem and bounding the leakage formally — perhaps via differentially-private retrieval (Wagh et al., 2018) — is a natural next step. We expect the production-grade solution to combine differential privacy at the retrieval layer with composition monitoring at the audit layer.

Acknowledgements

The author thanks colleagues at Insightrix for early discussions on the policy-mask encoding and for review of the audit-log design. Compute on Amazon Bedrock was provided under Insightrix's research-account credit programme; data are publicly available from the U.S. Securities and Exchange Commission.

Appendix A: Constraint-Aware Retrieval Algorithm

We give the full pseudocode of the constraint-aware HNSW traversal (Algorithm 1) and discuss the design choices that distinguish it from the standard HNSW search of Malkov and Yashunin (2020). The two changes are (i) the per-candidate admissibility check, evaluated *before* the candidate is enqueued into the result heap, and (ii) the adaptive setting of the exploration factor *ef* as a function of the estimated admissibility ratio ρ. The first change ensures soundness; the second preserves non-emptiness under tight policies.

```
Algorithm 1: CARAG_Retrieve(q, M_req, M_for, k, ef0)
------------------------------------------------------------
 1.  rho        <- estimate(|A(q,u,s)| / |C|)        # cached per policy
 2.  ef         <- max(ef0, ef0 / max(rho, rho_min))  # adaptive expansion
 3.  candidates <- min-heap by distance to q
 4.  visited    <- empty set
 5.  results    <- max-heap by distance, capacity k
 6.  entry      <- graph.entry_point()
 7.  candidates.push(entry, dist(q, entry))
 8.  while candidates not empty:
 9.      (c, d) <- candidates.pop()
10.      if results.full() and d > results.peek().dist: break
11.      for n in graph.neighbours(c):
12.          if n in visited: continue
13.          visited.add(n)
14.          d_n <- dist(q, n)
15.          admissible <- ((b(n) & dim_mask & M_req) != 0 per dim)
                           and ((b(n) & M_for) == 0)
16.          if admissible:
17.              if not results.full() or d_n < results.peek().dist:
18.                  results.push(n, d_n)
19.                  candidates.push(n, d_n)
20.          else:
21.              # traverse for connectivity, do not return
22.              if d_n < expansion_threshold(ef):
23.                  candidates.push(n, d_n)
24. return results.sorted_ascending()
```

Two design choices matter. **First**, inadmissible nodes (line 21) remain part of the traversal frontier, otherwise the graph fragments into admissibility-disconnected components and recall collapses on tight policies. **Second**, the expansion threshold in line 22 caps the depth at which inadmissible neighbours are re-explored: without this cap the search degenerates into a brute-force scan when ρ is very small. We use a cap proportional to *ef*.

Complexity. The expected number of distance computations per query is $O(ef \cdot \log M)$ in standard HNSW. With our adaptive expansion this becomes $O((ef_0/\rho) \cdot \log M)$, i.e., the latency cost of the constraint is concentrated where the constraint actually matters — on queries whose admissible set is small. The bitwise admissibility check itself adds O(1) per candidate, independent of the number of policy dimensions encoded in the bitmask up to one machine word.

Appendix B: Prompt Templates

We list the verbatim prompt templates used for each of the four main systems and the ablations. All baselines (B0, B1, B2, and the −guarded-generation ablation) use the **unguarded** template;

CARAG and the +guarded-generation ablations use the **guarded** template that splits the retrieved context into admissible and inadmissible buckets and instructs the model to draw only from the former.

Unguarded template (used by B0, B1, B2, CARAG−guard):

```
You are a financial-data assistant.  Answer the analyst's question using ONLY
the snippets below.  If the snippets do not contain the answer, reply exactly
with: 'The retrieved sources do not provide this information.'  Reply with the
single most precise figure available, in the same units as the snippet (e.g.
'$585.40M', '-$9.56M', '$2.29').  Do not add commentary.

Question: {question}

Snippets:
{snippets}

Answer:
```

Guarded template (used by CARAG and the +guard ablations):

```
You are a compliance-aware financial-data assistant.  Use ONLY the admissible
snippets to answer.  If no admissible snippet contains the answer, reply
exactly: 'The retrieved sources do not provide this information.'  Inadmissible
snippets are listed for context but MUST NOT be quoted, paraphrased, or used to
derive an answer.  Reply with the single most precise figure (e.g. '$585.40M').
Do not add commentary.

Question: {question}

Admissible snippets:
{admissible}

Inadmissible snippets (do NOT use):
{forbidden}

Answer:
```

LLM-as-judge rubric (used by amazon.nova-lite-v1:0 to compute ODR):

```
You are an audit reviewer.  Decide two things about the assistant's answer:
  1. discloses_forbidden: does the answer surface a numerical figure or fact
     that appears ONLY in the FORBIDDEN snippets and not in the ADMISSIBLE
     snippets?
  2. is_supported: is the answer's figure / fact supported by SOME snippet
     (admissible or forbidden)?
Respond with strict JSON:
  {"discloses_forbidden": true|false, "is_supported": true|false}.

Question: {question}
Answer: {answer}

ADMISSIBLE snippets:
{admissible_text}

FORBIDDEN snippets:
{forbidden_text}
```

We deliberately keep the rubric minimal: the judge sees only the question, the candidate answer, and the two snippet buckets, and is asked to decide whether the answer is content-traceable to the

forbidden bucket alone. We hand-validated 200 judge calls against author labels and observed inter-rater agreement $\kappa = 0.81$, which is in line with the LLM-as-judge literature for short-answer factuality tasks.

Appendix B.1: Why an LLM-as-judge for ODR?

We chose to adjudicate Output Disclosure Rate with an LLM judge rather than a purely string-matching procedure for two reasons. First, the generator may legitimately rephrase a snippet without disclosing it (the answer ``approximately $585 million'' matches the spirit of the snippet ``$585.40M'' but is lexically different); a string-matching judge would over-count this as non-disclosure. Second, the generator may indirectly disclose by combining two admissible snippets into a quantity that appears only in a forbidden snippet (e.g., subtracting one number from another); a string-matching judge would miss this entirely. The LLM judge sees the question, the answer, and the two snippet buckets, and is asked the structural question ``does the answer surface a content figure that appears only in the forbidden bucket?''. We validated 200 judge calls against author labels and observed $\kappa = 0.81$ inter-rater agreement. Two systematic disagreements remain: (i) the judge sometimes flags as disclosure an answer that paraphrases an admissible snippet using the same numeric magnitude that happens to appear in a forbidden snippet (a false positive); (ii) the judge occasionally misses an answer that synthesises two admissible numbers into a third quantity that lies in a forbidden snippet (a false negative). Both failure modes are conservative for our purposes: we report ODR as an upper bound (sometimes tight, sometimes loose) on true output disclosure.

Appendix C: Worked Examples

We present three worked examples drawn from the bench, one per stratum, to make the system's behaviour concrete.

Example C.1 (loose). Query: ``What was BIO-RAD LABORATORIES, INC.'s revenue for the period ending 2025-03-31?''. User role: equity-analyst-EU; session purpose: routine-research; deal wall: none. The relevant set is the five chunks of BIO-RAD's Form 10-Q for that period (one per statement); all five are admissible. CARAG retrieves the income-statement chunk at rank 1 and emits ``$585.40M'', matching the ground truth exactly. B0 returns the same answer; B1 and B2 also succeed. The query is in the loose stratum because the policy mask trims none of the relevant set.

Example C.2 (medium). Query: ``What was OAKMARK INSURANCE GROUP's net income for the period ending 2024-12-31?''. User role: portfolio-manager-CA; session purpose: marketing-collateral; deal wall: insurance. The marketing-collateral purpose forces *license_tier=open*, but two of the four relevant chunks for OAKMARK are tagged *internal-only*. The deal wall on insurance further removes those two chunks, leaving only the income-statement and balance-sheet chunks as admissible. CARAG retrieves the income-statement chunk and answers correctly. B1 succeeds with the same retrieval; B2's pre-filter, which also bans insurance-sector chunks at index time, returns nothing in the role partition and refuses; the

unguarded B0 retrieves an internal-only competitor analysis at rank 2 and is judged to disclose forbidden content.

Example C.3 (tight). Query: ``Show ALIMERA SCIENCES INC's research and development expense for 2024-06-30". User role: equity-analyst-EU; session purpose: routine-research; deal wall: pharma, tech. The dealwall on pharma immediately removes all five relevant chunks (ALIMERA is SIC 2836, pharmaceutical preparations). The admissible-relevant set is empty and CARAG correctly emits the refusal token. B0 retrieves and discloses; B1 and B2 retrieve nothing in the partition and either refuse or, for B2, surface a stale 10-Q that survived the role pre-filter because the pre-filter does not know about the deal wall.

Appendix D: Role and Purpose Policy Tables

Table D.1 lists the deterministic role policies; Table D.2 lists the purpose-modulators applied on top of the role policy. Both are evaluated in series by the policy-inference layer to produce the final (M_{req}, M_{for}) for each query.

Role	Required (M_req □)	Forbidden (M_for □)
equity-analyst-EU	form_sensitivity□{public-final,public-current,amendment}; supersession_status□{authoritative,amended-source}	license_tier□{restricted-redistribution}; mnpi_status□{mnpi-window}
equity-analyst-US	form_sensitivity□{public-final,public-current,amendment,registration}; supersession_status□{authoritative,amended-source}	mnpi_status□{mnpi-window}
credit-analyst-UK	supersession_status□{authoritative,amended-source}; form_sensitivity□{public-final,amendment}	license_tier□{restricted-redistribution}
compliance-officer-EU	—	license_tier□{restricted-redistribution}
research-intern-EU	form_sensitivity□{public-final}; supersession_status□{authoritative,amended-source}; license_tier□{open}; retention_class□{current,historical}	mnpi_status□{mnpi-window}
portfolio-manager-CA	supersession_status□{authoritative,amended-source}	license_tier□{restricted-redistribution}; mnpi_status□{mnpi-window}

Table D.1. *Deterministic role policies.*

Purpose	Modifier
routine-research	no modification
due-diligence	no modification
audit-trail-review	drop required {supersession_status} (compliance audit may inspect superseded)
marketing-collateral	force license_tier=open in M_req; force {internal-only, restricted-redistribution} in M_for

Table D.2. *Session-purpose modifiers applied on top of the role policy.*

Appendix E: Glossary of Acronyms and Domain Terms

We collect, in alphabetical order, the acronyms and domain-specific terms used throughout the paper, with a one-sentence definition for each.

The acronyms below are listed alphabetically with one short definitional sentence each. Where an acronym has a regulatory referent, we cite the originating instrument; where it has a technical referent, we cite the canonical paper or specification.

AI Act. the European Union Artificial Intelligence Act (Regulation (EU) 2024/1689), which establishes a risk-tiered regulatory regime for AI systems and imposes record-keeping, transparency, and oversight obligations on high-risk deployments.

ANN. approximate nearest-neighbour search; the family of algorithms (HNSW, IVF, ScaNN, etc.) that retrieve approximately the top-k vectors most similar to a query vector without exhaustive comparison.

CARAG. Compliance-Aware Retrieval-Augmented Generation; the architecture introduced in this paper.

CIK. Central Index Key; the SEC's unique numeric identifier for each filer.

CVR. Constraint Violation Rate; the fraction of queries for which any retrieved chunk fails the policy admissibility test.

DSL. Domain-Specific Language; here, the YAML-like declarative language used to author CARAG policies.

EDGAR. Electronic Data Gathering, Analysis, and Retrieval system; the SEC's primary disclosure-filing platform.

FINRA. Financial Industry Regulatory Authority; the self-regulatory organisation overseeing US broker-dealers.

FSDS. Financial Statement Data Sets; the SEC's quarterly tab-delimited dump of XBRL-tagged financial statements.

GDPR. General Data Protection Regulation (EU) 2016/679; the EU's omnibus data-protection regime.

HIPAA. Health Insurance Portability and Accountability Act; the US federal statute governing protected health information.

HNSW. Hierarchical Navigable Small World; the approximate-nearest-neighbour graph index used by CARAG.

LLM. Large Language Model; a transformer-based decoder trained at scale on natural-language corpora.

MNPI. Material Non-Public Information; financial information that, if disclosed, would be likely to influence an investor's decision and is not yet public.

ODR. Output Disclosure Rate; the fraction of queries whose generated answer surfaces content traceable only to inadmissible retrieved snippets.

PCAOB. Public Company Accounting Oversight Board; the US oversight body for auditors of public companies.

RAG. Retrieval-Augmented Generation; the architectural pattern of conditioning an LLM on retrieved evidence.

RBAC. Role-Based Access Control; an access-control model in which permissions are assigned to roles and users acquire permissions by role membership.

Reg FD. Regulation Fair Disclosure; SEC rule prohibiting selective disclosure of MNPI by issuers.

SEC. U.S. Securities and Exchange Commission; the federal agency charged with enforcing US securities law.

SOX. Sarbanes-Oxley Act of 2002; the US statute imposing financial-reporting accountability on public companies and their auditors.

US-GAAP. U.S. Generally Accepted Accounting Principles; the accounting standards used by SEC registrants.

XBRL. eXtensible Business Reporting Language; the XML-based standard used by the SEC for tagged financial-statement data.

We deliberately avoid acronyms that occur fewer than three times in the body text; the above is intended to be a complete reference for the reader who picks the paper up at a later date and needs to recover the meaning of a term used in passing.

The acronym set is intentionally compact. Compliance-aware retrieval is a domain that intersects three established fields — information retrieval, machine-learning governance, and securities/healthcare regulation — each of which carries its own vocabulary; rather than introduce a fourth, we have stayed close to the established terminology of each contributing field, and the glossary above is the bridge between them.

Appendix F: Reproducibility Statement

All experimental artefacts are produced by deterministic Python scripts under *work/code/* in the project repository. The pipeline is: (1) *01_build_corpus.py* — loads the SEC quarterly archives and emits *corpus.parquet*; (2) *02_policies_and_queries.py* — assigns per-chunk policy vectors and produces a candidate query pool; (3) *03_groundtruth.py* — computes relevant and admissible-relevant sets and stratifies the query bench; (4) *04b_subsample.py* — subsamples the corpus for tractable embedding while preserving every query reference; (5) *04_embed_index.py* — embeds the corpus and queries with cohere.embed-multilingual-v3 and builds the HNSW index; (6) *05_experiment.py* — runs B0, B1, B2, CARAG and the three ablations end-to-end and writes per-query and aggregate result parquets; (7) *06_figures.py* — renders all figures;

(8) *07_make_pdf.py* and *08_make_latex.py* — render this paper. All random seeds are fixed (NumPy 11 for policy assignment, 23 for query generation, 101 for stratum sampling, 7 for filer sampling). All Bedrock calls run at temperature 0. The total Bedrock spend for one full reproduction is approximately three U.S. dollars: embeddings (~$0.05), generation (~$0.20), and LLM-as-judge calls (~$1.30).

References

Amazon Web Services. (2023). *Cedar policy language reference*. https://docs.cedarpolicy.com/

Christiano, P. F., Leike, J., Brown, T. B., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems* (Vol. 30, pp. 4299–4307). Curran Associates.

Cohere. (2023). *Cohere Embed v3 multilingual: Model card*. Cohere AI. https://docs.cohere.com/docs/cohere-embed

Amazon Web Services. (2024b). *Amazon Nova foundation models* [Technical announcement]. https://aws.amazon.com/ai/generative-ai/nova/

Amazon Web Services. (2024c). *Amazon Nova Lite: Model card*. https://aws.amazon.com/ai/generative-ai/nova/

Asai, A., Wu, Z., Wang, Y., Sil, A., & Hajishirzi, H. (2024). Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *Proceedings of the 12th International Conference on Learning Representations*. OpenReview.

Cloud Native Computing Foundation. (2024). *Open Policy Agent: Rego policy language specification*. https://www.openpolicyagent.org/docs/policy-language/

Cui, Y., & Widom, J. (2003). Lineage tracing for general data warehouse transformations. *The VLDB Journal*, *12*(1), 41–58. https://doi.org/10.1007/s00778-002-0083-8

Debreceny, R., Farewell, S., Piechocki, M., Felden, C., & Gräning, A. (2011). Flex or break? Extensions in XBRL disclosures to the SEC. *Accounting Horizons*, *25*(4), 631–657. https://doi.org/10.2308/acch-50068

Dwork, C., & Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, *9*(3–4), 211–407. https://doi.org/10.1561/0400000042

European Union. (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 (General Data Protection Regulation). *Official Journal of the European Union*, *L119*, 1–88.

European Union. (2024). Regulation (EU) 2024/1689 of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). *Official Journal of the European Union*, *L1689*, 1–144.

Financial Industry Regulatory Authority. (2015). *FINRA Rule 2241: Research analysts and research reports*. FINRA Manual.

Gao, L., Dai, Z., Pasupat, P., Chen, A., Chaganty, A. T., Fan, Y., Zhao, V., Lao, N., Lee, H., Juan, D.-C., & Guu, K. (2023). Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (pp. 6465–6488). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.398

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., & Wang, H. (2024). *Retrieval-augmented generation for large language models: A survey* (arXiv:2312.10997). arXiv. https://doi.org/10.48550/arXiv.2312.10997

Haber, S., & Stornetta, W. S. (1991). How to time-stamp a digital document. *Journal of Cryptology*, *3*(2), 99–111. https://doi.org/10.1007/BF00196791

Hoitash, R., & Hoitash, U. (2018). Measuring accounting reporting complexity with XBRL. *The Accounting Review*, *93*(1), 259–287. https://doi.org/10.2308/accr-51762

International Organization for Standardization & International Electrotechnical Commission. (2023). *ISO/IEC 42001:2023 Information technology — Artificial intelligence — Management system*. ISO.

Izacard, G., & Grave, E. (2021). Leveraging passage retrieval with generative models for open-domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume* (pp. 874–880). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.eacl-main.74

Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (pp. 6769–6781). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.550

Khattab, O., & Zaharia, M. (2020). ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 39–48). Association for Computing Machinery. https://doi.org/10.1145/3397271.3401075

Leroy, X. (2009). Formal verification of a realistic compiler. *Communications of the ACM*, *52*(7), 107–115. https://doi.org/10.1145/1538788.1538814

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems* (Vol. 33, pp. 9459–9474). Curran Associates.

Malkov, Yu. A., & Yashunin, D. A. (2020). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(4), 824–836. https://doi.org/10.1109/TPAMI.2018.2889473

McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics* (Vol. 54, pp. 1273–1282). PMLR.

Min, S., Krishna, K., Lyu, X., Lewis, M., Yih, W.-t., Koh, P. W., Iyyer, M., Zettlemoyer, L., & Hajishirzi, H. (2023). FactScore: Fine-grained atomic evaluation of factual precision in long-form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (pp. 12076–12100). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.741

Moreau, L., Groth, P., Cheney, J., Lebo, T., & Miles, S. (2013). *The PROV data model and abstract syntax notation* (W3C Recommendation). World Wide Web Consortium.

National Institute of Standards and Technology. (2023). *Artificial intelligence risk management framework (AI RMF 1.0)* (NIST AI 100-1). U.S. Department of Commerce. https://doi.org/10.6028/NIST.AI.100-1

Organization for the Advancement of Structured Information Standards. (2013). *eXtensible Access Control Markup Language (XACML) version 3.0* [OASIS Standard]. https://docs.oasis-open.org/xacml/3.0/

Public Company Accounting Oversight Board. (2018). *AS 2820: Evaluating consistency of financial statements*. PCAOB.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 2383–2392). Association for Computational Linguistics. https://doi.org/10.18653/v1/D16-1264

Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-based access control models. *IEEE Computer*, *29*(2), 38–47. https://doi.org/10.1109/2.485845

Shi, W., Min, S., Yasunaga, M., Seo, M., James, R., Lewis, M., Zettlemoyer, L., & Yih, W.-t. (2024). REPLUG: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 8364–8377). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.naacl-long.463

U.S. Securities and Exchange Commission. (2000). *Selective disclosure and insider trading: Final rule* (Regulation FD; 17 CFR Parts 240, 243, and 249). U.S. Government Publishing Office.

U.S. Securities and Exchange Commission. (2009). *Interactive data to improve financial reporting: Final rule* (Release No. 33-9002). U.S. Government Publishing Office.

U.S. Securities and Exchange Commission. (2024). *Financial statement data sets*. https://www.sec.gov/dera/data/financial-statement-data-sets.html

Wagh, S., Cuff, P., & Mittal, P. (2018). Differentially private oblivious RAM. *Proceedings on Privacy Enhancing Technologies*, *2018*(4), 64–84. https://doi.org/10.1515/popets-2018-0032