WHEN EVIDENCE UPDATES CHANGE WHICH CLINICAL QA ROUTE MAY WIN: STATE-INDEXED LATENCY EVALUATION

Anonymous authors
Paper under double-blind review

ABSTRACT

A latency percentile has no stable service meaning when an evidence update can change which physical routes still implement a clinical request. The route code and nominal runtime may remain fixed while a new contraindication makes its cached evidence stale, out of scope, or conflict-incomplete and removes that route from the eligible denominator. AdmiTrace-Q makes this membership transition executable. It compiles each query into four pre-execution certificate checks—provenance, temporal scope, conflict coverage, and freshness—ranks only routes certified against the current graph and cache watermarks, and records post-execution claim–evidence conformance without deleting failures. A fixed candidate ledger then separates measured opportunity, service-bucket exposure, and the deployed selector's response. On 847 clinician-authored questions, an unrestricted denominator lowers actual-oracle P95 from 3.24 s to 2.89 s; 98/847 (11.6%) questions contain a strictly faster ineligible route, 83/847 (9.8%) cross an actual-oracle service bucket, and 61/847 (7.2%) change the estimated-latency selector's decision. A disjoint 463-query public drug-label and guideline-update workload reproduces the three effects on 47 (10.2%), 36 (7.8%), and 29 (6.3%) questions. With all policies restricted to the same eligible set, adaptive selection reaches 3.24 s P95 and 4.87 s P99, versus 4.18 s and 6.92 s for OfflineGraph-Tuned, while a paired blinded audit improves completeness by +0.13 [+0.03, +0.23] and evidence F1 by +0.03 [+0.01, +0.06]. State-indexed eligibility prevents omitted evidence work from being rewarded as clinical QA speed and supplies a stable measurement target for systems serving mutable corpora.

1 INTRODUCTION

A route that is valid before a drug-label revision can become invalid immediately afterward without changing its code, identifier, or nominal runtime. Before the revision, its cached summary and watermark satisfy the question's evidence state; afterward, the same route omits a newly applicable contraindication and fails freshness, temporal-scope, and possibly conflict-visibility requirements. Continuing to rank that route does more than tolerate a stale answer: it changes which observations are allowed to define "fastest." The latency denominator has moved while the benchmark still presents the difference as a systems gain.

The movement is large enough to alter operational conclusions. On 847 clinician-authored questions over a 1.52M-node evidence graph, admitting all executable routes lowers the actual-oracle P95 from 3.24 s to 2.89 s. The full query-level ledger reveals three different consequences. An ineligible route is strictly faster on 98/847 (11.6%) questions; 83/847 (9.8%) questions cross a service-level bucket under measured-latency oracles; and the deployed estimated-latency selector changes its decision on 61/847 (7.2%) questions. The first two describe post-hoc opportunity and bucket exposure, whereas the third records an online routing decision. A separately authored public workload of 463 drug-label and guideline-update questions reproduces the same distinction with 47, 36, and 29 affected questions.

Current evaluation does not expose this denominator transition. RAG, GraphRAG, freshness, and biomedical QA benchmarks measure retrieval or answer behavior after execution (Lewis et al., 2020; Guu et al., 2020; Hu et al., 2025; Pan et al., 2024; Tsatsaronis et al., 2015; Jin et al., 2019; Vu et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2024; Ouyang et al., 2025; Zhang et al., 2026; Cao et al., 2026); adaptive routing benchmarks compare cost and quality among available paradigms (Wang et al., 2026). Database optimizers, conversely, rank physical plans after the logical request has already supplied a feasible set (Marcus et al., 2019; Deshpande et al., 2007; Avnur & Hellerstein, 2000). Neither perspective specifies when a physical clinical-QA route ceases to implement the same request after evidence mutates. Post-execution answer scoring can identify damage, but it cannot retroactively decide whether omitted update work deserved latency credit.

AdmiTrace-Q targets that missing measurement object: route membership indexed by the compiled request and live evidence state. Each fully instantiated route receives a deterministic certificate for provenance rendering, temporal predicates, conflict-group coverage, and cache dependencies. Screening those four clauses before latency ranking produces the state-specific set $\mathcal{A}(L(q), G_t, W_t)$; selected outputs then receive a separate claim–evidence conformance record. A query-level outcome row retains the route status, reason codes, timing, selector estimates, and conformance result, so neither NoEligibleRoute nor an unfavorable output is silently converted into a successful fast response. The accompanying all-candidate ledger evaluates the membership intervention while preserving the graph snapshot, cache state, candidate identities, prompts, hardware, and measured runtimes.

The central technical contribution is an executable latency estimand: it defines which route–state pairs implement the service, decomposes the resulting denominator change into measured-oracle and deployed-selector effects, and traces every exclusion to an evidence obligation. The systems evaluation then measures route-selection efficiency inside that corrected denominator. On the shared eligible set, AdmiTrace-Q reaches 3.24 s P95 and 4.87 s P99, compared with 4.18 s and 6.92 s for OfflineGraph-Tuned; changing only the internal route policy lowers P95 from 5.49 s to 3.24 s. A paired 240-query blinded audit reports gains of +0.13 [+0.03, +0.23] in completeness and +0.03 [+0.01, +0.06] in evidence F1. Together, the two workloads, clause relaxations, matched controls, and outcome audits show that evidence obligations change latency rankings while preserving an efficient eligible-route operating point.

2 Related Work

Fresh and conflicting evidence. FreshQA and FreshPrompt evaluate current-world factuality and search-augmented generation (Vu et al., 2024); HoH tests retrieval and generation when outdated and current evidence coexist (Ouyang et al., 2025); and Re3 learns relevance–recency retrieval and conflict-aware filtering (Cao et al., 2026). HealthContradict isolates biomedical reasoning under correct, incorrect, and contradictory contexts (Zhang et al., 2026). These works determine which evidence is retrieved or how a model behaves after receiving it. AdmiTrace-Q addresses an earlier systems question: whether a physical route that cannot enforce the requested evidence state may enter a latency competition at all. This distinction matters even when a downstream answer metric is available, because an output penalty cannot undo speed credit already assigned to work the route skipped.

Adaptive RAG and clinical QA. RAGRouter-Bench standardizes multiple RAG paradigms and studies query–corpus-dependent effectiveness–efficiency routing (Wang et al., 2026). RAG, KGQA, GraphRAG, and biomedical QA otherwise emphasize answer correctness, context relevance, reasoning, or citation support (Lewis et al., 2020; Guu et al., 2020; Izacard et al., 2021; Hu et al., 2025; Huang et al., 2019; Saxena et al., 2020; Yasunaga et al., 2021; Pan et al., 2024; Tsatsaronis et al., 2015; Jin et al., 2019; Singhal et al., 2023; Sohn et al., 2025; Liang et al., 2025). Biomedical graphs and vocabularies provide the structured evidence required by such systems (Rotmensch et al., 2017; Nicholson & Greene, 2020; Belleau et al., 2008; Breit et al., 2020). Our adaptive selector is evaluated only after all policies share the same state-specific eligible set. Thus the routing result answers which certified route is fastest, while the paper's main contribution determines the denominator within which that question is meaningful.

Correctness-gated optimization over changing state. SQL-RewriteBench assigns optimization credit only after a case-specific correctness contract succeeds and retains full-denominator outcomes (Long et al., 2026). AdmiTrace-Q shares the principle that validity precedes speed credit, but mutable evidence requires route membership to vary with the query–state pair and exposes

provenance, temporal, conflict, and freshness reasons for each transition. Query languages, learned optimizers, and adaptive processors traditionally rank plans inside a feasible logical request (Francis et al., 2018; Angles et al., 2018; Marcus et al., 2019; Deshpande et al., 2007; Avnur & Hellerstein, 2000); incremental maintenance, temporal databases, graph caching, and provenance represent the state on which feasibility can depend (Gupta & Mumick, 1995; Fan et al., 2013; McSherry et al., 2013; Shah et al., 2015; Cheng et al., 2009; Papailiou et al., 2015; Buneman et al., 2001; Cheney et al., 2009; Snodgrass, 1999; Jensen & Snodgrass, 1999). AdmiTrace-Q turns those state signals into a pre-ranking membership predicate and a paired candidate ledger, isolating denominator sensitivity from changes to the estimator or runtime stack.

3 State-Indexed Eligibility and Route Selection

A *route* P is a fully instantiated physical candidate: its family, query template, parameter binding, retrieval configuration, and cache object together define its identity. Holding that identity fixed does not hold its benchmark membership fixed. If an update advances graph state from G_t to G_{t+1} or invalidates a cache dependency in W_t, the same P may implement the request before the update and omit required work afterward. The measured object is consequently not a global plan roster but the query–state-specific set of routes that still implement the service.

The maintained labeled property graph makes this transition observable. Canonical clinical concepts are entity nodes; evidence edges carry provenance π, temporal validity τ, confidence c, and conflict-group identifiers. Derived edges retain pointers to their supporting evidence, and materialized objects retain dependency watermarks. Public biomedical sources and a private de-identified EMR slice are combined in the deployed graph, producing mixed-source evidence, updates, and access paths; Appendix A specifies the vocabulary and source details.

A question q is compiled into a logical request

$$L(q) = (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{O}),$$

where $\mathcal{E}$ collects the linked entities, $\mathcal{R}$ gives the predicates for relations and aggregation, $\mathcal{T}$ identifies the temporal window requested, and $\mathcal{O}$ sets the output policy: it specifies whether citations to edge-level evidence are required or passage-level provenance is allowed as a fallback.

A route certificate K_P is constructed entirely from information available before generation: the bound route template, enforced evidence renderer, temporal predicates, conflict-group traversal requirements, and cache dependencies with their watermarks. A physical route is *eligible* when the certificate guarantees surfaced provenance, enforces the requested temporal scope or a scoped fallback, preserves or flags covered conflict groups, and validates every reused object against relevant changes since materialization. These are capabilities and state checks, not a retrospective judgment that happens to favor a fast output.

Preselection eligibility is therefore a Boolean predicate over the route certificate and current state:

$$\mathrm{Elig}(K_P; L(q), G_t, W_t) = C_{\mathrm{prov}} \wedge C_{\mathrm{temp}} \wedge C_{\mathrm{conflict}} \wedge C_{\mathrm{fresh}}.$$

Each term is a deterministic check over route capabilities, requested temporal scope, conflict-group coverage, and watermark validation. For an evidence mutation $u : t \to t+1$, the membership transition

$$\Delta_u(P, q) = \mathrm{Elig}(K_P; L(q), G_{t+1}, W_{t+1}) - \mathrm{Elig}(K_P; L(q), G_t, W_t)$$

identifies routes added to or removed from the latency denominator. The clinically important case in this paper is $\Delta_u(P, q) = -1$: an unchanged fast path ceases to implement the request because the evidence state changed.

Let $\mathcal{A}(L(q), G_t, W_t) \subseteq \mathcal{C}(q)$ denote the routes whose certificates satisfy these constraints under graph state G_t and cache-watermark state W_t. The online serving decision is

$$\widehat{P}_{\mathrm{on}}(q) = \arg \min_{P \in \mathcal{A}(L(q), G_t, W_t)} \widehat{T}(P),$$

where $\widehat{T}(P)$ estimates request-ingress-to-first-token latency. The eligible set $\mathcal{A}$ is the measurement variable; $\widehat{T}$ is a replaceable ranking component inside it. Appendix E and Table 14 separate this invariant from the current runtime realization.

Algorithm 1 Preselection eligibility and output conformance

Require: query q, live graph G_t, watermark state W_t
Ensure: selected route, output, and trace Z, or NoEligibleRoute

$L \leftarrow \textsc{CompileRequest}(q)$
$\mathcal{C} \leftarrow \textsc{EnumerateCandidates}(L, G_t, W_t)$; $\mathcal{A} \leftarrow \emptyset$; $Z \leftarrow \emptyset$
for all $P \in \mathcal{C}$ **do**
  $K_P \leftarrow \textsc{BuildRouteCertificate}(P, L, G_t, W_t)$
  $R_P \leftarrow \emptyset$
  if $C_{\text{prov}}(K_P, L) = 0$ **then**
    $R_P \leftarrow R_P \cup \{\textsc{MissingProvenance}\}$
  if $C_{\text{temp}}(K_P, L) = 0$ **then**
    $R_P \leftarrow R_P \cup \{\textsc{TemporalScope}\}$
  if $C_{\text{conflict}}(K_P, L, G_t) = 0$ **then**
    $R_P \leftarrow R_P \cup \{\textsc{HiddenConflict}\}$
  if $C_{\text{fresh}}(K_P, W_t) = 0$ **then**
    $R_P \leftarrow R_P \cup \{\textsc{StaleState}\}$
  $Z \leftarrow Z \cup \{(P, K_P, R_P, W_t)\}$
  if $R_P = \emptyset$ **then**
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{P\}$
if $\mathcal{A} = \emptyset$ **then**
  return NoEligibleRoute, Z
for all $P \in \mathcal{A}$ **do**
  $\widehat{T}(P) \leftarrow \textsc{EstimateLatency}(P)$
$\widehat{P}_{\text{on}} \leftarrow \arg\min_{P \in \mathcal{A}} \widehat{T}(P)$
$(Y, E, T_{\text{TTFT}}) \leftarrow \textsc{Execute}(\widehat{P}_{\text{on}}, L)$
$H \leftarrow \textsc{CheckOutputConformance}(Y, E, L, G_t, W_t)$
return $\widehat{P}_{\text{on}}, Y, \textsc{EmitTrace}(Z, H, T_{\text{TTFT}}, \widehat{T})$

Algorithm 1 fixes the timing order: request compilation, certificate construction, clause checks, and estimated-latency ranking occur before selected-route execution and are included in time to first token. Output conformance is evaluated afterward, so answer quality cannot determine retrospective membership. If every certificate fails, the algorithm emits NoEligibleRoute instead of silently timing a relaxed fallback. For K route certificates, indexed clause checks and winner selection are $O(K)$.

The complete outcome row is

$$O(q) = (R_q, T_q, H_q, Z_q),$$

where R_q is the selected route or NoEligibleRoute, T_q is the observed TTFT when a route executes, H_q is the post-output conformance decision, and Z_q contains certificate reasons, watermarks, and selector estimates. Every scheduled query retains an $O(q)$ row. For a latency target ℓ, successful service is the joint event

$$U_\ell(q) = \mathbb{1}[R_q \neq \textsc{NoEligibleRoute} \ \wedge \ H_q = 1 \ \wedge \ T_q \leq \ell].$$

Thus NoEligibleRoute and failed conformance have $U_\ell = 0$ for every target; neither can improve a successful-service count by returning quickly or by disappearing from the data. The results keep TTFT, route status, and audited output quality visible as separate columns because that decomposition identifies whether a failure arose before routing, during execution, or in the produced evidence.

The offline measurement executes every candidate against the same immutable query-state snapshot to record actual time-to-first-token $T(P)$. These additional candidate executions are experimental audit work; online latency includes only compilation, certificate checks, ranking, and the selected route. Let

$$\widehat{P}_{\text{on}} = \arg\min_{P \in \mathcal{A}} \widehat{T}(P), \quad \widehat{P}_{\text{off}} = \arg\min_{P \in \mathcal{C}} \widehat{T}(P),$$
$$P^{\text{orc}}_{\mathcal{A}} = \arg\min_{P \in \mathcal{A}} T(P), \quad P^{\text{orc}}_{\mathcal{C}} = \arg\min_{P \in \mathcal{C}} T(P).$$

All four selectors break equal-latency ties by the stable, fully instantiated route identifier. Let $b(\cdot)$ map measured latency to the fixed service buckets ≤ 3 s, 3–5 s, 5–10 s, and > 10 s. The paired ledger reports

$$\begin{aligned} D_T(q) &= T(P_{\mathcal{A}}^{\mathrm{orc}}) - T(P_{\mathcal{C}}^{\mathrm{orc}}), \\ F(q) &= \mathbb{1}[P_{\mathcal{C}}^{\mathrm{orc}} \notin \mathcal{A} \ \wedge \ D_T(q) > 0], \\ B(q) &= \mathbb{1}[b(T(P_{\mathcal{C}}^{\mathrm{orc}})) < b(T(P_{\mathcal{A}}^{\mathrm{orc}}))], \\ S(q) &= \mathbb{1}[\widehat{P}_{\mathrm{on}} \neq \widehat{P}_{\mathrm{off}}]. \end{aligned}$$

Here F is a strictly faster ineligible actual-latency opportunity, B is actual-oracle bucket exposure, and S is an operational change in the estimated selector. By definition $B = 1$ implies $F = 1$; S has no set-inclusion relation with either oracle quantity because estimator error can suppress or induce an online switch. This separation makes the reported 98 opportunity cases, 83 bucket exposures, and 61 selector changes logically distinct. The intervention preserves $\mathcal{C}$, G_t, W_t, prompts, hardware, and candidate measurements while changing only eligibility membership.

Eligibility-to-observation map. Each clause has a certificate field, an executable rejection action, a latency consequence, and a direct test.

Clause	Certificate field	Rejection condition	Benchmark consequence	Evidence block
Provenance	Evidence-rendering guarantee	The route cannot surface support at the required granularity.	Unsupported route templates cannot compete.	Output audit; provenance relaxation.
Temporal	Bound temporal predicate	The route omits $\mathcal{T}$ without a scoped fallback.	Out-of-window shortcuts cannot lower a percentile.	Temporal-only relaxation; temporal family.
Conflict	Conflict-group coverage	The route cannot preserve or flag covered contradictory evidence.	One-sided synthesis cannot become the selected route.	Conflict-only relaxation; safety/aggregation slices.
Freshness	Cache dependencies and watermarks	A contributing object changed after materialization.	Pre-update reuse is excluded from selection.	Update stress; freshness-only relaxation.

The compiled request, route certificates, selector map, and trace form one measurement interface. Every query and graph state induce an eligible route set, every exclusion carries a reason code, and every selected route can be traced to the state against which it was checked. Section 4 measures the oracle and operational consequences in two separately constructed workloads.

Three plan families are considered by AdmiTrace-Q: *graph-first* addresses confident requests with low fan-out, *hybrid* addresses ambiguous linking, and *text-fallback* addresses sparse graph coverage where passage-level provenance is still available. Certificate checks evaluate each instantiated route against the same compiled request and live state before the estimator ranks eligible candidates.

The deployment instantiates that search space with intent routing, terminology-backed entity linking, parameterized graph queries, dense biomedical retrieval, and LLM synthesis. Appendix A reports model families, versions, hardware, worker settings, compiler diagnostics, and comparator configurations.

Table 1 defines the primary deployment regime: a large maintained graph and heterogeneous analyst-style questions that activate different eligibility requirements. The graph scale places the latency comparison in an operational regime, while the five query families stress distinct combinations of provenance, conflict, temporal, and freshness checks. This heterogeneity creates a meaningful route-selection problem after screening and enables the family-conditioned distortion analysis in Section 4.

For each eligible candidate route P, the planner estimates time to first token using

$$\widehat{T}(P) = T_{\mathrm{compile}} + T_{\mathrm{link}} + T_{\mathrm{elig}} + T_{\mathrm{cache}} + T_{\mathrm{graph}} + T_{\mathrm{vector}} + T_{\mathrm{post}} + T_{\mathrm{llm}}. \tag{1}$$

The estimate combines compilation, certificate screening, and workload-dependent signals: graph depth, sampled fan-out, predicate selectivity, vector top-k, and cache state. The planner chooses

Table 1: Deployment scope and benchmark summary for the reported study.

Item	Value
Graph nodes	1,521,847
Evidence edges	3,814,293
Derived edges	412,058
Public clinical documents	127,000
De-identified EMR encounters	58,000
Benchmark questions	847 clinician-authored questions
Question mix	187 entity lookup, 241 multi-hop, 168 aggregation, 139 safety, and 112 temporal queries

the eligible route with minimum $\widehat{T}(P)$. The runtime account aggregates T_{elig} and ranking in the eligibility-and-plan-selection row; Appendix H reports ranking accuracy and error structure.

AdmiTrace-Q maintains caches for subgraphs and derived summaries; an entry's validation watermark is the most recent `updated_at` timestamp across the nodes and edges contributing to it. An entry may be reused only if its cached watermark is at least as recent as the current graph watermark. Stale reuse therefore violates eligibility and cannot be tolerated as an optimization error.

4 Evidence Obligations Alter the Measured Winner

The evaluation follows the causal order of the proposed measurement interface. First, with candidate routes and their measured latencies held constant, how often does enforcing the four evidence obligations alter an actual-latency opportunity, a service bucket, or the deployed selector? Second, which clauses and query families produce the change, and does it recur under a separately constructed update workload? Third, after all systems share the corrected eligible set, how much tail latency can adaptive routing recover and what answer/evidence behavior accompanies that point?

Time to first token (TTFT) starts at request ingress and includes compilation, certificate checks, ranking, and selected-route execution. The complete outcome row in Section 3 preserves status and post-output conformance beside timing; the blinded quality audits retain their full stated sample denominators. Executing every candidate is offline measurement work used only to construct actual-latency oracles and never enters online TTFT. Primary-workload intervals use 1,000 bootstrap resamples over the 847 paired query records; the public-update setting nests a stratified query bootstrap over five fixed request-order replays.

The unchanged-stack fixed graph-first policy—called Always graph-first in Table 9 and −Cost-based planner in Table 12—is the policy-isolation control. OfflineGraph-Tuned is a stronger external deployment baseline that shares the workload, graph snapshot, worker classes, cache budget, eligibility predicate, and timing protocol while retaining its own query hints and entity linker. Its 4.18 s P95 is therefore a matched-denominator systems comparison, whereas the internal control's 5.49 s P95 isolates selection policy inside AdmiTrace-Q.

The principal denominator result is insulated from planner fitting: F and B use recorded candidate TTFT and fixed route identities, with no latency model in either oracle. Only S asks how the deployed estimator responds to the membership change. Planner quality and comparator performance are reported later as separate evidence about the system operating inside the eligible set.

4.1 Three Effects of Relaxing Eligibility

For each query, the paired ledger holds the graph snapshot, prompts, Azure hardware, cache state, instantiated candidates, estimator, and candidate TTFT measurements fixed. Eligibility is then applied or relaxed to obtain the two actual-latency oracles and the two estimated-latency selectors defined in Section 3.

Table 2: Same-stack eligibility sensitivity on the preserved candidate pool. Opportunity and bucket rows compare actual-latency oracles; selector change compares deployed estimated-latency decisions. Clause counts are query-level and non-exclusive.

Metric	Gate-relaxation summary
Actual-oracle P95, eligible / unrestricted	3.24 s / 2.89 s.
Strictly faster ineligible oracle opportunity, $F = 1$	98/847 (11.6%).
Actual-oracle bucket exposure, $B = 1$	83/847 (9.8%).
Deployed estimated-selector change, $S = 1$	61/847 (7.2%).
Median actual-oracle gap among opportunity cases	0.27 s.
Violated clauses on ineligible oracle routes	Provenance 44, temporal 31, conflict 19, freshness 28.

The three rows are separate estimands, not a funnel. The 83 bucket exposures are a subset of the 98 strictly faster actual-oracle opportunities, while the 61 selector changes compare estimated decisions and need not coincide with either oracle set. Opportunity incidence is 11.6% (95% CI 9.6%–13.9%); the selector-change rate is 7.2% (95% CI 5.7%–9.0%). Clause tags overlap because one ineligible oracle route may violate multiple obligations. Provenance and temporal scope account for the largest single-clause effects, and sensitivity peaks on temporal, safety, and aggregation questions. The 214-query public-only slice preserves the direction, tying the effect to route semantics rather than one source type.

Table 3: One-clause relaxations on the 847-query paired ledger. F is a strictly faster ineligible actual-oracle opportunity, B is actual-oracle bucket exposure, and S is an estimated-selector change. Counts are query-level; overlapping mechanisms need not sum to the all-clause row.

Relaxed requirement	Oracle P95	F	B	S
Provenance	3.02 s	62	45	34
Temporal scope	3.07 s	57	42	31
Conflict visibility	3.14 s	29	21	16
Freshness	3.10 s	41	33	24
All four	2.89 s	98	83	61

Every requirement is empirically active: relaxing any one creates nonzero opportunity, bucket, and operational-selection effects. Provenance and temporal scope are the largest individual contributors, while conflict visibility and freshness explain distinct cases rather than redundant checks. Appendix D adds confidence intervals and dominant query families, showing that the aggregate result is not driven by one undifferentiated gate.

The treatment is checked at both sides of route execution. Before selection, held-out intent classification reaches 0.940 macro-F1, entity linking reaches 96.8% accuracy on 500 manually annotated questions, and the template library covers 98.7% of the primary workload (Appendix A). After execution, the separate 200-pair citation audit identifies 23 residual error instances—8 linking, 6 temporal over-pruning, 5 conflict omission, and 4 hallucination despite supplied evidence—which remain outcome failures rather than being used to revise route membership. These diagnostics complement the deterministic certificate traces by exposing compilation and output errors that a latency-only table would hide.

4.2 Disjoint Public-Update Workload

The second setting uses 463 disjoint questions over a separately constructed public graph of drug labels and guideline updates: 103 entity, 132 multi-hop, 84 aggregation, 78 safety, and 66 temporal queries. Its evidence graph, update stream, query authorship, and graph snapshot differ from the

Table 4: Eligibility sensitivity on the 463-query public drug-label and guideline-update workload. Oracle and deployed-selector quantities follow the definitions in Section 3; clause counts are non-exclusive.

Metric	Result
Actual-oracle P95, eligible / unrestricted	3.58 s [3.39, 3.79] / 3.17 s [2.99, 3.35]
Strictly faster ineligible opportunity, $F = 1$	47/463 (10.2%), CI [7.7%, 13.2%]
Actual-oracle bucket exposure, $B = 1$	36/463 (7.8%), CI [5.5%, 10.7%]
Deployed estimated-selector change, $S = 1$	29/463 (6.3%), CI [4.3%, 8.9%]
Violated clauses on ineligible oracle routes	Prov. 21; temporal 16; conflict 10; freshness 12

Table 5: Matched eligible serving and blinded behavior audit on the public-update workload. Both policies use the same predicate, graph, queries, workers, cache budget, and latency protocol.

Metric	AdmiTrace-Q	Fixed graph-first
P95	3.58 s [3.39, 3.79]	4.31 s [4.05, 4.60]
P99	5.41 s	6.88 s
Acceptable, $n = 180$	166/180 (92.2%)	162/180 (90.0%)
Evidence F1	0.85	0.82

primary deployment, while the route families, model, estimator, worker allocation, and cache policy remain fixed. Eligibility-on and eligibility-off analyses share the graph, prompts, candidates, worker allocation, cache budget, and estimator. We replay five fixed request orders (seeds 11, 29, 47, 71, and 101), report medians across replays, and form 95% intervals with a stratified query bootstrap nested over replay order.

The same logical pattern appears under the new evidence and query process: 36 bucket exposures sit within 47 strictly faster ineligible opportunities, while 29 estimated-selector changes measure a distinct operational response. This workload-transfer test carries the route-eligibility effect beyond the primary mixed-source graph under a shared serving stack.

On the 180-item paired blinded audit, the acceptable-answer difference is +2.2 pp [-2.0, +6.7], and the evidence-F1 difference is +0.03 [+0.01, +0.05]. Together with the lower P95, these measurements show that adaptive selection within the eligible set improves latency while maintaining answer acceptability and increasing surfaced-evidence F1. Appendix B gives workload and replay details.

4.3 Latency on a Shared Eligible Set

AdmiTrace-Q and OfflineGraph-Tuned share the graph snapshot, benchmark queries, Azure worker classes, Redis cache budget, eligibility predicate, and TTFT protocol. Their serving implementations differ: OfflineGraph-Tuned retains fixed graph-first choices, hand-tuned Neo4j hints, and ColBERT-based linking, whereas AdmiTrace-Q selects among eligible route families. TextRAG and GraphRAG-OD provide non-graph and query-time construction context. The unchanged-stack fixed graph-first rows in Tables 9 and 12 isolate the selection policy itself.

Table 6: Request-ingress-to-first-token latency across 847 benchmark queries. Brackets show 95% query-bootstrap confidence intervals for P95. Lower is better.

System	P50 (s)	P95 (s)	P99 (s)
AdmiTrace-Q	1.42	3.24 [3.08, 3.41]	4.87
OfflineGraph-Tuned	1.53	4.18 [3.94, 4.43]	6.92
TextRAG	2.14	4.87 [4.61, 5.14]	7.21
GraphRAG-OD	7.83	18.42 [17.21, 19.68]	27.61

AdmiTrace-Q lowers P95 from 4.18 s to 3.24 s (22.5%) and P99 from 6.92 s to 4.87 s (29.6%) relative to OfflineGraph-Tuned under the shared eligibility protocol. Because that external comparison retains baseline-specific hints and linking, the internal fixed-policy diagnostic provides the cleaner

policy intervention: changing only route selection lowers P95 from 5.49 s to 3.24 s (Table 9). On the matched 240-query blinded audit, acceptable counts are 225/240 versus 220/240; completeness differs by +0.13 [+0.03, +0.23], evidence precision by +0.04 [+0.01, +0.07], and evidence F1 by +0.03 [+0.01, +0.06]. The eligible-set comparison therefore combines lower tail latency with improved surfaced-evidence measurements.

At the 5 s target, AdmiTrace-Q reaches 14.2 QPS. GraphRAG-OD remains the contextual construction-cost baseline, TextRAG anchors non-graph retrieval, and the fixed-policy result locates the benefit in selecting among eligible route families.

4.4 Answer and Evidence Quality at the Eligible Operating Point

Five board-certified specialists each scored all 1,000 randomized single-system displays for 500 family-stratified questions paired across AdmiTrace-Q and TextRAG (500 questions per system). Each display showed a question, identity-masked answer, and cited evidence. The four 1–5 scores report cross-item standard deviations; Fleiss' κ uses five ratings per display. Evidence precision, recall, and F1 compare returned units against a reference packet adjudicated before revealing system labels. A separate three-expert, 200-question citation audit supplies Appendix J's failure taxonomy.

Table 7: Blinded paired-system audit on 500 questions (500 displays per system). Rows report mean 1–5 Likert scores; each $\pm$ is the cross-question standard deviation, and κ is Fleiss' agreement over five ratings per display.

Metric	AdmiTrace-Q	TextRAG	κ
Correctness	4.72 ± 0.41	4.31 ± 0.68	0.78
Completeness	4.18 ± 0.63	3.87 ± 0.81	0.71
Safety	4.81 ± 0.32	4.52 ± 0.54	0.82
Actionability	4.04 ± 0.71	3.62 ± 0.89	0.69
Overall acceptable	94.2%	87.4%	0.75

The descriptive point estimates favor AdmiTrace-Q on every dimension: overall acceptability is 94.2% versus 87.4%, evidence precision is 0.91 versus 0.78, recall is 0.84 versus 0.71, and F1 is 0.87 versus 0.74. The paired OfflineGraph-Tuned audit supplies intervals aligned with the latency comparison. Both audits show latency gains without sacrificing surfaced evidence.

5 Discussion

State-indexed eligibility changes latency evaluation when evidence changes the service. Across both workloads, roughly one in ten questions exposes a faster ineligible route; service buckets change for a substantial subset, as do deployed routes, with nonzero effects from every single-clause relaxation. Temporal, safety, and aggregation questions are most exposed to omitted update checks, contradiction coverage, or evidence rendering. Eligible-route optimization still lowers tail latency, while paired audits favor completeness and surfaced-evidence F1.

Evaluation establishes transfer across two route populations with three route families and one model/provider stack: a mixed-source clinical graph and a disjoint public update workload with different evidence, revisions, and query authors. Operation requires explicit provenance, temporal-validity, conflict-group, and cache-dependency metadata; certificates must follow the local evidence model. The reusable measurement decomposition distinguishes state-dependent membership, measured opportunity, service-bucket exposure, deployed-selector behavior, and post-output conformance.

6 Conclusion

Latency is comparable only among routes implementing the same evidence-state request. AdmiTrace-Q certifies provenance, temporal scope, conflict visibility, and freshness before selection, then separately records output conformance. Across both workloads, eligibility changes operational decisions while retaining quality and latency gains: omitted update work receives no speedup credit.

AI USE STATEMENT

Generative AI tools assisted manuscript preparation, including language and presentation editing and this submission-readiness review. The authors reviewed and verified every resulting change and take responsibility for the paper's technical claims, evidence, and final content.

ETHICS STATEMENT

The de-identified private EMR component and retrospective expert audit were covered by institutional review procedures. The audit evaluates system outputs rather than clinical interventions. No model output was returned to patients or substituted for clinical decision-making, and deployment use requires expert inspection of cited evidence.

REPRODUCIBILITY STATEMENT

Section 3 specifies the route-eligibility estimand and screening procedure; Section 4 defines selectors, comparison roles, analysis units, and uncertainty; and Appendices A–J report model versions, hardware, worker settings, workload construction, clause-level analyses, public-update evaluation, and human-evaluation protocols.

REFERENCES

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pp. 72–78, Minneapolis, Minnesota, USA, 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-1909. URL https://aclanthology.org/W19-1909/.

Renzo Angles, Marcelo Arenas, Pablo Barceló, Peter A. Boncz, George H. L. Fletcher, Claudio Gutierrez, Tobias Lindaaker, Marcus Paradies, Stefan Plantikow, Juan F. Sequeda, Oskar van Rest, and Hannes Voigt. G-core: A core for future graph query languages. In *Proceedings of the 2018 International Conference on Management of Data*, pp. 1421–1432, 2018. doi: 10.1145/3183713.3190654. URL https://ir.cwi.nl/pub/27561/27561.pdf.

Alan R Aronson and François-Michel Lang. An overview of metamap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3):229–236, 2010. doi: 10.1136/jamia.2009.002733. URL https://pubmed.ncbi.nlm.nih.gov/20442139/.

Ron Avnur and Joseph M Hellerstein. Eddies: Continuously adaptive query processing. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pp. 261–272, 2000. doi: 10.1145/342009.335420. URL https://dl.acm.org/doi/10.1145/342009.335420.

François Belleau, Marc-Alexandre Nolin, Nicole Tourigny, Philippe Rigault, and Jean Morissette. Bio2rdf: Towards a mashup to build bioinformatics knowledge systems. *Journal of Biomedical Informatics*, 41(5):706–716, 2008. doi: 10.1016/j.jbi.2008.03.004. URL https://www.sciencedirect.com/science/article/pii/S1532046408000415.

Anna Breit, Simon Ott, Asan Agibetov, and Matthias Samwald. Openbiolink: A benchmarking framework for large-scale biomedical link prediction. *Bioinformatics*, 36(13):4097–4098, 2020. doi: 10.1093/bioinformatics/btaa274. URL https://arxiv.org/abs/1912.04616.

Peter Buneman, Sanjeev Khanna, and Wang Chiew Tan. Why and where: A characterization of data provenance. In *Database Theory—ICDT 2001*, volume 1973 of *Lecture Notes in Computer Science*, pp. 316–330. Springer, 2001. doi: 10.1007/3-540-44503-X_20. URL https://doi.org/10.1007/3-540-44503-X_20.

Jiawei Cao, Jie Ouyang, Mingyue Cheng, Zhaomeng Zhou, Chunli Liu, Yupeng Li, Zirui Liu, and Shijin Wang. Re3: Relevance & recency retrieval for mitigating temporal hallucination. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25735–25760, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1180. URL https://aclanthology.org/2026.acl-long.1180/.

James Cheney, Laura Chiticariu, and Wang-Chiew Tan. Provenance in databases: Why, how, and where. *Foundations and Trends in Databases*, 1(4):379–474, 2009. doi: 10.1561/1900000006. URL https://doi.org/10.1561/1900000006.

James Cheng, Yiping Ke, and Wilfred Ng. Efficient query processing on graph databases. *ACM Transactions on Database Systems*, 34(1):1–48, 2009. doi: 10.1145/1508857.1508859. URL https://dl.acm.org/doi/10.1145/1508857.1508859.

Amol Deshpande, Zachary Ives, and Vijayshankar Raman. Adaptive query processing. *Foundations and Trends in Databases*, 1(1):1–140, 2007. doi: 10.1561/1900000001. URL https://www.nowpublishers.com/article/DownloadSummary/DBS-001.

Wenfei Fan, Xin Wang, and Yinghui Wu. Incremental graph pattern matching. *ACM Transactions on Database Systems*, 38(3):18:1–18:47, 2013. doi: 10.1145/2489791. URL https://dl.acm.org/doi/10.1145/2489791.

Nadime Francis, Alastair Green, Paolo Guagliardo, Leonid Libkin, Tobias Lindaaker, Victor Marsault, Stefan Plantikow, Mats Rydberg, Petra Selmer, and Andrés Taylor. Cypher: An evolving query language for property graphs. In Gautam Das, Christopher M. Jermaine, and Philip A. Bernstein (eds.), *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*, pp. 1433–1445. ACM, 2018. doi: 10.1145/3183713.3190657. URL https://homepages.inf.ed.ac.uk/libkin/papers/sigmod18.pdf.

Ashish Gupta and Inderpal Singh Mumick. Maintenance of materialized views: Problems, techniques, and applications. *IEEE Data Engineering Bulletin*, 18(2):3–18, 1995. URL https://sigmod.org/publications/dblp/db/journals/debu/GuptaM95.html.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3929–3938. PMLR, 2020. URL https://proceedings.mlr.press/v119/guu20a.html.

Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. Grag: Graph retrieval-augmented generation. *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 4145–4157, 2025. doi: 10.18653/v1/2025.findings-naacl.232. URL https://aclanthology.org/2025.findings-naacl.232/.

Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. Knowledge graph embedding based question answering. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, pp. 105–113, 2019. doi: 10.1145/3289600.3290956. URL https://ir.webis.de/anthology/2019.wsdm_conference-2019.17/.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021. doi: 10.48550/arXiv.2112.09118. URL https://arxiv.org/abs/2112.09118.

Christian S Jensen and Richard T Snodgrass. Temporal data management. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):36–44, 1999. doi: 10.1109/69.755613. URL https://doi.org/10.1109/69.755613.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2567–2577, 2019. doi: 10.18653/v1/D19-1259. URL https://aclanthology.org/D19-1259/.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020. doi: 10.1093/bioinformatics/btz682. URL https://academic.oup.com/bioinformatics/article/36/4/1234/5566506.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Sichu Liang, Linhai Zhang, Hongyu Zhu, Wenwen Wang, Yulan He, and Deyu Zhou. RGAR: Recurrence generation-augmented retrieval for factual-aware medical question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 4006–4033, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-emnlp.214. URL https://aclanthology.org/2025.findings-emnlp.214/.

Jiang Long, Tianci Gao, Shiyuan Hao, Haochen Zhang, Shuncheng Liu, and Jiang Zhang. SQL-RewriteBench: A correctness-gated, full-denominator benchmark for statement-level SQL rewriting [experiment,analysis & benchmark]. *arXiv preprint arXiv:2607.09251*, 2026. doi: 10.48550/arXiv.2607.09251. URL https://arxiv.org/abs/2607.09251.

Ryan Marcus, Parimarjan Negi, Hongzi Mao, Chi Zhang, Mohammad Alizadeh, Tim Kraska, Olga Papaemmanouil, and Nesime Tatbul. Neo: A learned query optimizer. *Proceedings of the VLDB Endowment*, 12(11):1705–1718, 2019. doi: 10.14778/3342263.3342644. URL https://www.vldb.org/pvldb/vol12/p1705-marcus.pdf.

Frank McSherry, Derek G Murray, Rebecca Isaacs, and Michael Isard. Differential dataflow. In *Proceedings of the 6th Biennial Conference on Innovative Data Systems Research*, 2013. URL https://www.cidrdb.org/cidr2013/Papers/CIDR13_Paper111.pdf.

Microsoft. microsoft/graphrag: v0.3.2. https://github.com/microsoft/graphrag/releases/tag/v0.3.2, 2024. Software release, August 26, 2024.

David N Nicholson and Casey S Greene. Constructing knowledge graphs and their biomedical applications. *Computational and Structural Biotechnology Journal*, 18:1414–1428, 2020. doi: 10.1016/j.csbj.2020.05.017. URL https://doi.org/10.1016/j.csbj.2020.05.017.

Jie Ouyang, Tingyue Pan, Mingyue Cheng, Ruiran Yan, Yucong Luo, Jiaying Lin, and Qi Liu. HoH: A dynamic benchmark for evaluating the impact of outdated information on retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6036–6063, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.301. URL https://aclanthology.org/2025.acl-long.301/.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599, 2024. doi: 10.1109/TKDE.2024.3352100. URL https://arxiv.org/abs/2306.08302.

Nikolaos Papailiou, Dimitrios Tsoumakos, Panagiotis Karras, and Nectarios Koziris. Graph-aware, workload-adaptive sparql query caching. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pp. 1777–1792, 2015. doi: 10.1145/2723372.2723714. URL https://www.cs.au.dk/~karras/sparql_caching.pdf.

Maya Rotmensch, Yoni Halpern, Abdulhakim Tlimat, Steven Horng, and David Sontag. Learning a health knowledge graph from electronic medical records. *Scientific reports*, 7(1): 5994, 2017. doi: 10.1038/s41598-017-05778-z. URL https://doi.org/10.1038/s41598-017-05778-z.

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4498–4507, 2020. doi: 10.18653/v1/2020.acl-main.412. URL https://doi.org/10.18653/v1/2020.acl-main.412.

Neil Shah, Danai Koutra, Tianmin Zou, Brian Gallagher, and Christos Faloutsos. Timecrunch: Interpretable dynamic graph summarization. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1055–1064, 2015. doi: 10.1145/2783258.2783321. URL https://dblp.org/rec/conf/kdd/ShahKZGF15.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Abubakr Babiker, Nathanael Schärli, Aakanksha Chowdhery, Philip Mansfield, Dina Demner-Fushman, Blaise Agüera y Arcas, Dale Webster, Greg S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023. doi: 10.1038/s41586-023-06291-2. URL https://doi.org/10.1038/s41586-023-06291-2.

Richard Thomas Snodgrass. *Developing time-oriented database applications in SQL*. Morgan Kaufmann, 1999. URL https://www.sigmod.org/publications/anthology/vol6/systems/timeOriented/timeOriented.htm.

Jiwoong Sohn, Yein Park, Chanwoong Yoon, Sihyeon Park, Hyeon Hwang, Mujeen Sung, Hyunjae Kim, and Jaewoo Kang. Rationale-guided retrieval augmented generation for medical question answering. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 12739–12753, Albuquerque, New Mexico, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.naacl-long.635. URL https://aclanthology.org/2025.naacl-long.635/.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R. Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, Yannis Almirantis, John Pavlopoulos, Nicolas Baskiotis, Patrick Gallinari, Thierry Artières, Axel-Cyrille Ngonga Ngomo, Norman Heino, Éric Gaussier, Liliana Barrio-Alvers, Michael Schroeder, Ion Androutsopoulos, and Georgios Paliouras. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16(1):138:1–138:28, 2015. doi: 10.1186/s12859-015-0564-6. URL https://doi.org/10.1186/s12859-015-0564-6.

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and Thang Luong. FreshLLMs: Refreshing large language models with search engine augmentation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13697–13720, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.813. URL https://aclanthology.org/2024.findings-acl.813/.

Ziqi Wang, Xi Zhu, Shuhang Lin, Haochen Xue, Minghao Guo, and Yongfeng Zhang. RAGRouter-Bench: A dataset and benchmark for adaptive RAG routing. *arXiv preprint arXiv:2602.00296*, 2026. doi: 10.48550/arXiv.2602.00296. URL https://arxiv.org/abs/2602.00296.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 535–546, 2021. doi: 10.18653/v1/2021.naacl-main.45. URL https://doi.org/10.18653/v1/2021.naacl-main.45.

Boya Zhang, Alban Bornet, Rui Yang, Nan Liu, and Douglas Teodoro. HealthContradict: Evaluating biomedical knowledge conflicts in language models. *npj Digital Medicine*, 9:152, 2026. doi: 10.1038/s41746-025-02336-0. URL https://doi.org/10.1038/s41746-025-02336-0.

A Implementation and Workload Details

All experiments ran on Microsoft Azure. The graph store uses 5× Standard E32ads v5 instances (32 vCPUs, 256 GiB RAM, AMD EPYC 7763) running Neo4j 5.15.0 Enterprise with 128 GiB page

cache. The query service runs on 2× Standard E16ads v5 instances (16 vCPUs, 128 GiB RAM) using Python 3.11 and PyTorch 2.1.2. The vector index runs on 1× Standard NC24ads A100 v4 instance (24 vCPUs, 220 GiB RAM, NVIDIA A100 80GB, CUDA 12.2) hosting Faiss 1.7.4 HNSW with $M = 32$ and `efSearch`=64. Redis 7.2.3 provides a 64 GiB cache with `allkeys-lru`, and the LLM is Azure OpenAI GPT-4-Turbo (`gpt-4-1106-preview`, 128K context, temperature 0.2, `max_tokens`=2048).

The compiler uses a fine-tuned DistilBERT intent classifier, lexical and fuzzy linking over UMLS-style aliases, and dense biomedical retrieval initialized from encoders in the BioBERT/ClinicalBERT family (Aronson & Lang, 2010; Lee et al., 2020; Alsentzer et al., 2019). Template selection chooses among 47 parameterized Cypher queries (Francis et al., 2018). On a held-out set of 2,480 questions, the intent classifier reaches macro-F1 0.940. The entity-linking cascade reaches 96.8% accuracy on a manually annotated set of 500 questions. Template selection covers 98.7% of the 847-question benchmark.

TextRAG uses BM25 retrieval in Elasticsearch 8.11.3, ColBERT reranking, top-10 passage retrieval, and GPT-4-Turbo with passage-level citations. GraphRAG-OD performs query-time graph construction using the Microsoft GraphRAG implementation (v0.3.2) (Microsoft, 2024), followed by graph traversal and summarization. The internal fixed-policy control is AdmiTrace-Q with route selection fixed to graph-first. OfflineGraph-Tuned uses the same evidence graph, 847-query benchmark, Azure worker classes, Redis cache budget, eligibility predicate, and TTFT protocol as AdmiTrace-Q, but retains its hand-tuned Neo4j query hints and ColBERT-based entity linking. GraphRAG-OD therefore supplies construction-cost context, TextRAG supplies non-graph answer/evidence context, OfflineGraph-Tuned supplies a matched-denominator deployment comparison, and the internal fixed policy isolates route selection. Table 26 pairs the deployment comparison with a blinded behavior audit from the same benchmark population.

Table 8: Blinded provenance audit on the same 500 questions per system as Table 7. Higher is better.

Metric	AdmiTrace-Q	TextRAG
Evidence precision	0.91	0.78
Evidence recall	0.84	0.71
F1	0.87	0.74
Fleiss' κ	0.73	0.68

B Disjoint Public-Update Workload

The public-update workload is constructed separately from the primary benchmark. Its evidence graph contains public drug labels and clinical guideline releases; document revisions define a chronological update stream, and every query is answered against the snapshot and watermark active at its replay time. A separate pool of pharmacists and clinical informaticians authored the questions from the public evidence sources without access to primary-workload prompts or route outcomes. The final set contains 463 questions: 103 entity, 132 multi-hop, 84 aggregation, 78 safety, and 66 temporal queries.

The implementation retains the three candidate families, four eligibility clauses, GPT-4-Turbo synthesis configuration, worker classes, Redis budget, and latency estimator described in Appendix A. Eligibility-on and eligibility-off analyses share a graph snapshot, candidate enumeration, prompt bundle, hardware allocation, and request order. The five replay orders use seeds 11, 29, 47, 71, and 101. Reported latency values are medians across replays; intervals use a stratified query bootstrap nested over replay order, preserving the five query-family denominators.

The matched fixed-policy comparison applies the same eligibility predicate to both policies. A 180-item sample is stratified by query family and evaluated by 4 domain experts. Each blinded item contains the public-source question, one identity-masked answer, and its surfaced evidence. Acceptability requires satisfactory correctness, completeness, safety, and actionability; evidence F1 uses returned evidence units as the scoring unit. The paired acceptable-answer difference is +2.2 pp [-2.0, +6.7], and the evidence-F1 difference is +0.03 [+0.01, +0.05].

Across settings, strictly faster ineligible opportunities change from 11.6% to 10.2%, actual-oracle bucket exposure from 9.8% to 7.8%, and estimated-selector changes from 7.2% to 6.3%. The separately constructed graph, update process, and query authorship show that all three effects recur under a different route population.

C Operational Support for Plan Selection, Cache Reuse, and Ablations

Table 9: Plan-selection quality against the eligible actual-latency oracle on the 847-query benchmark. Higher oracle match and lower regret are better.

Strategy	Oracle Match	P95 (s)	Regret (s)
Cost-based (AdmiTrace-Q)	94.2%	3.24	0.18
Always graph-first	67.4%	5.49	1.82
Always hybrid	58.1%	4.73	1.24
Random	33.3%	7.21	3.47

The oracle is the fastest eligible plan family measured under the same graph snapshot and cache state, and regret is averaged over all 847 queries. Even after ineligible routes are screened out, one fixed policy remains a poor proxy for the fastest eligible family. When the planner misses, the penalty appears as tail delay rather than negligible noise. The cost-based planner therefore matters because it searches the eligible set more effectively. Appendix H reports additional planner-fit diagnostics.

Table 10: Cold versus warm cache latency. Lower latency and higher hit rate are better.

State	Hit Rate	P50 (s)	P95 (s)	P99 (s)
Cold	0.0%	1.87	6.78	9.42
Warm	67.3%	1.42	3.24	4.87

Table 10 shows that reuse is operationally material: priming raises hit rate to 67.3% and cuts P95 latency from 6.78 s to 3.24 s. The gain is concentrated in the tail rather than only at the median, so cache state must be fixed when comparing tail latency in this deployment. The warm row also anchors the paper's main operating point for matched latency, planner quality, and ablation results, which keeps those comparisons on the same service regime. In other words, the headline 3.24 s result is not mixing warm-cache and cold-cache conditions. The table therefore turns cache state from a background implementation detail into a controlled part of the benchmark contract.

Table 11: Cache behavior under injected updates across 1,968 auditable reuse events in the archived replays. One auditable reuse event is one attempted cache or derived-summary reuse checked against the current watermark. Stale-hit entries report whether any stale reuse was observed within each setting; aggregated across all rows, the archived replays contain 0/1,968 observed stale-hit events. Lower false-invalidation rates and lower latency are better.

Upd/s	Hit Rate	False Inv.	Stale Hit	P95 (s)
0	67.3%	0.0%	0 obs.	3.24
5	64.2%	1.7%	0 obs.	3.41
20	58.7%	4.3%	0 obs.	3.82
100	42.1%	12.4%	0 obs.	5.21

Table 11 shows the safety side of that reuse. Across the archived replays, stale-hit monitoring records 0/1,968 observed stale-hit events, while false invalidations rise from 0.0% to 12.4% as update pressure increases from 0 to 100 upd/s. Treating those replays as a zero-count sample yields a one-sided 95% Clopper–Pearson upper bound of 0.15% on the stale-hit rate under the measured replay conditions. The stress test therefore documents conservative invalidation under churn, the intended trade-off for enforcing freshness in this workload.

Under heavier churn, the system gives up hit rate and accepts more false invalidations rather than serve stale state. This asymmetry makes freshness an enforced route-eligibility cost: update pressure raises tail latency because the runtime recomputes evidence instead of crediting stale reuse.

Table 12: Runtime ablation for AdmiTrace-Q. Lower latency and higher QPS are better.

Configuration	P50 (s)	P95 (s)	QPS	Evid. Prec.
Full AdmiTrace-Q	1.42	3.24	14.2	0.91
− Cost-based planner	1.68	5.49	9.8	0.91
− Subgraph cache	1.87	6.78	8.4	0.91
− Derived summaries	1.58	4.12	11.7	0.89
− Fan-out caps	1.52	8.73	7.1	0.91
− All caching	1.91	7.21	6.8	0.91

Table 12 locates the tail-latency gain within the eligible operating point across routing and validated reuse. The Full AdmiTrace-Q row is the same warm-cache, no-update operating point reported in Tables 6 and 9. Removing the planner or the subgraph cache produces the largest latency increases, while removing derived summaries also lowers evidence precision. The −Cost-based planner row is the internal fixed-policy diagnostic used throughout the paper. Together with the paired eligibility analyses, these rows distinguish the roles of route screening, adaptive selection, and validated reuse.

Table 13: Compute accounting for the reported experiment blocks.

Experiment block	Worker allocation and measured unit	Wall-clock accounting
Primary latency, distortion, and planner suite	5× E32ads v5, 2× E16ads v5, 1× NC24ads A100 v4, Redis; 847 queries per complete pass.	3.8 h per complete pass; 34.6 h total elapsed.
Public-update workload	Same worker classes and cache allocation; 463 queries under each fixed request order.	2.4 h per replay; 12.0 h across five replays.
Cache and update stress	Same serving stack; no-update plus 5/20/100 upd/s settings.	1.6 h per setting; 6.4 h total elapsed.
Blinded audit inference	Same serving stack for answer and evidence generation; expert review time excluded.	0.9 h per 500 outputs; 3.1 h total inference elapsed.
Total allocated compute	E32ads, E16ads, and A100 vector-index workers across all reported blocks.	280.5 E32 node-h; 112.2 E16 node-h; 56.1 A100 GPU-h.

Table 13 reports wall-clock and resource accounting for each block. Node-hours multiply elapsed time by allocated workers; the A100 hosted Faiss rather than the externally served GPT-4-Turbo endpoint. Primary and public-update runs use identical worker classes, update stress reuses the online stack, and expert annotation time is excluded from system compute.

Table 14: Invariant eligibility specification versus the current AdmiTrace-Q instantiation.

Item	Invariant specification	Current instantiation
Timed object	Lowest measured TTFT among routes eligible under $L(q)$, G_t, and W_t.	Three route families searched by a cost-based selector on the Azure stack.
Provenance	The route certificate guarantees evidence rendering; completed output is checked claim by claim.	UMLS-backed linking, 47 Cypher templates, and edge- or passage-level rendering.
Temporal scope	The certificate binds the requested window or a scoped fallback.	Temporal filters and scoped fallback generation.
Conflict visibility	The route preserves or flags covered conflict groups.	Conflict-group storage and route-specific evidence rendering.
Freshness	Every reused object passes dependency-watermark validation.	Redis reuse with `updated_at` dependency watermarks.

Table 14 separates the measured object from its runtime realization. The left column fixes the semantics used in the paired ledger; the right column records how AdmiTrace-Q realizes them.

Table 15: Comparator map. Each analysis has one selector and one denominator.

Analysis	Selector and denominator	Evidence
Eligibility effect	Actual-latency oracles for opportunity and buckets; estimated selectors for online change, all on one paired ledger.	Clause, query-family, public-only, and public-update analyses.
Serving latency	Deployed AdmiTrace-Q versus OfflineGraph-Tuned on a shared eligible set; internal fixed graph-first control for policy isolation.	Planner regret, cache state, update stress, and ablations.
Output quality	AdmiTrace-Q versus TextRAG on 500 questions; paired AdmiTrace-Q versus OfflineGraph-Tuned audit on 240.	Acceptability, evidence precision/recall/F1, and citation errors.

Table 15 prevents selector leakage across analyses: actual-latency oracles diagnose opportunity and bucket exposure, estimated selectors measure online decisions, and deployed-policy comparisons use one shared eligible set.

D Clause-Specific Eligibility Effects

Each clause is relaxed in turn while the graph snapshot, prompt bundle, Azure worker classes, candidate families, cache budget, estimator, and candidate measurements remain fixed. The full-eligibility baseline is Section 4. As in the main result, opportunity and bucket columns use actual-latency oracles, while selector change compares estimated-latency decisions.

Every single-clause relaxation produces non-zero oracle opportunities, selector changes, and bucket exposures. Provenance and temporal scope are the largest individual sources, freshness remains active on update-sensitive questions, and conflict visibility contributes a smaller distinct effect. The full relaxation is largest because one route can violate several requirements.

E Eligibility Mechanisms

Table 17 connects each certificate clause to the state it reads, the route it excludes, and the latency work that exclusion preserves.

The certificate inputs are available before answer generation; completed answers are handled by the separate conformance check in Algorithm 1. This division avoids using output quality to delete unfavorable latency observations.

Table 16: Clause-specific eligibility sensitivity on 847 queries. Only the named clause is relaxed.

Relaxed clause	Oracle P95	Faster ineligible opportunity	Selector change	95% CI	Bucket exposure	Dominant families
Provenance only	3.02 s	62/847 (7.3%)	34/847 (4.0%)	[2.7%, 5.5%]	45/847 (5.3%)	Safety; aggregation
Temporal only	3.07 s	57/847 (6.7%)	31/847 (3.7%)	[2.4%, 5.1%]	42/847 (5.0%)	Temporal; aggregation
Conflict only	3.14 s	29/847 (3.4%)	16/847 (1.9%)	[1.1%, 2.8%]	21/847 (2.5%)	Aggregation; safety
Freshness only	3.10 s	41/847 (4.8%)	24/847 (2.8%)	[1.9%, 3.9%]	33/847 (3.9%)	Temporal; multi-hop
All clauses	2.89 s	98/847 (11.6%)	61/847 (7.2%)	5.7% to 9.0%	83/847 (9.8%)	Safety; temporal; aggregation

Table 17: Route-certificate semantics in the current deployment.

Clause	Certificate input	Excluded route	Work retained by eligibility	Main evidence
Provenance	Evidence-renderer capability and required granularity	Template that cannot expose edge- or passage-level support	Citation-bearing evidence assembly	Tables 2, 7, and 8
Temporal	Request window and bound temporal predicates	Route without in-window execution or scoped fallback	Time filtering and scoped response construction	Tables 16 and 19
Conflict	Covered conflict groups and traversal policy	Route that can silently omit a covered contradictory branch	Multi-evidence traversal and synthesis	Tables 20 and 24
Freshness	Cache dependencies and current watermarks	Reuse whose dependency changed after materialization	Update-aware recomputation	Tables 11 and 10

F ROUTE-LEDGER SUPPORT FOR ELIGIBILITY ANALYSIS

All summaries use the same 847-query benchmark, fixed graph snapshot, prompt/version bundle, instantiated candidates, estimator, and candidate TTFT measurements. Applying or relaxing eligibility changes the compared set while leaving those records fixed. Table 18 shows how one row supports the distinct oracle and deployed-selector quantities.

Table 18: Fields required to reconstruct the paired eligibility estimands.

Field	Recorded value	Derived quantity
Query and state IDs	Stable query, graph snapshot, cache state, and request-order identifiers	Pairing across eligibility conditions
Instantiated route ID	Family, template, parameters, retrieval configuration, and cache object	Exact selector identity and deterministic ties
Eligibility record	Boolean membership plus non-exclusive clause reason codes	Eligible set $\mathcal{A}$ and clause analyses
Measured and estimated TTFT	$T(P)$ and $\widehat{T}(P)$ for each route	Actual-latency oracles versus online selectors
Oracle records	$P^{\mathrm{orc}}_{\mathcal{A}}$, $P^{\mathrm{orc}}_{\mathcal{C}}$, and their fixed buckets	D_T, faster opportunity F, and bucket exposure B
Selector records	$\widehat{P}_{\mathrm{on}}$ and $\widehat{P}_{\mathrm{off}}$	Estimated-selector change S
Analysis strata	Query family and mixed-source/public-only indicator	Family and source-restricted summaries

Table 18 makes the previously conflated quantities reconstructible from one query-level record. In particular, B is computed from actual-oracle buckets and S from estimated-selector identities, so their counts need not be ordered.

Table 19: Query-family breakdown for same-stack eligibility sensitivity. Shares use the fixed denominators in Table 1.

Query family	Faster ineligible opportunity	Selector change	Dominant clause
Entity lookup	12/187 (6.4%)	6/187 (3.2%)	Missing surfaced provenance on otherwise direct retrieval paths.
Multi-hop	25/241 (10.4%)	14/241 (5.8%)	Provenance and freshness interact when intermediate summaries are reused.
Aggregation	22/168 (13.1%)	15/168 (8.9%)	Conflict visibility matters when multiple partial evidence chains must be reconciled.
Safety	21/139 (15.1%)	14/139 (10.1%)	Provenance and conflict clauses dominate high-risk recommendation queries.
Temporal	18/112 (16.1%)	12/112 (10.7%)	Temporal scope and freshness dominate update-sensitive evidence requests.
95% bootstrap CI over query IDs	9.6% to 13.9%	5.7% to 9.0%	Both effects remain visible under query resampling.

Entity lookup is least affected because its evidence chains and route space are compact. Safety, temporal, and aggregation questions show larger effects because they jointly stress provenance visibility, contradiction handling, and time scope. The entity-lookup floor and safety/temporal peak make eligibility sensitivity a structured workload property rather than a uniform constant.

Table 20: Route archetypes behind eligibility sensitivity.

Query family	Faster route if eligibility is relaxed	Violated clauses	Latency consequence
Safety	A graph-first path returns a concise recommendation from a sparse summary and omits the citation-bearing edge bundle that exposes contraindications.	Provenance + conflict	The eligible route spends additional time surfacing the contradictory evidence packet.
Temporal	A cached summary assembled before the latest update answers a time-scoped query without rechecking the requested window.	Temporal + freshness	The ineligible route skips update-aware recomputation.
Aggregation	A hybrid route collapses several evidence chains and silently drops a discordant branch.	Conflict + provenance	The eligible route retains the branch and supporting citations, which can place its oracle in a slower bucket.

Table 20 connects aggregate counts to recurring route mechanisms and the work omitted by an ineligible shortcut.

G Portable Public-Only Slice

We isolate source composition with a public-only workload containing 214 benchmark questions whose evidence packets are satisfied by the 127K public-document subset and corresponding public graph edges. The eligibility clauses, prompt bundle, candidate plan families, and hardware envelope remain fixed while the EMR-derived evidence slice is removed. This within-deployment intervention complements the separately constructed public-update workload under a source-restricted route population.

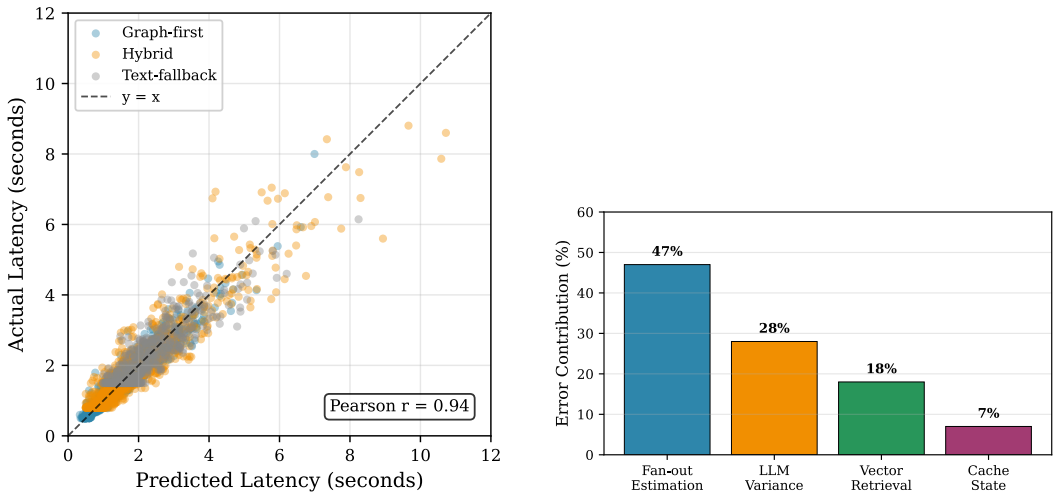
Table 21: Eligibility sensitivity on the portable public-only slice.

Metric	Public-only slice summary
Queries in slice	214.
Eligible actual-oracle P95	3.61 s.
Unrestricted actual-oracle P95	3.18 s.
Faster ineligible opportunity	19/214 (8.9%).
Estimated-selector change	11/214 (5.1%).
Dominant clauses	Temporal scope and surfaced provenance.

The public-only slice remains eligibility-sensitive: relaxation lowers the actual-oracle P95 by 0.43 s and changes 11 estimated-selector decisions. Its lower absolute rates provide a heterogeneous within-deployment check independent of EMR-derived evidence.

H Additional Planner Analyses

Prediction error concentrates in high-fan-out and statistics-stale cases rather than spreading uniformly across queries. This pattern explains how the planner attains 94.2% oracle match while retaining identifiable error modes.



(a) Predicted versus actual latency.

(b) Prediction-error decomposition.

Figure 1: Planner-fit diagnostics for the latency estimator used by AdmiTrace-Q.

For Table 9, the oracle is the fastest eligible plan family among graph-first, hybrid, and text-fallback under the same graph snapshot and cache state. Oracle match counts ties, and regret averages the latency gap to that oracle over all 847 queries, assigning zero to tie-or-match cases. Planner misses concentrate in underestimated fan-out for newly popular entities, overestimated vector-retrieval utility for rare diseases, and temporal-selectivity errors.

Table 22 quantifies the error scale by plan family. The 203 ms overall MAE is small relative to the multi-second route gaps in Table 6, supporting ranking adequacy for winner selection.

Table 22: Latency-estimation accuracy across the 847-query benchmark.

Metric	Graph-first	Hybrid	Text-fallback	Overall
MAE (ms)	187	241	176	203
RMSE (ms)	312	398	295	347
MAPE (%)	14.2	18.7	13.5	15.8
P95 error (ms)	524	687	512	583
Max error (ms)	1,847	2,312	1,932	2,312

Hybrid plans are hardest because they inherit variance from retrieval and graph expansion. Their larger error remains compatible with the oracle-match result because route selection depends on ordering candidates at the scale of multi-second tail differences.

I Additional Cache and Service Analyses

The warm-cache breakdown separates control overhead from retrieval and synthesis. Eligibility screening plus plan selection and cache lookup consume 0.9% of median TTFT, while graph traversal and LLM summarization dominate. In the update-stress analysis, one auditable reuse event is an attempted subgraph-cache or derived-summary reuse checked against the current watermark. Across 1,968 events, monitoring records zero stale hits; the one-sided 95% Clopper–Pearson upper bound is 0.15%.

Table 23: Warm-cache latency breakdown for AdmiTrace-Q (P50 in milliseconds).

Component	P50 (ms)	% of total
Intent classification	18	1.3%
Entity linking	31	2.2%
Eligibility + plan selection	8	0.6%
Cache lookup	4	0.3%
Graph traversal	302	21.3%
Subgraph assembly	28	2.0%
LLM summarization	862	60.7%
Network/serialization	167	11.8%
Total	1,420	100%

Table 23 makes the warm-cache operating point explicit. LLM summarization remains the dominant latency term, while graph traversal is the largest non-LLM systems cost. Certificate screening and route ranking are recorded together in the 8 ms eligibility-and-plan-selection row. The selector can therefore matter materially through its downstream route choice while consuming a small share of TTFT itself. Read together with Tables 10 and 12, the breakdown locates the latency headroom in cache state, traversal, and summarization.

J Quality Breakdown and Failure Analysis

Recall errors in the provenance audit arise from missing derived edges for rare drug–disease pairs (7%), temporal filters excluding recent evidence (5%), and entity-linking failures (4%). Precision errors arise from overly broad traversals (5%) and conflict-group edges included when not requested (4%). These categories align with the planner and compiler error modes and identify evidence assembly as the primary target for further improvement. No sampled answer was judged to contain factually incorrect medical advice with serious safety implications.

Table 24: Analysis-unit crosswalk for the separate 200-pair citation-audit subsample.

Analysis view	Unit	Reported use
Citation-audit sample	200 blinded query–answer pairs	Stratified inspection of missing citations, citation mismatches, and unflagged conflicts.
Failure taxonomy	23 AdmiTrace-Q citation-error instances	Instance counts in Figure 2; one answer can contribute multiple error instances.
Multi-label interpretation	Error instance	Separates retrieval, eligibility, and generation causes without collapsing co-occurring failures.

The taxonomy counts 23 error instances and permits multiple labels per answer. Entity-linking failures account for 8 instances (35%), temporal-filter over-pruning for 6 (26%), conflict-group omission for 5 (22%), and hallucination despite provided evidence for 4 (17%). The distribution separates failures in retrieval, eligibility, and generation and places most observed errors before final answer synthesis.

The paired-system answer/provenance audit samples 500 questions from the 847-query benchmark and evaluates one output from each system, yielding 500 displays per system. All five board-certified specialists score every independently randomized display, which defines the five-rater unit used by Fleiss' κ. Before system identities are revealed, the specialists adjudicate a reference evidence packet for each question from the benchmark evidence state; precision, recall, and F1 compare each system's returned evidence units with that packet. A separate three-expert audit inspects 200 query–answer pairs. The four 1–5 dimensions use anchored endpoints: 1 indicates materially incorrect, incomplete, unsafe, or unactionable output; 3 indicates partially satisfactory output requiring substantive revision;

and 5 indicates a complete, supported, clinically appropriate response. Instructions require penalties for unsupported claims, omitted contradictions, temporal mismatch, and risky phrasing. Evaluators completed the study within salaried clinical-informatics quality-assurance duties and received no per-item incentive.

For the behavior comparison aligned with matched latency, we draw 240 benchmark queries from the same 847-query workload, stratified by the five query families in Table 1. Auditors see the original question, one blinded system answer, and the surfaced evidence returned by that system; AdmiTrace-Q and OfflineGraph-Tuned share the graph snapshot, eligibility denominator, and blinded rubric. The table reports mean answer-level scores, acceptable-rate percentages, surfaced-provenance precision/recall/F1, and paired differences with 95% intervals.

Table 25: Family stratification for the matched 240-query paired audit.

Query family	Sampled queries	Why this stratum is preserved
Entity lookup	53	Keeps direct evidence-lookup behavior visible instead of letting harder families dominate the packet.
Multi-hop	68	Preserves the route family with the largest graph-expansion heterogeneity.
Aggregation	48	Keeps conflict-handling and evidence-composition questions visible in the matched audit.
Safety	39	Preserves the family where unsupported or contradiction-hiding answers are most consequential.
Temporal	32	Preserves the family most exposed to time-scope and freshness failures.
Total	240	Provides the paired audit sample reported in Table 26.

The stratification preserves the five-family benchmark mix and aligns the behavior audit with the route population used for latency measurement.

Table 26: Aggregate paired blinded audit on 240 sampled benchmark queries. Paired differences are AdmiTrace-Q minus OfflineGraph-Tuned.

Metric	AdmiTrace-Q	OfflineGraph-Tuned	Paired delta (95% CI)
Correctness	4.69	4.63	+0.06 [-0.02, +0.14]
Completeness	4.16	4.03	+0.13 [+0.03, +0.23]
Safety	4.80	4.75	+0.05 [-0.01, +0.11]
Actionability	4.01	3.92	+0.09 [+0.00, +0.19]
Acceptable	225/240 (93.8%)	220/240 (91.7%)	+2.1 pp [-1.0, +5.4]
Evid. precision	0.90	0.86	+0.04 [+0.01, +0.07]
Evid. recall	0.83	0.80	+0.03 [+0.00, +0.06]
Evid. F1	0.86	0.83	+0.03 [+0.01, +0.06]

The matched audit records favorable point estimates for every answer and provenance metric. Completeness differs by +0.13 [0.03, 0.23], evidence precision by +0.04 [0.01, 0.07], and evidence F1 by +0.03 [0.01, 0.06]. These paired measurements connect the latency advantage to answer completeness and surfaced evidence.

Table 27: Inferential protocol for the matched 240-query paired audit.

Protocol component	Definition	Role in Table 26
Sampling frame	240 queries stratified by the five fixed families	Aligns the audit population with the 847-query latency benchmark.
Blinded unit	One question, one identity-masked answer, and that answer's surfaced evidence	Prevents system identity and side-by-side contrast from influencing scoring.
Pairing key	Common benchmark query ID across the two policies	Defines within-query deltas for answer and provenance metrics.
Uncertainty unit	Paired bootstrap over sampled query IDs	Produces the reported 95% intervals while preserving answer–evidence pairing.
Adjudication	Five specialists using the anchored four-dimension rubric	Converts item scores into means and acceptable-answer counts.
Compensation	Salaried quality-assurance duty; no per-item incentive	Removes speed-based or output-count incentives from the audit.

Table 27 fixes the matched sampling, blinding, pairing, bootstrap, adjudication, and compensation conventions used to produce Table 26.

Table 28: Blinded audit form used for human evaluation.

Template field	Visible content and decision role
Benchmark question	The original clinician-authored query, shown verbatim so the evaluator can judge scope, temporality, and requested action.
Blinded system answer	One system output with identity removed; evaluators judge correctness, completeness, safety, and actionability from this text.
Surfaced supporting evidence	The edge- or passage-level citations returned by that same system, shown together with the answer so unsupported claims and omitted contradictions can be penalized directly.
Score controls	Four required 1–5 radio groups with anchored labels at 1, 3, and 5, plus Acceptable / Not acceptable.
Error flags	Required checkboxes for unsupported claim, temporal mismatch, hidden conflict, unsafe recommendation, and other with free text.
Rubric reminder	Evaluators must penalize unsupported claims, temporal mismatch, contradiction suppression, and clinically risky phrasing even when the prose is fluent.

The form forces a decision on all four quality dimensions and acceptability before submission. Error flags connect the human judgment to the provenance, temporal, conflict, and safety analyses without revealing system identity.

Entity linking is the largest residual source, followed by temporal over-pruning and conflict omission. Most observed citation-error instances therefore arise during evidence assembly and eligibility-sensitive retrieval before final answer wording. Together, the anchored form, randomized single-

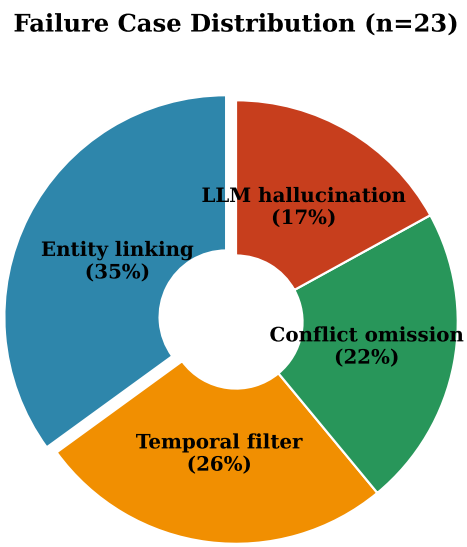


Figure 2: Instance-level taxonomy of the 23 AdmiTrace-Q citation-error cases described in Appendix J.

system presentation, five ratings per display, pre-adjudicated reference packet, and paired query bootstrap align the evaluation interface and uncertainty unit with the benchmark questions.

K SOFTWARE AND DATA GOVERNANCE

The de-identified private EMR component and retrospective expert audit were covered by institutional review procedures. The audit evaluates system outputs rather than clinical interventions. No part of the study returns model output to patients or substitutes for clinical decision-making. The intended use is decision support for analysts or clinicians who can inspect cited evidence, and human review remains required when evidence is sparse, conflicting, or recently updated.

The implementation uses Neo4j Enterprise, Redis, Faiss, Elasticsearch, LangChain, Azure OpenAI GPT-4-Turbo, ColBERT, and BioBERT/ClinicalBERT-family encoders. Commercial services follow their product terms, open-source components retain their upstream licenses, terminology resources follow jurisdiction-specific agreements, public documents follow source-specific terms, and the de-identified EMR slice remains governed by institutional data-use controls.