Supplementary Information for Every-successful-replay route admission changes first-token latency rankings in clinical question answering

Rui Li*, Jason Zhao, Shuang Cao, Alexandre Duprey, and Ruihua Liu
Department of AI, Hill Research, Princeton, NJ 08540, USA
*Correspondence: rui.li@hillresearch.ai

S1 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 [1–3]. Template selection chooses among 47 parameterized Cypher queries [4]. On the pre-benchmark 2,480-question development evaluation, 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.

The study-specific TextRAG comparator 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) [5], followed by graph traversal and summarization. Heuristic-GF is MedRouteGuard with a fixed graph-first plan. OfflineGraph-Tuned uses the same evidence graph, the same 847-query benchmark, the same Azure worker classes, the same Redis cache budget, the same admission contract, and the same latency protocol as MedRouteGuard; admitted-answer counts are reported separately. It keeps a fixed graph-first serving policy with hand-tuned Neo4j query hints and ColBERT-based entity linking. OfflineGraph-Tuned is therefore the matched fixed graph-serving comparator for the Q2 latency claim; TextRAG supports the specified Q1 output audit, while GraphRAG-OD is descriptive. Table S14 supplies the paired quality bridge for Q2 by auditing MedRouteGuard and OfflineGraph-Tuned on the same sampled questions and admission contract.

Table S1 maps the maintained-service experiment blocks with retained accounting to fixed worker pools, wall-clock duration, and aggregate compute.

The first row anchors online serving results to the exact worker classes used in the main paper. The cache and update analyses reused that serving stack rather than a separate simulator. Reviewer time is excluded from system compute because it measures human evaluation rather than inference. The total is the sum of the three non-overlapping measured blocks.

Table S1: Retained compute accounting for the maintained-service experiment blocks. CPU node-hours include the five graph-store and two query-service workers; GPU hours refer to the NC24ads A100 v4 worker.

Experiment block	Primary workers / measured unit	Wall-clock and aggregate compute
Matched latency, throughput, and planner analyses	Full online stack: 5× E32ads v5 Neo4j workers, 2× E16ads v5 query-service workers, 1× NC24ads A100 v4 vector-index worker, and Redis; measured on the 847-query benchmark.	18.4 h; 128.8 CPU node-hours; 18.4 A100 hours.
Cache, warm/cold, and update analyses	Same serving stack plus injected update replays on the maintained graph; measured on the no-update run and the 5/20/100 upd/s stress settings.	11.7 h; 81.9 CPU node-hours; 11.7 A100 hours.
Retrospective answer and citation audits	Blinded answer and citation packets exported from the same benchmark; system inference used the same serving stack, while reviewer time is outside system compute.	6.3 h; 44.1 CPU node-hours; 6.3 A100 hours.
Total retained compute for these blocks	Three experiment blocks above; excludes clinician review time and the separately executed medication-safety workload.	36.4 h; 254.8 CPU node-hours; 36.4 A100 hours.

S2 Operational Support for Plan Selection, Cache Reuse, and Ablations

Table S2: Plan-selection quality under the admissible oracle on the 847-query benchmark. Higher oracle match and lower regret are better.

Strategy	Oracle Match	P95 (s)	Regret (s)
Cost-based (MedRouteGuard)	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 admissible plan family measured under the same graph snapshot and cache state, and regret is averaged over all 847 questions. The table shows that this workload requires routing inside the admitted set: the fixed graph-first policy reaches 67.4% oracle match and 5.49 s planner-diagnostic P95, compared with 94.2% and 3.24 s for the cost-based planner. When the planner misses, the penalty appears in tail delay. The cost-based planner therefore improves search within a fixed admission denominator, while Section S6 reports additional fit diagnostics.

Table S3: 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 S3 shows that reuse is operationally material: priming raises hit rate to 67.3% and cuts deployed MedRouteGuard first-generated-token P95 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 anchors the Q2 deployed-system operating point and the online planner and ablation diagnostics. Q3's gate-on retrospective-oracle P95 happens also to be 3.24 s, but it remains a separate fixed-candidate estimand. The table therefore turns cache state from a background implementation detail into a controlled part of the benchmark contract.

Table S4: 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. P95 is the deployed-stack first-generated-token diagnostic. 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 S4 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 within the archived replays. The stress test documents conservative invalidation under churn and quantifies its increasing latency cost.

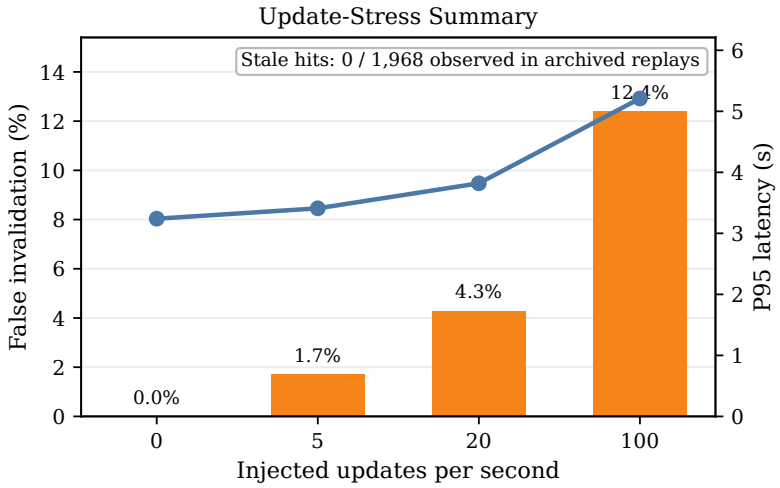

Figure S1: Update-stress summary backed by Table S4: false invalidation rises with update rate, while stale-hit monitoring records 0/1,968 observed stale-hit events in the archived replays.

Figure S1 turns that zero-count table into the intended trade-off view. Under heavier churn, the system gives up hit rate and accepts more false invalidations rather than serve stale state. The x-axis

therefore tracks the cost of refusing stale reuse, not the benefit of a permissive cache. The intended asymmetry is simple: the system gives up speed before it gives up freshness. That asymmetry makes the freshness clause an observed systems cost within the benchmark rather than an untimed metadata check.

Table S5: Runtime configuration diagnostics at the warm-cache, no-update operating point. Lower latency and higher evidence precision are better; rows compare complete configurations using the archived aggregate operating-point summaries.

Configuration	P50 (s)	P95 (s)	Evid. Prec.
Full MedRouteGuard	1.42	3.24	0.91
− Cost-based planner	1.68	5.49	0.91
− Subgraph cache	1.87	6.78	0.91
− Derived summaries	1.58	4.12	0.89
− Fan-out caps	1.52	8.73	0.91
− All caching	1.91	7.21	0.91

At this archived operating point, every shown removal had a higher P95 point estimate than the full configuration. The Full MedRouteGuard row is the warm-cache, no-update point reported in the main-text deployed-system latency table and Table S2; the −Cost-based planner row is the internal fixed-policy diagnostic rather than OfflineGraph-Tuned. The only lower evidence-precision point estimate was observed after removal of derived summaries. These configuration diagnostics show how route selection and dependency-aware reuse operate together; the main-text membership intervention remains the controlled analysis of admission.

S3 Clause-Robustness of the Boundary Audit

We tested clause-level robustness by recomputing the retrospective oracle on the same fixed 847-question executions while relaxing one clause label at a time. The graph snapshot, prompt bundle, Azure worker classes, candidate plan families, cache budget, gate computations, and latency estimator remained fixed. The full-gate baseline is the admitted benchmark reported in the main-text Results; the comparison variants ignore provenance only, temporal only, conflict only, freshness only, or all four labels. For each variant we report apparent P95 latency, excluded-candidate speed events, median-route winner changes, candidate-pool bucket-set changes, dominant affected question families, and 95% bootstrap intervals over question IDs.

Table S6: Clause-robustness audit for the same-stack boundary effect. Every row uses the fixed 847-question benchmark, graph snapshot, hardware envelope, prompt bundle, candidate plan families, and latency estimator; only the named candidate-level admissibility label is relaxed. A qualifying fast replay comes from a candidate classified analysis-inadmissible; the replay itself is not assigned the named failed-clause tag.

Ignored label	Apparent P95	Excluded-candidate speed event	Winner flip	95% CI	Bucket-set change	Dominant affected families
Provenance only	3.02 s	62/847 (7.3%)	34/847 (4.0%)	[2.7%, 5.5%]	45/847 (5.3%)	Candidate-level provenance tags concentrate in safety and aggregation questions.
Temporal only	3.07 s	57/847 (6.7%)	31/847 (3.7%)	[2.4%, 5.1%]	42/847 (5.0%)	Candidate-level temporal tags concentrate in temporal and aggregation questions.
Conflict only	3.14 s	29/847 (3.4%)	16/847 (1.9%)	[1.1%, 2.8%]	21/847 (2.5%)	Candidate-level conflict tags concentrate in aggregation and safety questions.
Freshness only	3.10 s	41/847 (4.8%)	24/847 (2.8%)	[1.9%, 3.9%]	33/847 (3.9%)	Candidate-level freshness tags concentrate in temporal and multi-hop questions.
All clauses	2.89 s	98/847 (11.6%)	61/847 (7.2%)	5.7% to 9.0%	83/847 (9.8%)	Joint relaxation most affects safety, temporal, and aggregation questions.

Table S6 shows a graded boundary effect. Provenance and temporal relaxations produce the largest single-clause winner changes, freshness affects update-sensitive questions, and conflict-only relaxation also changes benchmark membership. Relaxing all four labels produces the largest effect, consistent with interacting clause families.

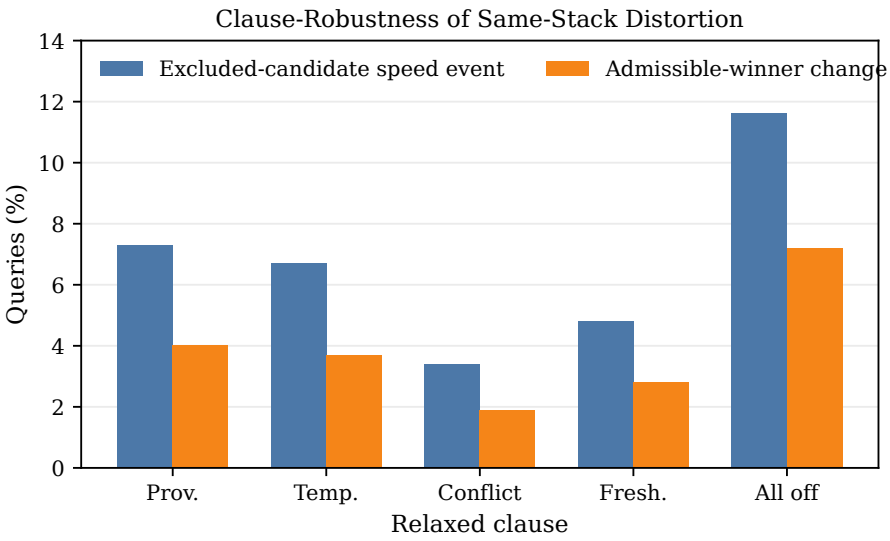

Figure S2: Clause-robustness view of the same-stack boundary audit. Bars show excluded-candidate speed events and median-latency winner changes for each candidate-level label relaxation on the 847-question benchmark. The figure does not assert that a qualifying replay failed the named clause.

Figure S2 makes the robustness pattern visible at a glance. Winner changes are largest when provenance or temporal checks are relaxed, but every single-clause relaxation remains above zero. The same benchmark-membership effect survives milder relaxations, and the clause families differ in magnitude rather than only in wording.

S4 Aggregate Support for the Boundary Audit

Because the main-text route-membership table is the direct audit of whether the timing boundary matters, this section reports the two auxiliary projections needed to interpret the aggregate: overlap among violated clauses and concentration by question family. Both use the same 847-question candidate pool, synchronized replay conditions, and question-level bootstrap described in Methods.

Table S7: Overlap structure among candidate-level failed-clause tags for excluded candidates associated with a speed event. Pair counts are question-level and non-exclusive because one candidate can carry more than two tags. Tags may originate from different replays and do not describe the clause state of the qualifying fast replay.

Violation pattern	Questions	Interpretation
Provenance + temporal	7	Associated excluded candidates carry both provenance and temporal tags across their measured-replay sets.
Provenance + freshness	6	Associated excluded candidates carry both provenance and freshness tags across their measured-replay sets.
Temporal + freshness	5	Associated excluded candidates carry both temporal and freshness tags across their measured-replay sets.
Provenance + conflict	4	Associated excluded candidates carry both provenance and conflict tags across their measured-replay sets.
Temporal + conflict	3	Associated excluded candidates carry both temporal and conflict tags across their measured-replay sets.
Conflict + freshness	2	Associated excluded candidates carry both conflict and freshness tags across their measured-replay sets.
Any two or more clauses	18/98 (18.4%)	Multi-clause failures are common enough that the aggregate distortion is not driven by one isolated admissibility clause.

Table S7 shows that the speed-event slice contains structured, overlapping candidate-level failures. Provenance and temporal overlap is the most common pair, followed by combinations involving freshness. The overlap structure supports the interpretation of benchmark membership as a multi-clause evidence boundary.

Table S8: Question-family breakdown for the same-stack distortion diagnostic. Shares use the fixed main-text cohort denominators. Clause descriptors are candidate-level tags and are not attributed to the replay producing a qualifying fast latency.

Question family	Excluded-candidate speed event	Median-route winner flip	Dominant relaxed clause
Entity lookup	12/187 (6.4%)	6/187 (3.2%)	Candidate-level provenance.
Multi-hop	25/241 (10.4%)	14/241 (5.8%)	Candidate-level provenance and freshness.
Aggregation	22/168 (13.1%)	15/168 (8.9%)	Candidate-level conflict.
Safety	21/139 (15.1%)	14/139 (10.1%)	Candidate-level provenance and conflict.
Temporal	18/112 (16.1%)	12/112 (10.7%)	Candidate-level temporal scope and freshness.
95% bootstrap CI over question IDs	9.6% to 13.9%	5.7% to 9.0%	Distortion remains visible after uncertainty is propagated over benchmark questions.

Table S8 shows where the same-stack distortion is concentrated. Entity-lookup questions are least affected, consistent with their shorter evidence chains and constrained route space. Safety, temporal, and aggregation questions are more fragile because they place simultaneous pressure on provenance visibility, contradiction handling, and scope control. The bootstrap rows propagate uncertainty over benchmark questions rather than presenting an unqualified pooled point estimate. The table therefore identifies clinically meaningful workload strata in which denominator distortion is most frequent.

Figure S3 makes the family concentration behind the main-text route-membership result visible at a glance. Winner flips are highest in safety (14/139 (10.1%)), temporal (12/112 (10.7%)), and aggregation (15/168 (8.9%)) questions, where provenance, contradiction handling, and time scope most directly affect clinical usefulness. The pattern concentrates membership sensitivity in evidence-intensive question families. Entity lookup has the lowest rate, showing that membership sensitivity increases with evidence-composition demands.

S5 Portable Public-Only Slice

To test whether the timing boundary remains behaviorally active outside the private-backed slice, we define a public-only workload containing 214 benchmark questions whose evidence packets can be satisfied from the 127K public-document subset and the corresponding public graph edges without EMR-derived evidence. The admissibility clauses, prompt bundle, candidate plan families, and hardware envelope are unchanged; only the private evidence slice is removed. This analysis isolates portability of the boundary effect, while the matched Q2 latency comparison remains anchored to the full workload.

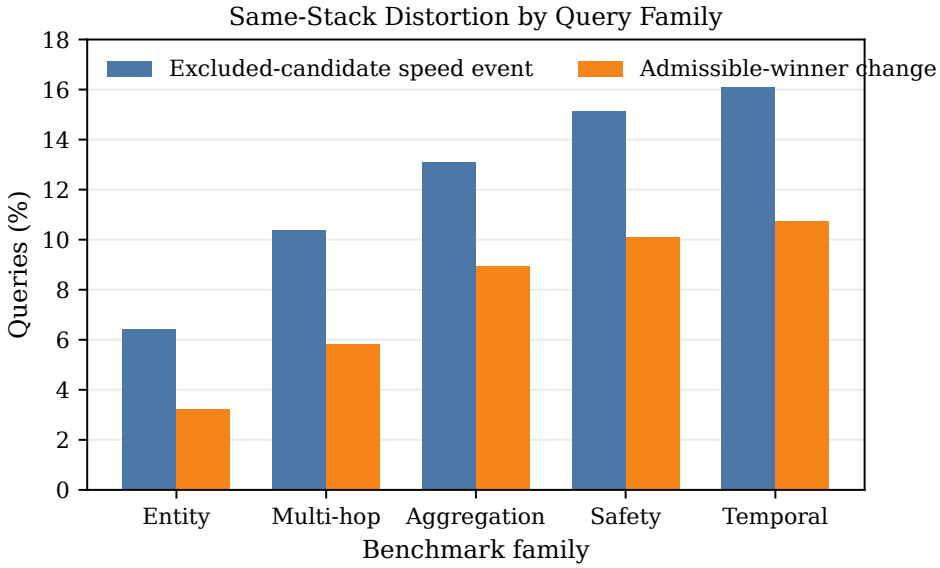

Figure S3: Distribution view of the same-stack membership diagnostic. Bars show excluded-candidate speed events and median-latency winner changes by question family. A qualifying replay inherits only the candidate-pool classification, not the candidate's failed-clause tags.

Table S9: Portable public-only slice for the gate-on versus gate-off distortion diagnostic.

Metric	Public-only slice summary
Questions in slice	214.
Admissible P95 latency	3.61 s.
Gate-off P95 latency	3.18 s.
Excluded-candidate speed events	19/214 (8.9%).
Median-latency route-winner changes	11/214 (5.1%).
Dominant relaxed labels	Candidate-level temporal-scope and provenance tags are most frequent among public-only winner changes.

The public-only slice remains boundary-sensitive: the gate-off analysis lowers apparent tail latency from 3.61 s to 3.18 s and changes 11/214 (5.1%) median-route winners. Its direction agrees with the full benchmark and separates the membership effect from the private EMR-backed evidence component. This slice evaluates portability of the route-membership diagnostic while the main Q2 comparison remains anchored to the matched full-workload baseline.

S6 Additional Planner Analyses

Table S10 supplies the planner-fit diagnostics. Prediction error concentrates in a small set of high-fan-out or statistics-stale cases rather than being uniformly diffuse, explaining how the planner attains high oracle match. These diagnostics characterize the evaluated deployment and statistics-refresh policy.

For Table S2, the oracle is the fastest admissible plan family among graph-first, hybrid, and text-fallback under the same graph snapshot and cache state used for the evaluated query. Oracle match counts ties. Reported regret is the latency gap to that oracle averaged over all 847 queries, so the 94.2% tie-or-match cases contribute zero regret. Planner misses are dominated by underestimated fan-out for newly popular entities, overestimated vector-retrieval utility for rare diseases, and temporal-selectivity errors.

Table S10 makes the error scale explicit for each plan family. This matters because the main paper's regret claim depends on ranking adequacy, not on exact runtime simulation. The overall column is therefore the key summary for the main claim, while the family-specific columns explain where the remaining misses come from. Read against the multi-second main-text latency gaps, the table shows why sub-second estimation noise can still be adequate for plan ranking.

Table S10: 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 both retrieval and graph expansion. Even so, the overall error scale remains small relative to the multi-second tails in the main-text latency results. The table therefore supports ranking adequacy inside the admissible set. The family breakdown also explains why a few high-fan-out misses can coexist with a strong oracle-match rate in Table S2. In other words, the estimator only needs to be ranking-adequate at the scale of route choice, not numerically exact on every query.

S7 Additional Cache and Service Analyses

The main paper includes the service-level latency and throughput figure; Section S2 reports the cold/warm and update-stress tables. This section adds the component-level warm-cache breakdown. The planner's control overhead remains small relative to retrieval and summarization even in a reuse-active regime, although these measurements remain specific to one deployment stack. For the update-stress setting referenced by Table S4, one auditable reuse event denotes one attempted

subgraph-cache or derived-summary reuse checked against the current watermark. Aggregated across the archived replays, the logs record 0/1,968 observed stale-hit events, so the 0.0% cells should be read as rounded zero counts. 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 within the archived replays.

Table S11: Archived warm-cache component P50 values for MedRouteGuard in milliseconds. Displayed percentages are rounded and sum to 100.2%.

Component	P50 (ms)	% of total
Intent classification	18	1.3%
Entity linking	31	2.2%
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 S11 makes the warm-cache operating point explicit. LLM summarization remains the dominant reported component, while graph traversal is the largest non-LLM systems component. Planner and cache bookkeeping are small in the archived component summary; admission checks were not retained as a separate component row. The table also explains why the planner can matter materially even though its own runtime is small: a routing decision influences a much larger downstream latency budget. Read together with Tables S3 and S5, the breakdown locates most reported latency in cache state, traversal, and summarization rather than plan-selection bookkeeping.

S8 Quality Breakdown and Citation-Error Mechanisms

Among the 500 MedRouteGuard output packets in the answer/provenance audit, reviewer-coded recall-error categories occurred for missing derived edges on rare drug–disease pairs (7% of packets), temporal filters excluding recent evidence (5%), and entity-linking failures (4%); categories were non-exclusive. Packet-level precision-error categories were overly broad traversals (5%) and conflict-group edges included when not requested (4%). These rates are distinct from the 23 citation-error instances in the separate mechanism taxonomy below. They localize residual failures to specific retrieval, temporal-filtering, linking, and conflict-group mechanisms. Within this sample, reviewers identified no factually incorrect medical advice with serious safety implications.

Table S12: Unit crosswalk for the separate 200-question blinded citation-mechanism audit.

Analysis view	Unit	Interpretation
Mechanism-audit sample	200 matched questions; 400 blinded query–answer items	Three citation experts inspected both system outputs to preserve blinded review context.
Reviewer exposure	1,200 output–expert assessments	Each expert assessed one blinded output for missing citations, citation mismatches, and unflagged conflicts.
Prespecified reported endpoint	23 MedRouteGuard citation-error instances	The endpoint is a descriptive mechanism taxonomy, not a second comparative provenance effect; one output can contribute multiple instances.

Table S12 separates matched-question, query–answer-item, output–expert, and error-instance denominators. The comparative provenance effect comes from the 500-pair audit in the main paper; this separate endpoint localizes MedRouteGuard error mechanisms. One audited output can contribute multiple instances when it simultaneously misses one citation, mismatches another, and fails to expose a conflicting edge. Within the 23-instance taxonomy, the most common causes are entity-linking failures (8 cases, 35%), temporal-filter over-pruning (6, 26%), conflict-group omission (5, 22%), and generation error despite provided evidence (4, 17%). This breakdown distinguishes retrieval, admission, and generation failures without treating instance percentages as question-level rates.

The 500-pair answer/provenance audit and separate 200-question citation-mechanism audit were sampled from the 847-question benchmark. Each blinded item showed one question, one system answer, and that answer's returned evidence; the mechanism audit therefore comprised 400 query–answer items and 1,200 output–expert assessments. Five board-certified specialists scored correctness, completeness, safety, and actionability; three citation experts classified support failures. Reviewers penalized unsupported claims, omitted contradictions, temporal mismatches, and clinically risky phrasing even when the prose was fluent. Comparative provenance counts, score distributions, reviewer-aware inference, agreement, and the missing-rating rule come from the 500-pair audit reported in Methods and Box 1.

For the matched Q2 bridge, we draw 240 benchmark queries from the same 847-query workload, stratified by the five main-text query families. Auditors see the original question, one blinded system answer, and the surfaced evidence returned by that system; MedRouteGuard and OfflineGraph-Tuned share the graph snapshot, question sample, admissibility contract, and blinded rubric, and only the serving policy changes. The table reports mean answer-level scores, acceptable-rate percentages, surfaced-provenance precision/recall/F1, and paired deltas with 95% intervals.

Table S13: Family stratification for the matched Q2 paired audit. The sample preserves the fixed five-family workload mix while holding the benchmark source, graph snapshot, question denominator, and admissibility contract fixed.

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 S14.

Table S13 shows that the Q2 audit preserves the benchmark mix across entity lookup, multi-hop, aggregation, safety, and temporal behavior. That balance aligns the paired quality bridge with the same question population and admissibility contract used for the latency claim.

Table S14: Aggregate paired blinded audit for the matched Q2 comparator on 240 sampled benchmark queries. Paired deltas are MedRouteGuard minus OfflineGraph-Tuned; positive is better.

Metric	MedRouteGuard	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 (%)	93.8	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]

Table S14 provides the aggregate paired quality bridge for the Q2 comparator. Acceptability was 93.8% for MedRouteGuard and 91.7% for OfflineGraph-Tuned, with a paired difference of +2.1 percentage points [−1.0, +5.4]; completeness and surfaced-provenance estimates favored MedRouteGuard on the matched subsample. The audit links the latency comparison to the same question population, evidence contract, and blinded quality rubric.

Table S15: Analysis specification for the matched Q2 paired audit.

Analysis component	Specification	Role in interpreting Table S14
Sampling	Unique benchmark question IDs and fixed five-family labels	Establishes that the audit is drawn from the same 847-question population as Q2.
Blinded item	One question, one system answer, and that answer's surfaced evidence	Fixes the judged unit and prevents side-by-side system cues.
Assignment	Two initial ratings per output with independent adjudication of acceptability or safety disagreements	Separates reviewer replication from the paired question-level analysis.
Observed rating completeness	960/960 primary ratings complete	Confirms that no sampled output was removed for a missing primary rating.

Table S15 fixes the Q2 audit population, sampled unit, blinded presentation, reviewer assignment, and complete primary-rating denominator before inference is specified.

Table S16: Endpoint and inference specification for the matched Q2 paired audit.

Analysis component	Specification	Role in interpreting Table S14
Adjudication	37/480 outputs (7.7%)	Reports the output-level frequency of independent acceptability or safety disagreement resolution.
Acceptability	Consensus score of 4 or 5 on every dimension and no serious safety flag	Makes the binary endpoint operational and clinically interpretable.
Paired inference	Question-level paired deltas with 95% intervals	Preserves pairing between systems rather than treating outputs as independent.
Missing outcomes	No imputation; missing items remain in the reported denominator	Prevents selective removal of difficult or failed outputs.

Tables S15 and S16 fix the audit unit, blinding, assignment, endpoint, pairing, and missing-outcome rule. These design choices attach the quality analysis to the same question population and admissibility contract as Q2. The resulting intervals address compatibility with the observed latency difference.

Table S17: Blinded audit-item template used in human evaluation.

Template field	Visible content and decision role
Benchmark question	The original clinician-authored query, shown verbatim so the reviewer can judge scope, temporality, and requested action.
Blinded system answer	One system output with system identity removed; reviewers 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.
Rubric reminder	Reviewers are instructed to penalize unsupported claims, temporal mismatch, contradiction suppression, and clinically risky phrasing even when the prose is fluent.

Table S17 makes the human audit concrete at the level of one reviewed output. The benchmark question anchors scope, the blinded answer defines the judged output, and the supporting evidence permits direct assessment of claim support. The rubric reminder identifies clinically relevant errors that must be penalized even when the prose is fluent.

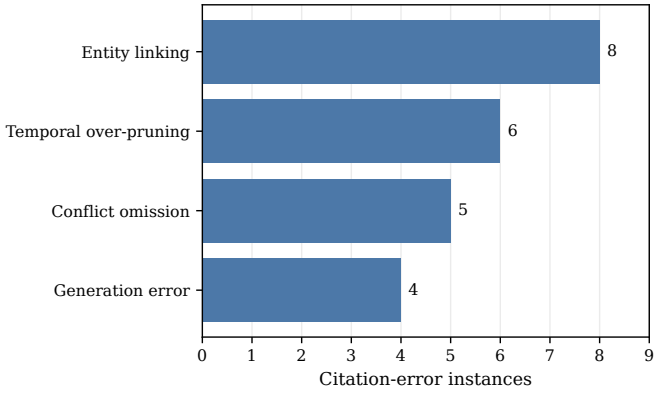

Figure S4: Instance-level taxonomy of the 23 MedRouteGuard citation-error cases described in this section.

Figure S4 presents the 23-case failure taxonomy as a distribution. Entity linking is the largest single source, followed by temporal over-pruning and conflict omission. Most residual audited failures therefore arise before final answer wording rather than from free-form generation alone. The instance-level composition complements the packet-level prevalence estimates reported above.

Tables S17 and S18 summarize the item interface, reviewer instructions, assignments, and analysis units.

Table S18: Protocol summary for the 500-pair answer/provenance audit and separate 200-question citation-mechanism audit.

Protocol element	Specification
Reviewed item content	One benchmark question, one blinded system answer, and the supporting cited evidence returned by that system.
Item order and blinding	Single-system packets with system identity removed; reviewers inspect one answer at a time rather than a side-by-side ballot.
Answer-quality rubric	Four 1–5 Likert dimensions: correctness, completeness, safety, and actionability.
Citation-mechanism focus	Descriptive MedRouteGuard error instances—missing citations, citation mismatches, and unflagged conflicts—in a separate stratified sample of 200 matched questions.
Evaluator pools	Five board-certified specialists for answer quality and three experts for citation-mechanism classification.
Reviewer assignment	Every answer output received three independent ratings; 3,000 ratings total and 600 per reviewer; every mechanism-audit output was rated by all three citation experts.
Instructions	Penalize unsupported claims, omitted contradictions, temporal mismatches, and clinically risky phrasing even when the prose is fluent; escalate any serious safety disagreement.
Interface behavior	Present one answer at a time, require all rubric fields before submission, and conceal the paired system output until the item is locked.
Compensation and independence	Reviewers were uncompensated beyond normal salary, did not author sampled questions, and had no role in system implementation.
Paired analysis	The unique question was the unit for acceptable-rate differences; reviewer and question random intercepts were used for ordinal outcomes.

Together, the tables define what each reviewer saw, how ratings were assigned, which errors were penalized, and how repeated ratings were analyzed. This makes the human-evaluation design interpretable alongside the raw counts and uncertainty in Box 1.

S9 Common-Packet Robustness of the Medication-Safety Review

The independent review copied the same frozen answer-and-evidence packet into both blinded conditions whenever the gate-on and gate-off retrospective oracles selected the same route. This occurred for 166/180 pairs. Independent presentation produced balanced condition-only disagreements on these identical objects: 4 versus 4 for material error and 2 versus 2 for unsafe omission. Constraining each identical packet to one common outcome therefore removes repeated-rating variation without changing either paired risk difference.

Table S19: Exact reconstruction of the common-packet robustness analysis. Condition-only discordances are gate-on-only versus gate-off-only. Subtracting identical-packet discordances from the full discordance table leaves the 14 route- and output-discordant pairs. Bracketed effect intervals are 95% percentile intervals from the enumerated paired-bootstrap distribution over those 14 questions.

Endpoint	Full condition totals	Full condition-only discordance	Identical-packet discordance	Output-discordant sensitivity
Material evidence-validity error	7/180 versus 18/180	5 versus 16	4 versus 4	1 versus 12; active-pair reduction 78.6 pp [42.9, 100.0]; exact $P = 0.0034$.
Unsafe omission	4/180 versus 13/180	3 versus 12	2 versus 2	1 versus 10; active-pair reduction 64.3 pp [28.6, 92.9]; exact $P = 0.0117$.

Within the 14 output-discordant pairs, condition totals were 2 versus 13 for material error and 1 versus 10 for unsafe omission. For material error, one pair was positive in both conditions, one was gate-on-only, 12 were gate-off-only, and none was negative in both; for unsafe omission, one was gate-on-only, ten were gate-off-only, and three were negative in both. The resulting gate-off minus gate-on risk differences were 78.6 percentage points [42.9, 100.0] and 64.3 points [28.6, 92.9], respectively. The paired error differences are therefore generated by changed outputs, while the balanced disagreements on repeated packets contribute uncertainty but no net condition effect.

Table S20: Descriptive request-to-first-generated-token latency for the non-matched implementations. These rows provide implementation context and are not part of the matched Q2 comparison in the main text.

Implementation	P50 (s)	P95 (s)	P99 (s)	Study role
TextRAG	2.14	4.87 [4.61, 5.14]	7.21	Study-specific text-only comparator for Q1 output auditing.
GraphRAG-OD	7.83	18.42 [17.21, 19.68]	27.61	Contextual query-time graph builder.

S10 Supplementary governance and external assets

The work uses public biomedical sources, commercial infrastructure, and existing software systems including Neo4j, Redis, Faiss, Elasticsearch, LangChain, ColBERT, BioBERT/ClinicalBERT-family encoders, and the Microsoft GraphRAG codebase. The principal named software and service terms in the reported stack are Neo4j Enterprise 5.15.0 under Neo4j's self-managed commercial software agreement, Redis 7.2.3 under BSD-3-Clause, Faiss 1.7.4 under MIT, Elasticsearch 8.11.3 under Elastic License 2.0, LangChain under MIT, and Azure OpenAI GPT-4-Turbo under the Microsoft Azure product terms. TextRAG and OfflineGraph-Tuned use ColBERT reranking or linking, and the compiler initializes dense retrieval from the BioBERT/ClinicalBERT family. Table S21 documents the principal software, model-family, vocabulary, and data-source classes used to interpret and reproduce the released aggregate analyses. The primary foreseeable harm is overreliance on evidence that is incomplete, stale, or internally conflicting.

Table S21 converts that compact narrative into a concise manifest organized by experimental role and release status.

Table S21: Principal software and model assets used in the reported experiments.

Asset	Role	Terms / release status
Neo4j Enterprise 5.15.0	Graph store	Neo4j self-managed commercial software agreement.
Redis 7.2.3	Online caching	BSD-3-Clause.
Faiss 1.7.4	Vector retrieval	MIT.
Elasticsearch 8.11.3	TextRAG BM25 retrieval	Elastic License 2.0.
LangChain	Orchestration utilities	MIT.
Azure OpenAI GPT-4-Turbo (gpt-4-1106-preview)	Evidence summarization	Microsoft Azure product terms.
ColBERT reranker / linker	TextRAG reranking and OfflineGraph-Tuned entity-linking support	Upstream code and checkpoint terms apply; weights are not redistributed.
BioBERT / ClinicalBERT-family encoders	Dense biomedical retrieval and compiler initialization	Upstream checkpoint terms apply; model weights are not redistributed.
Microsoft GraphRAG implementation (v0.3.2)	GraphRAG-OD baseline	Upstream project terms apply.

Table S22 records the terminology and data-source classes whose redistribution conditions differ from software licenses.

Table S22: Terminology and data-source assets used in the reported experiments.

Asset	Role	Terms / release status
UMLS / SNOMED CT / RxNorm vocabularies	Normalization and entity linking	Upstream terminology licenses and jurisdiction-specific terms apply; terminology content is not redistributed.
Public biomedical document slice	Public evidence source class for the 127K public documents	Upstream publisher or government terms vary by source; the document corpus is not redistributed.
De-identified EMR slice	Private deployment evidence	Subject to institutional review and data-use controls; not redistributed.

Tables S21 and S22 separate software, services, vocabularies, public documents, and private data because those assets carry different licensing and redistribution constraints. Together they document the principal asset classes needed to interpret the evidence boundary.

References

[1] 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/`.

[2] 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`.

[3] 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*, pages 72–78, Minneapolis, Minnesota, USA, 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-1909. URL `https://aclanthology.org/W19-1909/`.

[4] 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, editors, *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*, pages 1433–1445. ACM, 2018. doi: 10.1145/3183713.3190657. URL `https://homepages.inf.ed.ac.uk/libkin/papers/sigmod18.pdf`.

[5] Microsoft. GraphRAG: Retrieval-augmented generation with graphs. `https://github.com/microsoft/graphrag`, 2025. Version v0.3.2. Accessed: 2025-12-26.