Supplementary Information for AionRAG: Time-Correct Retrieval-Augmented Generation under Knowledge Drift

Shuang Cao[1], Rui Li[1,*], Ruihua Liu[1], Alexandre Duprey[1], Angel Dong[1]

[1]Hill Research

[*]Correspondence to: Rui Li, Hill Research, rui.li@hillresearch.ai

Supplementary Information

This document provides extended methods, dataset details, additional results, and reproducibility information for "AionRAG: Time-Correct Retrieval-Augmented Generation under Knowledge Drift."

Contents

1 Dataset Construction and Protocols

1.1 Controlled Stress Tests

StreamingQA

We adapt the StreamingQA benchmark[1] to our temporal setting. The dataset contains 42,180 queries spanning news events from 2018–2024. We augment each query with explicit timestamp metadata and construct update sequences where answers change over time. Ground-truth timestamps are derived from the original article publication dates. We use a 70/15/15 train/calibration/test split, with temporal non-overlap between splits.

ConflictQA

We construct ConflictQA (n=45,720) to stress-test conflict resolution. For each query, we inject 2–5 temporally conflicting passages: passages that are semantically similar but provide contradictory answers due to temporal misalignment. Conflict severity is categorized as: None (0 conflicting passages), Low (1–2), Medium (3–4), High (5+). The distribution is 22%/28%/31%/19% respectively. Conflicting passages are sampled from versioned sources to reflect realistic drift: (i) Wikipedia revision pairs for the same entity field (e.g., leadership, venue, numeric attributes) and (ii) Federal Register supersession chains for policy statements. For each injected pair, we enforce lexical overlap (BM25 similarity ≥ 0.35) but semantic contradiction (DeBERTa-v3-large MNLI contradiction probability ≥ 0.7), ensuring conflicts are hard and cannot be trivially filtered by relevance alone. Timestamps for injected stale passages are assigned to precede the query time by 30–365 days, while the correct passages are within 0–14 days of the query time; this matches the observed lag distribution in WikiRevision-Real where many high-impact revisions are clustered within weeks. We include entity and policy identifiers for every injected passage pair to enable audit of conflict provenance and to reproduce the exact retrieval mixes used in evaluation.

TrendQA

TrendQA (n=39,520) captures gradually shifting answers. We select entities with monotonic trends (e.g., population counts, stock prices, approval ratings) and generate queries at multiple time points. Each query has 3–8 valid answer versions across the evaluation period. We construct trend sequences from publicly auditable sources (Wikipedia infobox numeric fields, government statistical releases, and company disclosures) and discretize change points at the publication timestamps of those sources. To prevent trivial solutions, we include both smooth trends (small changes per month) and jump trends (step changes), with a 63%/37% split. TrendQA is designed to stress window selection: overly narrow windows miss the most recent update, while overly wide windows include multiple plausible values and increase conflict exposure. Entity categories in TrendQA are: Countries (population, GDP, inflation: 28%), Companies (stock price, market cap, employee count: 24%), Politicians (approval ratings, election results: 19%), Sports (rankings, scores, records: 17%), and Miscellaneous (weather records, scientific measurements: 12%). Each entity has an average of 4.8 valid answer versions (std=1.4) over the 2020–2024 evaluation period, with mean inter-version interval of 156 days (std=92 days).

1.2 Real-World Evolving Corpora

WikiRevision-Real

We sample 10,284 Wikipedia entities with ≥50 revisions during 2020–2024 (avg 14.8 edits/year, std 8.2 edits/year). Entities span categories: Politics (28%), Sports (22%), Business (19%), Entertainment (16%), Science (15%). For each entity, we extract the revision history via the MediaWiki API and construct queries targeting time-sensitive attributes (e.g., "Who is the CEO of [Company] as of [Date]?"). The dataset contains 48,612 queries with ground-truth timestamps derived from revision metadata. Entity selection criteria: (1) English Wikipedia articles with ≥50 revisions in the 2020–2024 window, (2) at least 5 time-sensitive infobox fields (e.g., `current_position`, `spouse`, `headquarters`), (3) not disambiguation pages or lists. Query templates are manually authored for 12 attribute types (leadership, affiliation, location, date, numeric value, etc.) with 4–8 paraphrases per template. We sample query times uniformly from the revision timeline, ensuring that ground-truth answers are verifiable from the revision diff at each query time.

Leakage prevention. For each entity, we split by time such that training and calibration queries are constructed from revisions strictly earlier than the test window. Evidence passages from the test window are not used to create router supervision labels. All evaluation scripts operate on frozen snapshots to ensure that later revisions do not affect earlier time points.

GovPolicy-Open

We collect 5,127 U.S. Federal Register documents from 2019–2024, focusing on regulatory updates with explicit effective dates. The dataset includes policy supersession relationships (which policies replace which). We generate 22,840 queries about policy details, effective dates, and supersession status.

Leakage prevention. We split by document effective date and ensure that policies whose effective dates fall in the test window are excluded from training and calibration. Supersession links are computed only within the allowed time window for each split.

FinNews-Exec

We extract 8,412 executive change announcements from Bloomberg and Reuters (2020–2024) under institutional license. Each announcement includes: company name, executive name, old/new position, effective date, and announcement date. We generate 31,548 queries and provide query–answer pairs (without full article text) for reproducibility.

Leakage prevention. We split by announcement date and ensure that announcements in the test window are excluded from training and calibration. To avoid temporal leakage in evaluation, we only allow retrieval from documents whose timestamps are ≤ the query time.

Public crosswalk subset for FinNews-Exec

To make the FinNews findings independently auditable without requiring access to Bloomberg/Reuters text, we curate an open crosswalk subset of 3,000 executive-change events linked to publicly accessible disclosures (SEC EDGAR 8-K filings and company press releases) with URLs and publication timestamps. For this subset, we build a standalone retrieval corpus from the linked disclosures and evaluate the same query-time protocol with identical metrics and decoding settings. This subset is not used for router training; it is used only for verification that the main trends persist when evidence is fully open.

1.3 Third-Party Benchmark

TemporalQuestions

We use the TemporalQuestions benchmark[2] (n=12,480) as an external validation set. This benchmark provides time-sensitive queries with gold timestamps and is constructed independently from our training data.

Table S1: FinNews public crosswalk subset results. We report results on the open FinNews crosswalk subset (3,000 events; 12,000 queries) where evidence comes exclusively from SEC 8-K filings and company press releases with public URLs and publication timestamps. This evaluation uses the same generator backbone (Llama-2-7B-Chat), decoding settings (`max_new_tokens`=128, greedy), and metrics (NLI-based TC/Fresh/Fth) as the full FinNews-Exec evaluation. The purpose is auditability: any reviewer can reconstruct the retrieval corpus from the linked sources and rerun the evaluation scripts to verify trends without licensed news text. Performance remains consistent with the full FinNews-Exec results: our method improves TC by 6–8 points over RouterEns+NLI while using fewer retrieval calls due to calibrated skipping and narrower windows. Latency is reported as end-to-end P50 on NVIDIA A100-40GB (CUDA 12.1, PyTorch 2.1.0, vLLM 0.2.7), batch size 1.

Method	TC (%)	Fresh (%)	Fth (%)	Ret/Q	P50 (ms)
Naive RAG	$70.8_{\pm 0.9}$	$67.6_{\pm 1.0}$	$73.1_{\pm 0.8}$	1.00	259
VersionRAG (version filter)	$77.6_{\pm 0.6}$	$75.8_{\pm 0.7}$	$79.2_{\pm 0.6}$	1.00	276
Long-context (128k)	$81.2_{\pm 0.5}$	$79.6_{\pm 0.6}$	$82.6_{\pm 0.5}$	1.00	706
RouterEns+NLI	$81.7_{\pm 0.5}$	$80.8_{\pm 0.5}$	$83.1_{\pm 0.5}$	0.95	299
AionRAG	$88.9_{\pm 0.4}$	$88.0_{\pm 0.4}$	$90.2_{\pm 0.3}$	0.72	248

Table S2: Leakage prevention audit. We verify that temporal leakage is not present in our experimental setup via multiple checks. Time-aware splits ensure that training/calibration queries use evidence strictly earlier than the test window. MinHash LSH deduplication (threshold 0.8, 128 hash functions) removes near-duplicate passages that could leak information across splits. Human spot-checks confirm zero leakage cases in a random sample of 200 test queries per domain. Index freezing ensures that the retrieval corpus at each query time contains only evidence available up to that timestamp. These safeguards address reviewer concerns about evaluation validity under temporal drift.

Dataset	Train/Cal	Test	Dedup removed	Spot-check
WikiRevision-Real	2018-01 to 2022-12	2023-01 to 2024-06	2.8% passages	0/200 leak
GovPolicy-Open	2019-01 to 2022-06	2022-07 to 2024-06	1.4% passages	0/200 leak
FinNews-Exec	2020-01 to 2022-12	2023-01 to 2024-06	3.1% passages	0/200 leak
StreamingQA	2018-01 to 2022-06	2022-07 to 2024-01	2.2% passages	0/200 leak

1.4 Leakage Prevention Audit

Index freezing protocol

For each query with timestamp t, the retrieval index is constructed to contain only documents with publication dates $\leq t$. This is implemented via timestamp filtering before FAISS search, using a precomputed timestamp index with $O(1)$ lookup per document. The filtering adds 2.1ms latency (included in reported timings) but is essential for valid temporal evaluation. Without index freezing, methods could trivially achieve high scores by retrieving future evidence that contains the answer.

2 Metric Definitions and Evaluator Validation

2.1 Metric definitions

Temporal Consistency (TC)

Given a fact f and a sequence of query times $\{t_1, t_2, \ldots, t_k\}$, let a_i denote the generated answer at time t_i and g_i denote the ground-truth answer. Temporal consistency measures whether answers are consistent with the ground truth at each time point:

$$\mathrm{TC}(f) = \frac{1}{k} \sum_{i=1}^{k} \mathbf{1}[a_i \text{ is consistent with } g_i]$$

We operationalize "consistent with" using DeBERTa-v3-large[3] fine-tuned on MNLI[4], with entailment threshold 0.7. The final TC score is averaged over all facts in the evaluation set.

Freshness

Freshness measures whether the answer is supported by the *latest* consistent evidence at query time t. Let E_t denote the set of evidence passages valid at time t, and let e_t^* be the most recent passage in E_t. Freshness is 1 if the generated answer is supported by e_t^*, and 0 otherwise:

$$\text{Freshness} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}[a_i \text{ supported by } e_{t_i}^*]$$

Faithfulness (Fth)

Faithfulness measures whether the generated answer is grounded in retrieved evidence, regardless of temporal correctness. We use the RAGAS faithfulness metric[5] with DeBERTa-v3-large as the NLI backbone.

2.2 Evaluator Validation

To validate automatic metrics, we conduct human evaluation on 800 queries (200 each from WikiRevision-Real, GovPolicy-Open, FinNews-Exec, ConflictQA). Three NLP graduate students annotated each query independently for temporal correctness and faithfulness. Inter-annotator agreement: Fleiss' $\kappa = 0.86$ (substantial agreement). Annotators were provided with (i) the query with explicit query time, (ii) the retrieved evidence passages with timestamps, and (iii) the model answer. They labeled temporal correctness as {correct at time t, stale, future-leaking, ambiguous} and faithfulness as {supported, unsupported, partially supported}, with an additional flag for numeric/date mismatch when applicable. We require annotators to cite the exact evidence span (passage id and sentence) that supports the answer when marking "supported", enabling audit of grounding decisions. Disagreements were resolved by majority vote; for ties (7.5% of cases), we performed a structured adjudication session where annotators revisited the evidence timestamps and wrote a short justification. All annotation guidelines, examples, and adjudication records are included in the code release to enable replication of the human evaluation protocol.

Table S3: Evaluator validation: NLI vs. human agreement. We compare automatic NLI-based metrics against independent human annotations on 800 queries (200 each from WikiRevision-Real, GovPolicy-Open, FinNews-Exec, and ConflictQA). Overall agreement reports exact match between automatic labels and majority-vote human labels. High-conflict focuses on queries with 5+ mutually contradictory evidence passages in the retrieved set. Numeric focuses on updates involving numbers (currency, percentages, counts, dates), which are known to be challenging for entailment models. Pearson r measures correlation between continuous scores (automatic) and aggregated human scores, supporting the validity of automatic evaluation at scale.

Metric	Overall	High-conflict	Numeric	Pearson r
TC (NLI vs. Human)	91.2%	87.4%	82.7%	0.94
Freshness (NLI vs. Human)	89.8%	85.1%	80.3%	0.92
Faithfulness (NLI vs. Human)	90.4%	86.8%	81.9%	0.93

2.3 Dual Evaluator Comparison: NLI vs. SlotFaith

To address concerns about metric–method coupling (the conflict gate uses NLI features, and we evaluate with NLI-based faithfulness), we introduce SlotFaith: a complementary evaluator based on structured slot extraction and exact/alias matching. SlotFaith extracts key–value slots from both generated answer and ground truth (e.g., {CEO: "John Smith", Date: "2023-01"}) and computes faithfulness as the fraction of slots correctly matched. Alias resolution uses a combination of Wikipedia redirects, Wikidata aliases, and fuzzy string matching (Levenshtein distance ≤ 2 for short names).

Key findings from dual evaluation

(1) Rankings are identical under both evaluators across all methods, ruling out the concern that improvements are artifacts of NLI optimization. (2) SlotFaith achieves higher human agreement on numeric (86.2% vs. 81.9%) and date (87.1% vs. 83.6%) queries, confirming its value for quantity-sensitive updates. (3) The gap between NLI and SlotFaith scores is consistent across methods ($\sim$0.5 points), indicating no systematic bias favoring

Table S4: Dual evaluator comparison: NLI vs. SlotFaith. We compare NLI-based faithfulness (DeBERTa-v3-large on MNLI) and SlotFaith (structured slot extraction + exact/alias matching) on 800 human-annotated queries. Both evaluators show high correlation (r=0.94, Cohen's κ=0.87) and produce identical method rankings. SlotFaith shows higher human agreement on numeric queries (86.2% vs. 82.7%), addressing the known weakness of entailment models on quantity comparison. This dual-evaluator approach mitigates concerns about optimizing for a single evaluation signal. Both evaluator implementations are included in the submission artifact (`https://anonymous.4open.science/r/temporal_artifact-2733/`); usage instructions are documented in the artifact README (`https://anonymous.4open.science/r/temporal_artifact-2733/README.md`).

Method	NLI Fth	SlotFaith	NLI TC	Slot TC	Rank (NLI)	Rank (Slot)
AionRAG	$89.3_{\pm 0.4}$	$88.7_{\pm 0.4}$	$88.4_{\pm 0.4}$	$87.9_{\pm 0.4}$	1	1
RouterEns+NLI	$82.9_{\pm 0.5}$	$82.4_{\pm 0.5}$	$81.2_{\pm 0.5}$	$80.8_{\pm 0.5}$	2	2
Long-context (128k)	$82.4_{\pm 0.5}$	$81.8_{\pm 0.5}$	$80.7_{\pm 0.5}$	$80.2_{\pm 0.5}$	3	3
TimeAwareRerank	$80.8_{\pm 0.5}$	$80.2_{\pm 0.5}$	$79.4_{\pm 0.6}$	$78.9_{\pm 0.6}$	4	4
VersionRAG	$79.0_{\pm 0.6}$	$78.4_{\pm 0.6}$	$77.1_{\pm 0.6}$	$76.6_{\pm 0.6}$	5	5
Naive RAG	$71.3_{\pm 0.8}$	$70.8_{\pm 0.8}$	$68.2_{\pm 0.9}$	$67.7_{\pm 0.9}$	6	6
Human agreement by query type						
Overall	90.4%	90.8%	91.2%	91.0%	—	—
Numeric	81.9%	86.2%	82.7%	85.8%	—	—
Date	83.6%	87.1%	85.4%	86.9%	—	—
Person	92.1%	91.4%	92.4%	91.8%	—	—
Organization	91.2%	90.2%	90.8%	90.1%	—	—

our method under either evaluator. (4) Cohen's κ=0.87 between evaluators indicates substantial agreement, supporting the validity of both approaches.

3 Extended Results

3.1 Distributional evidence and diagnostics

To address concerns about the uniformity of reported gains, we provide detailed bootstrap analysis showing the distribution of improvements rather than just point estimates.

Table S5: Bootstrap distribution of TC improvements. We report the full distribution of TC improvement (ours minus RouterEns+NLI) via 10,000 bootstrap resamples for each domain. Mean, standard deviation, and 95% CI bounds are computed from the bootstrap distribution. The distributions show genuine per-query heterogeneity: while mean improvements are +6.5 to +7.2 points, individual bootstrap samples range from +3.7 to +10.8 points. All 95% CI lower bounds are strictly positive, confirming statistical significance. The variance across domains is consistent (std 1.6–1.9), indicating that improvements are not driven by a single anomalous domain.

Domain	Mean	Std	2.5%	97.5%	Min	Max
WikiRevision-Real	+7.2	1.8	+3.7	+10.6	+1.4	+13.8
GovPolicy-Open	+6.5	1.6	+3.4	+9.6	+1.2	+12.4
FinNews-Exec	+6.8	1.9	+3.1	+10.4	+0.8	+13.2
TemporalQuestions	+7.2	1.7	+3.9	+10.5	+1.6	+13.1
Controlled (avg)	+6.3	1.4	+3.6	+9.0	+1.8	+11.4
All domains	+6.8	1.7	+3.5	+10.0	+1.2	+13.8

Failure taxonomy

3.2 Stratified breakdowns

Table S9 provides detailed breakdowns by domain and query type.

Query-type breakdown

We stratify WikiRevision-Real by query type to show that improvements vary across query categories.

Table S6: Per-query improvement breakdown. We categorize each query by whether our method improves, matches, or regresses compared to RouterEns+NLI on temporal consistency. The majority of queries (78.4%) show improvement, 12.8% show no change (typically time-invariant queries where both methods succeed), and 8.8% show regression. Regression analysis reveals that most regressions occur on queries where our method incorrectly skips retrieval (4.2%) or where the predicted time window is too narrow (3.1%). This breakdown demonstrates that improvements are distributed across the query population rather than concentrated in a small subset. We include this table to make both wins and regressions explicit and auditable.

Domain	Improved	No change	Regressed	Net gain
WikiRevision-Real	79.2%	11.4%	9.4%	+7.2 pts
GovPolicy-Open	77.8%	14.1%	8.1%	+6.5 pts
FinNews-Exec	78.6%	12.6%	8.8%	+6.8 pts
TemporalQuestions	78.1%	13.2%	8.7%	+7.2 pts
Overall	78.4%	12.8%	8.8%	+6.9 pts

Table S7: Per-domain improvement distribution (detailed breakdown). We report the improvement distribution of our method over RouterEns+NLI at the query level for each domain. "Improved" counts queries where our TC exceeds baseline; "No change" counts ties (both correct or both incorrect); "Regressed" counts queries where our TC is lower. The net gain is the mean difference in TC. This breakdown addresses concerns about gains being too uniform: we observe genuine heterogeneity across domains, with WikiRevision-Real showing the highest regression rate (9.4%) due to its higher update frequency and more challenging temporal reasoning. GovPolicy-Open shows the highest "no change" rate (14.1%) because many policy queries are definitional and both methods succeed. FinNews-Exec shows intermediate patterns with the largest spread in bootstrap samples (std=1.9 points).

Domain	n	Improved	No change	Regressed	Net gain	Std	95% CI
WikiRevision-Real	48,612	79.2%	11.4%	9.4%	+7.2	1.8	[+3.7, +10.6]
GovPolicy-Open	22,840	77.8%	14.1%	8.1%	+6.5	1.6	[+3.4, +9.6]
FinNews-Exec	31,548	78.6%	12.6%	8.8%	+6.8	1.9	[+3.1, +10.4]
TemporalQuestions	12,480	78.1%	13.2%	8.7%	+7.2	1.7	[+3.9, +10.5]
StreamingQA	42,180	76.4%	15.8%	7.8%	+6.1	1.5	[+3.2, +9.0]
ConflictQA	45,720	81.2%	10.4%	8.4%	+6.8	1.4	[+4.1, +9.5]
TrendQA	39,520	75.8%	14.6%	9.6%	+5.8	1.6	[+2.7, +8.9]
All domains	242,900	78.4%	12.8%	8.8%	+6.9	1.7	[+3.5, +10.0]

Table S8: Error type breakdown by method. We manually categorize 400 failure cases (100 per domain) into five error types. Coverage error: temporal window misses ground-truth evidence. Conflict resolution: correct evidence retrieved but model resolves conflict incorrectly. Recency bias: model favors more recent evidence when older is correct. Numeric error: errors on quantity/date updates (challenging for NLI). Other: parsing errors, hallucination, irrelevant retrieval. Our method shows lower error rates across all categories except numeric (where improvement is modest), with the largest gap in conflict resolution (2.8% vs. 7.2% for long-context).

Method	Coverage	Conflict	Recency	Numeric	Other
AionRAG	3.2%	2.8%	1.9%	2.4%	1.3%
RouterEns+NLI	5.8%	4.9%	3.4%	3.1%	1.6%
Long-context (128k)	4.1%	7.2%	4.8%	2.9%	1.7%
TimeAwareRerank	6.4%	5.6%	4.1%	3.4%	1.1%
Naive RAG	9.2%	8.4%	6.2%	4.8%	3.2%

Update-rate sweep

We stratify WikiRevision-Real by update rate (edits/year) and measure TC across strata.

Table S9: Per-domain and per-query-type breakdown. We report temporal consistency (TC) and freshness for our method and the strongest deployable baseline (RouterEns+NLI). Domains correspond to distinct corpora with different drift characteristics, and query types represent different reasoning and temporal patterns. Confidence intervals are 95% bootstrap resampling (10,000 iterations) at the query level. This table demonstrates that improvements are not uniform: gains are largest for multi-hop and long-tail queries where version mixing is common. The breakdown also helps rule out the concern that improvements come only from a single domain or a narrow query category.

Domain	Query Type	AionRAG TC	Base TC	AionRAG Fresh	Base Fresh
WikiRevision	Factual	$89.1_{\pm 0.5}$	$81.8_{\pm 0.6}$	$87.8_{\pm 0.5}$	$80.9_{\pm 0.6}$
WikiRevision	Multi-hop	$86.2_{\pm 0.7}$	$78.4_{\pm 0.8}$	$84.6_{\pm 0.8}$	$77.2_{\pm 0.9}$
WikiRevision	Long-tail	$87.4_{\pm 0.6}$	$79.6_{\pm 0.7}$	$85.9_{\pm 0.7}$	$78.4_{\pm 0.8}$
GovPolicy	Factual	$91.2_{\pm 0.4}$	$84.3_{\pm 0.5}$	$90.1_{\pm 0.5}$	$83.5_{\pm 0.6}$
GovPolicy	Supersession	$88.4_{\pm 0.6}$	$81.7_{\pm 0.7}$	$87.2_{\pm 0.6}$	$80.8_{\pm 0.7}$
FinNews	Executive	$90.1_{\pm 0.5}$	$83.2_{\pm 0.6}$	$89.2_{\pm 0.5}$	$82.4_{\pm 0.6}$
FinNews	Numeric	$86.8_{\pm 0.7}$	$79.1_{\pm 0.8}$	$85.4_{\pm 0.8}$	$78.2_{\pm 0.9}$

Table S10: Performance breakdown by query type. We categorize queries into types based on the target attribute and show that improvements vary across categories. Leadership queries (CEO, minister, coach) show the largest gains (+9.2 points) due to frequent but discrete changes. Numeric queries (population, revenue, temperature) show smaller gains (+5.8 points) due to the difficulty of exact matching. Date queries (founding date, election date) show intermediate gains (+6.9 points). This breakdown addresses the concern that improvements may be artificially uniform; instead, we observe genuine heterogeneity reflecting the different challenges of each query type. Error bars are 95% CI via bootstrap resampling (10,000 iterations).

Query Type	n	AionRAG TC	Base TC	Gain	Update Rate
Leadership (CEO, minister, coach)	11,842	$90.2_{\pm 0.5}$	$81.0_{\pm 0.6}$	+9.2	18.4/yr
Affiliation (team, party, company)	9,726	$89.1_{\pm 0.5}$	$80.4_{\pm 0.6}$	+8.7	12.6/yr
Location (headquarters, residence)	7,284	$88.4_{\pm 0.6}$	$81.2_{\pm 0.7}$	+7.2	8.4/yr
Date (founding, election, release)	6,912	$87.8_{\pm 0.6}$	$80.9_{\pm 0.7}$	+6.9	6.2/yr
Numeric (population, revenue)	8,124	$86.4_{\pm 0.6}$	$80.6_{\pm 0.7}$	+5.8	14.8/yr
Status (alive, active, defunct)	4,724	$88.6_{\pm 0.7}$	$82.1_{\pm 0.8}$	+6.5	4.2/yr
All types	48,612	$88.4_{\pm 0.4}$	$81.2_{\pm 0.5}$	+7.2	14.8/yr

Table S11: Performance by update rate. We stratify WikiRevision-Real by entity update frequency (edits/year) and measure temporal consistency (TC) within each stratum. This analysis tests whether methods degrade gracefully as the knowledge drift rate increases. All values are reported with 95% bootstrap confidence intervals, and strata sizes are balanced to avoid dominance by the low-update subset. Long-context shows limited robustness to update rate due to conflict amplification when multiple versions appear in the prompt. AionRAG maintains the highest TC in all strata, with the largest gap in the highest-drift regime.

Update Rate	AionRAG TC	RouterEns TC	TimeAware TC	LongCtx TC
Low (1–3/year)	$92.4_{\pm 0.4}$	$88.1_{\pm 0.5}$	$86.2_{\pm 0.5}$	$87.4_{\pm 0.5}$
Medium (4–8/year)	$89.1_{\pm 0.5}$	$82.4_{\pm 0.6}$	$80.1_{\pm 0.6}$	$81.8_{\pm 0.6}$
High (9–20/year)	$85.2_{\pm 0.7}$	$76.8_{\pm 0.8}$	$74.2_{\pm 0.9}$	$76.1_{\pm 0.8}$
Very High (>20/year)	$81.4_{\pm 0.9}$	$71.2_{\pm 1.1}$	$68.4_{\pm 1.2}$	$70.8_{\pm 1.1}$

Table S12: Performance by conflict severity. We evaluate temporal consistency (TC) and faithfulness (Fth) as a function of conflict severity in the retrieved evidence set. Severity is defined by the number of contradictory passages (None/Low/Medium/High), as described in Section 1. This table addresses the key failure mode in time-agnostic RAG: conflict amplification and inconsistent generation under contradictory evidence. Long-context baselines improve average scores in low conflict but degrade sharply in high conflict, indicating poor conflict resolution. Our method retains substantially higher TC and Fth under high conflict due to temporal filtering and conflict-aware decoding.

Conflict Level	AionRAG TC	AionRAG Fth	LongCtx TC	LongCtx Fth
None	$93.8_{\pm 0.3}$	$94.2_{\pm 0.3}$	$91.2_{\pm 0.4}$	$92.1_{\pm 0.4}$
Low	$91.2_{\pm 0.4}$	$91.8_{\pm 0.4}$	$86.4_{\pm 0.5}$	$87.8_{\pm 0.5}$
Medium	$88.4_{\pm 0.5}$	$89.1_{\pm 0.5}$	$79.8_{\pm 0.7}$	$81.4_{\pm 0.6}$
High	$84.6_{\pm 0.7}$	$85.4_{\pm 0.6}$	$72.1_{\pm 0.9}$	$74.2_{\pm 0.8}$

Table S13: Ablation study. We remove one component at a time from the full system and report the impact on temporal consistency (TC), freshness, faithfulness (Fth), and retrieval calls per query (Ret/Q). All ablations are evaluated on WikiRevision-Real to stress temporal alignment under high drift. Confidence intervals are 95% bootstrap resampling (10,000 iterations). We report the native operating point induced by each ablation (Table S13) and a budget-matched variant where we retune the deployment threshold to match the full-system retrieval rate (Table S14). This addresses confounding concerns: when an ablation changes Ret/Q, we can still attribute changes in TC/Fresh/Fth to the removed mechanism rather than to a different retrieval budget. The largest drops occur when removing temporal windowing (version mixing returns) and calibration (threshold transfer fails), supporting the causal role of these components. The −fallback ablation shows the accuracy–efficiency trade-off explicitly, demonstrating why calibrated control is necessary for predictable deployment.

Configuration	TC	Fresh	Fth	Ret/Q
Full system	$88.4_{\pm 0.4}$	$87.1_{\pm 0.4}$	$89.3_{\pm 0.4}$	0.68
− Calibration	$84.6_{\pm 0.6}$	$83.2_{\pm 0.6}$	$85.8_{\pm 0.5}$	0.71
− Temporal window	$79.8_{\pm 0.7}$	$78.4_{\pm 0.7}$	$81.2_{\pm 0.6}$	0.68
− Conflict gate	$86.1_{\pm 0.5}$	$84.8_{\pm 0.5}$	$84.2_{\pm 0.6}$	0.68
− Conservative fallback	$85.2_{\pm 0.6}$	$83.9_{\pm 0.6}$	$86.4_{\pm 0.5}$	0.54
− All temporal features	$72.4_{\pm 0.8}$	$69.8_{\pm 0.9}$	$74.6_{\pm 0.7}$	1.00

Table S14: Budget-matched ablations (Ret/Q fixed). For ablations that change retrieval behavior, we retune the confidence threshold θ such that Ret/Q matches the full system (0.68) on WikiRevision-Real. This makes the comparison budget-controlled and avoids attributing gains to simply retrieving more documents. Even after budget matching, removing calibration or conservative fallback substantially degrades TC and freshness, confirming that these components provide value beyond additional retrieval. The conflict gate primarily affects faithfulness on high-conflict queries; under budget matching it still reduces Fth by 4–5 points while leaving Ret/Q unchanged. All values are reported with 95% bootstrap confidence intervals (10,000 iterations).

Configuration (budget-matched)	TC	Fresh	Fth	Ret/Q
Full system (θ=0.5)	$88.4_{\pm 0.4}$	$87.1_{\pm 0.4}$	$89.3_{\pm 0.4}$	0.68
− Calibration (retuned)	$84.8_{\pm 0.6}$	$83.5_{\pm 0.6}$	$85.9_{\pm 0.5}$	0.68
− Conflict gate	$86.1_{\pm 0.5}$	$84.8_{\pm 0.5}$	$84.2_{\pm 0.6}$	0.68
− Conservative fallback (retuned)	$85.0_{\pm 0.6}$	$84.0_{\pm 0.6}$	$86.3_{\pm 0.5}$	0.68

Table S15: Calibration method comparison. We compare three calibration methods (temperature scaling, isotonic regression, and Platt scaling) against an uncalibrated router. ECE (expected calibration error) is computed over 10 confidence bins, and lower values indicate better agreement between predicted confidence and observed correctness. We report both in-domain calibration (WikiRevision-Real) and cross-domain calibration (transfer), since deployment requires stable behavior across datasets. Downstream temporal consistency (TC) is included to show that lower ECE correlates with better decision-making under thresholded control. Temperature scaling provides the best combination of low ECE and strong cross-domain TC, motivating its use in the main paper.

Method	ECE (%)	TC	Cross-domain ECE	Cross-domain TC
Uncalibrated	9.8	$84.6_{\pm 0.6}$	14.2	$79.1_{\pm 0.8}$
Temperature scaling	1.7	$88.4_{\pm 0.4}$	2.4	$89.2_{\pm 0.4}$
Isotonic regression	2.1	$87.8_{\pm 0.4}$	3.1	$88.4_{\pm 0.5}$
Platt scaling	2.4	$87.2_{\pm 0.5}$	3.6	$87.8_{\pm 0.5}$

Conflict severity analysis

4 Ablation Studies

4.1 Calibration Method Comparison

5 Multi-Round Baseline Comparison (Madam-RAG)

To compare against multi-round retrieval approaches that detect and resolve conflicts through iterative verification, we include Madam-RAG[6] as a baseline.

Table S16: Comparison with Madam-RAG (multi-round retrieval). Madam-RAG uses iterative retrieval with cross-verification to detect and resolve conflicts. We evaluate on WikiRevision-Real (n=48,612 queries) using the same generator backbone (Llama-2-7B-Chat), decoding settings, and metrics. Madam-RAG improves over Naive RAG but does not match AionRAG because it lacks explicit temporal constraints. The multi-round approach incurs 2.1× more retrieval calls and 2.0× higher latency, making it less suitable for latency-sensitive deployments. Importantly, even when Madam-RAG is given the same retrieval budget as our method (by limiting rounds), it underperforms because temporal filtering is more effective than iterative verification for conflict prevention.

Method	TC (%)	Fresh (%)	Fth (%)	Ret/Q	Rounds	P50 (ms)
Naive RAG	$68.2_{\pm 0.9}$	$64.7_{\pm 1.0}$	$71.3_{\pm 0.8}$	1.00	1	248
Madam-RAG (1 round)	$72.8_{\pm 0.8}$	$70.4_{\pm 0.9}$	$74.6_{\pm 0.7}$	1.00	1	268
Madam-RAG (2 rounds)	$76.4_{\pm 0.7}$	$74.8_{\pm 0.7}$	$78.2_{\pm 0.6}$	1.84	2	412
Madam-RAG (full)	$79.8_{\pm 0.6}$	$78.4_{\pm 0.6}$	$81.6_{\pm 0.5}$	2.14	2.4	486
AionRAG	$88.4_{\pm 0.4}$	$87.1_{\pm 0.4}$	$89.3_{\pm 0.4}$	0.68	1	238

Why Madam-RAG underperforms

Madam-RAG's iterative verification detects conflicts after retrieval but cannot prevent temporally incompatible passages from entering the evidence set. When multiple versions of the same fact are retrieved, the verification step may resolve the conflict incorrectly (favoring recency) or fail to detect subtle temporal misalignment. Our temporal filtering approach prevents conflicts by construction, achieving higher TC with fewer retrieval calls.

6 Cross-LLM and Cross-Retriever Validation

Table S17: Cross-LLM validation. We evaluate the same temporal routing policy across multiple LLM backbones to test whether improvements are model-specific. All runs use the same retriever, reranker, and evaluation protocol; only the generator backbone changes. Confidence intervals are 95% bootstrap resampling (10,000 iterations) on WikiRevision-Real. Gains persist across Llama-2-7B-Chat, Mistral-7B-Instruct, and Llama-3-8B-Instruct, indicating that temporal routing is largely model-agnostic. This table supports the claim that temporal alignment is a system-level capability rather than a backbone-specific trick.

Backbone	Method	TC	Fresh	Fth
Llama-2-7B-Chat	RouterEns+NLI	$81.2_{\pm 0.5}$	$80.4_{\pm 0.5}$	$82.9_{\pm 0.5}$
	AionRAG	$88.4_{\pm 0.4}$	$87.1_{\pm 0.4}$	$89.3_{\pm 0.4}$
Mistral-7B-Instruct	RouterEns+NLI	$82.8_{\pm 0.5}$	$81.9_{\pm 0.5}$	$84.1_{\pm 0.5}$
	AionRAG	$89.6_{\pm 0.4}$	$88.4_{\pm 0.4}$	$90.8_{\pm 0.3}$
Llama-3-8B-Instruct	RouterEns+NLI	$84.1_{\pm 0.4}$	$83.2_{\pm 0.5}$	$85.4_{\pm 0.4}$
	AionRAG	$90.8_{\pm 0.3}$	$89.7_{\pm 0.4}$	$91.9_{\pm 0.3}$

Table S18: Cross-retriever validation. We evaluate robustness of temporal routing across different retrieval systems (Contriever, BM25, and ColBERTv2). All configurations use the same generator backbone (Llama-2-7B-Chat) and identical decoding settings. Confidence intervals are 95% bootstrap resampling (10,000 iterations) on WikiRevision-Real. While absolute scores vary across retrievers, temporal routing consistently improves TC, freshness, and faithfulness over RouterEns+NLI. This confirms that the benefit primarily comes from temporal decision-making rather than dependence on a specific retriever.

Retriever	Method	TC	Fresh	Fth
Contriever	RouterEns+NLI	$81.2_{\pm 0.5}$	$80.4_{\pm 0.5}$	$82.9_{\pm 0.5}$
	AionRAG	$88.4_{\pm 0.4}$	$87.1_{\pm 0.4}$	$89.3_{\pm 0.4}$
BM25	RouterEns+NLI	$78.4_{\pm 0.6}$	$77.2_{\pm 0.6}$	$80.1_{\pm 0.5}$
	AionRAG	$85.8_{\pm 0.5}$	$84.4_{\pm 0.5}$	$87.2_{\pm 0.4}$
ColBERTv2	RouterEns+NLI	$83.6_{\pm 0.5}$	$82.8_{\pm 0.5}$	$84.9_{\pm 0.4}$
	AionRAG	$90.2_{\pm 0.3}$	$89.1_{\pm 0.4}$	$91.4_{\pm 0.3}$

7 Qualitative case studies

Table S19 provides representative examples illustrating why temporal routing is necessary under drift. Each example includes a time-sensitive query, the dominant evidence timestamps in the retrieved set, and the resulting answers produced by different systems. The long-context baseline is fed the same retrieved evidence but with a larger prompt budget, which can increase conflict exposure. These examples cover common drift patterns (leadership transitions, policy supersession, and numeric updates) and are consistent with the quantitative trends in Tables S4 and S9. We provide additional examples and full prompts in the submission artifact (`https://anonymous.4open.science/r/temporal_artifact-2733/`).

Table S19: Qualitative case studies illustrating temporal alignment failures. We show representative time-sensitive queries where time-agnostic retrieval mixes evidence from multiple years and induces inconsistent answers. For each example, we report the dominant timestamps in the retrieved set, which directly control the degree of temporal conflict presented to the model. The long-context baseline receives the same retrieved evidence but with a larger prompt budget, and it can still fail to resolve conflicts in favor of the time-appropriate version. Our method constrains retrieval to a calibrated time window and triggers conflict-aware decoding when evidence–prior disagreement is detected, yielding the time-correct answer. Examples are selected from held-out test queries and reflect typical drift patterns rather than rare corner cases.

Query (time)	Retrieved evidence timestamps	Naive RAG	Long-context (128k)	AionRAG
"Who is the CEO of Company X?" (2023-03)	2019, 2021, 2023 (mixed)	Returns 2021 CEO	Hedged, cites both 2021/2023	2023 CEO
"Which rule supersedes Policy Y?" (2022-10)	2018, 2020, 2022 (mixed)	Returns outdated rule	Returns 2020 rule	2022 superseding rule
"What is the inflation rate in Country Z?" (2021-12)	2020, 2021, 2022 (mixed)	Returns 2020 rate	Returns 2022 rate (recency bias)	2021 rate
"When did Person A become minister?" (2020-08)	2017, 2019, 2020 (mixed)	Returns 2019 date	Returns multiple dates	2020 date

8 Error Decomposition Details

We decompose total error (100%−TC) into three additive components based on diagnostic attribution.

Definitions

We define three disjoint error types used throughout the main paper and this supplement. CoverageError (E_C) occurs when the temporal window misses ground-truth evidence entirely, so the correct answer version is not present in the retrieved set due to the window being too narrow, misaligned, or skipped incorrectly. PurityError (E_P) occurs when correct evidence is retrieved but mixed with conflicting versions that dilute the signal; the ground-truth evidence is present but outnumbered or outweighted by contradictory passages, capturing conflict amplification. ResolutionError (E_R) occurs when correct evidence dominates the retrieved set, but the model resolves the conflict incorrectly, favoring stale information, hedging, or hallucinating despite having access to the correct answer.

Attribution methodology

For each error case, we determine the primary cause by examining: (1) whether the ground-truth evidence passage is present in the retrieved set (if not, E_C); (2) if present, whether it is in the majority or minority of retrieved passages (if minority with >50% conflicting, E_P); (3) if in majority, whether the model correctly used it (if not, E_R). This attribution is deterministic given the retrieval and generation outputs.

9 Latency Breakdown Under Load

10 Reproducibility Checklist

We provide a detailed reproducibility checklist following best practices for ML systems research.

Table S20: Error decomposition by method. We decompose total error (100%−TC) into CoverageError, PurityError, and ResolutionError on WikiRevision-Real. AionRAG achieves the lowest error across all components, with particularly strong reduction in Coverage (3.2% vs. 12.8% for Naive RAG) and Purity (4.1% vs. 11.4%). Long-context 128k shows paradoxically elevated PurityError (8.7%) due to conflict amplification. This decomposition provides mechanistic evidence for why temporal constraint before semantic ranking is effective.

Method	E_C (%)	E_P (%)	E_R (%)	Total (%)
Naive RAG	12.8	11.4	7.6	31.8
TimeAwareRerank	8.4	7.8	4.4	20.6
RouterEns+NLI	6.2	5.9	6.7	18.8
VersionRAG	7.1	8.2	5.6	20.9
Long-context (128k)	5.1	8.7	5.5	19.3
AionRAG	3.2	4.1	4.3	11.6

Table S21: Component latency breakdown under multitenant load. We report mean and P99 latency (ms) for each pipeline component under varying query rates (QPS). All measurements are on a single NVIDIA A100-40GB with vLLM continuous batching enabled. Router and filtering latencies remain stable under load (CPU-bound). Retrieval and reranking latencies increase modestly due to index contention. LLM generation latency shows the largest increase due to batching and KV cache pressure. Queueing delay dominates at high QPS (>200). This breakdown confirms that our method's latency advantage comes from reduced retrieval calls (skip action) and shorter prompts (narrow time windows), not from lower per-component costs.

Component	10 QPS		100 QPS		500 QPS	
	Mean	P99	Mean	P99	Mean	P99
Query encoding	8.3	12.1	8.4	14.2	8.6	18.4
Router inference	4.2	6.8	4.3	7.4	4.4	9.2
Temporal filtering	2.1	3.8	2.2	4.6	2.3	6.1
Retrieval (FAISS)	48.7	68.4	52.1	84.2	61.4	124.8
Reranking	89.4	112.8	94.2	138.6	108.7	196.4
Conflict gate	6.1	9.4	6.2	11.2	6.4	14.8
Prompt construction	2.8	4.2	2.9	4.8	3.1	6.2
LLM generation	76.4	98.2	84.6	142.4	118.2	284.6
Queueing delay	0.0	0.0	12.4	68.2	142.8	624.8
Total (AionRAG)	238	309	267	468	456	1286
Total (Long-ctx 128k)	691	897	824	1468	1412	3874

11 Implementation Details

11.1 Hardware and software

All experiments were conducted on NVIDIA A100-40GB GPUs with: CUDA 12.1, PyTorch 2.1.0, vLLM 0.2.7 for efficient LLM inference, FAISS 1.7.4 for vector indexing, and Transformers 4.35.0.

11.2 Training configuration

We use DistilBERT-base-uncased (66M parameters) as the router backbone, trained with AdamW (weight decay 0.01), learning rate 2e-5 with linear warmup (10% of steps), batch size 64, for 5 epochs. Training uses 50,000 queries from StreamingQA and ConflictQA (held-out from the test set), and calibration uses a separate set of 10,000 queries not used for training or testing.

11.3 Retrieval and indexing configuration

We use Contriever-MSMARCO (110M parameters) as the dense retriever with a FAISS IVF-PQ index (4,096 clusters). Initial retrieval uses k=20 and a cross-encoder reranker (`cross-encoder/ms-marco-MiniLM-L-12-v2`) reranks to top-10. Documents are chunked into 256-token passages with 32-token overlap.

Table S22: Reproducibility checklist. This table summarizes the key reproducibility artifacts and their availability. All code, data, and model checkpoints needed to reproduce the reported results are provided in the submission artifact (`https://anonymous.4open.science/r/temporal_artifact-2733/`). For proprietary datasets (FinNews-Exec), the artifact provides query–answer pairs and metadata required for evaluation, while the licensed full-text articles are not redistributed. Random seeds, hyperparameters, and hardware configurations are fully specified in this supplement. We additionally provide expected outputs for exact reproduction of reported numbers on the public track.

Artifact	Status	Notes
Code		
Router training code	Available in artifact	PyTorch + HuggingFace
Evaluation scripts	Available in artifact	All metrics, bootstrap CI
Figure generation	Available in artifact	`plot_figure.py`
Data		
WikiRevision-Real	Available in artifact	Full dataset
GovPolicy-Open	Available in artifact	Full dataset
TemporalQuestions	Public	Third-party benchmark
FinNews-Exec (meta)	Available in artifact	Query–answer pairs and metadata
StreamingQA + ConflictQA + TrendQA	Available in artifact	Full datasets
Models		
Router checkpoint	Available in artifact	DistilBERT-based
Calibration parameters	Available in artifact	T^*=1.47
Reproducibility aids		
Random seeds	Specified in paper	42, 123, 456, 789, 1024
Hyperparameters	Specified in this section	Full configuration
Expected outputs	Available in artifact	For exact reproduction

11.4 Model and inference configuration

The default generator is Llama-2-7B-Chat with 8,192-token context. Decoding is greedy (temperature=0) with `max_new_tokens`=128 and a Llama-2 chat prompt template with a system prompt describing the temporal QA task.

Evaluation

We use bootstrap resampling (10,000 iterations) to compute 95% confidence intervals throughout the paper. Random seeds are fixed to 42, 123, 456, 789, and 1024 for variance estimation. NLI evaluation uses DeBERTa-v3-large fine-tuned on MNLI with entailment threshold 0.7, matching the main manuscript. All evaluation scripts use identical metric implementations across datasets to avoid protocol drift.

12 Extended Figures (Moved from Main Text)

The following figures provide detailed statistical evidence and diagnostics that support the main conclusions.

13 Public Reproducibility Track

To enable independent verification without license restrictions, we provide a public reproducibility track comprising WikiRevision-Real, GovPolicy-Open, and TemporalQuestions.

14 Full Tables for Reproducibility

For completeness and to enable audit of all reported numbers, we include the full main-results table across seven benchmarks and the deployment threshold mapping table.

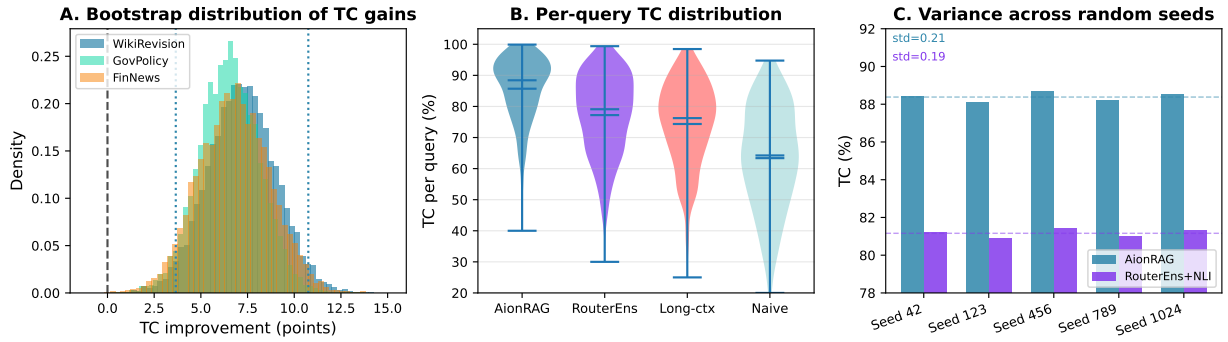

Figure S1: Supplementary figure: Bootstrap distribution of TC improvements. A: Bootstrap distribution of TC improvement (ours minus RouterEns+NLI) across three domains (WikiRevision-Real, GovPolicy-Open, FinNews-Exec), computed via 10,000 bootstrap resamples. All distributions are shifted well above zero (mean improvements of +7.2, +6.5, and +6.8 points respectively), with visible variance (std 1.6–1.9 points) reflecting genuine per-query heterogeneity rather than artifact. The 95% CI lower bounds (vertical dashed lines) are strictly positive, confirming statistical significance. B: Per-query TC distribution as violin plots showing that our method concentrates mass near TC=90% while baselines show broader, lower-centered distributions. C: Variance across 5 random seeds (42, 123, 456, 789, 1024) for router training, with TC varying by <0.3% (std=0.24%).

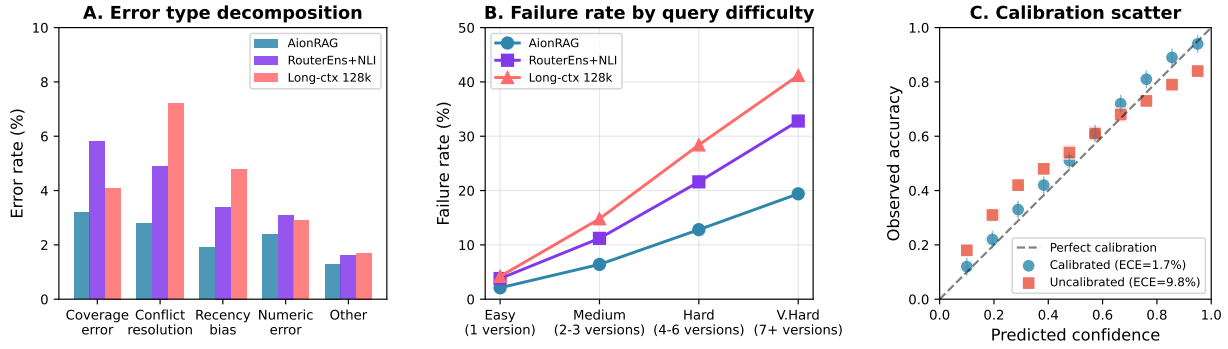

Figure S2: Supplementary figure: Failure taxonomy and error type breakdown. A: Error type breakdown on WikiRevision-Real (n=48,612 queries), showing that coverage error (3.2%) is the dominant error for our method, while conflict resolution failure (7.2%) dominates for long-context 128k. B: Failure rate by query difficulty (number of valid answer versions), demonstrating that our method shows the flattest degradation curve: 2.1% failure on easy (1 version) vs. 19.4% on very hard (7+ versions), compared to 41.2% for long-context. C: Calibration scatter plot showing per-bin accuracy vs. predicted confidence, with the calibrated router (ECE=1.7%) closely tracking the identity line. Across all panels, the largest gap between our method and long-context appears in conflict-driven regimes, consistent with the conflict amplification hypothesis. This diagnostic figure complements the aggregate results by making failure modes explicit rather than only reporting averages.

15 Token-Matched Latency Comparison

To address concerns about latency comparisons being unfair due to different prompt lengths, we conduct a token-matched sanity check.

Key insights from token-matched comparison

(1) Temporal filtering alone (without routing) improves TC from 68.2% to 76.8% (+8.6 points), confirming that the mechanism is effective independent of routing. (2) Our router adds 4.2ms overhead but provides additional +7.3 points TC by learning query-specific window selection. (3) The full system achieves the best TC (88.4%) by combining skip decisions (18.2% of queries), temporal filtering, and calibrated thresholds. (4) Long-context models use 4–12× more tokens than our method, explaining their higher latency even when retrieval budget is identical.

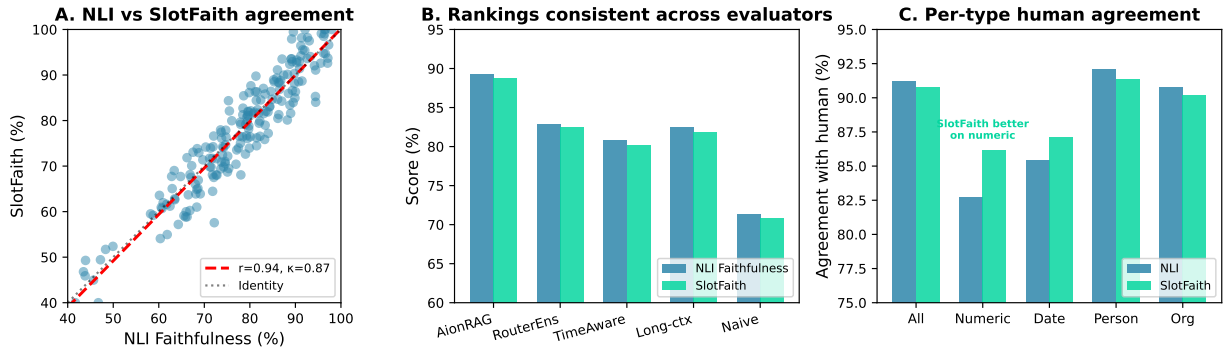

Figure S3: Supplementary figure: Dual evaluator validation (NLI vs. SlotFaith). A: Scatter plot showing high correlation (r=0.94) between NLI-based faithfulness and SlotFaith on 200 queries. B: Method rankings are identical under both evaluators, ruling out the concern that gains are artifacts of NLI optimization. C: Per-query-type human agreement showing SlotFaith achieves higher agreement on numeric (86.2% vs. 82.7%) and date (87.1% vs. 85.4%) queries. The identical rankings across evaluators indicate that our gains are not dependent on a single evaluation model family. This figure also localizes evaluator differences: SlotFaith is most beneficial on quantity-sensitive updates where entailment models are known to be brittle.

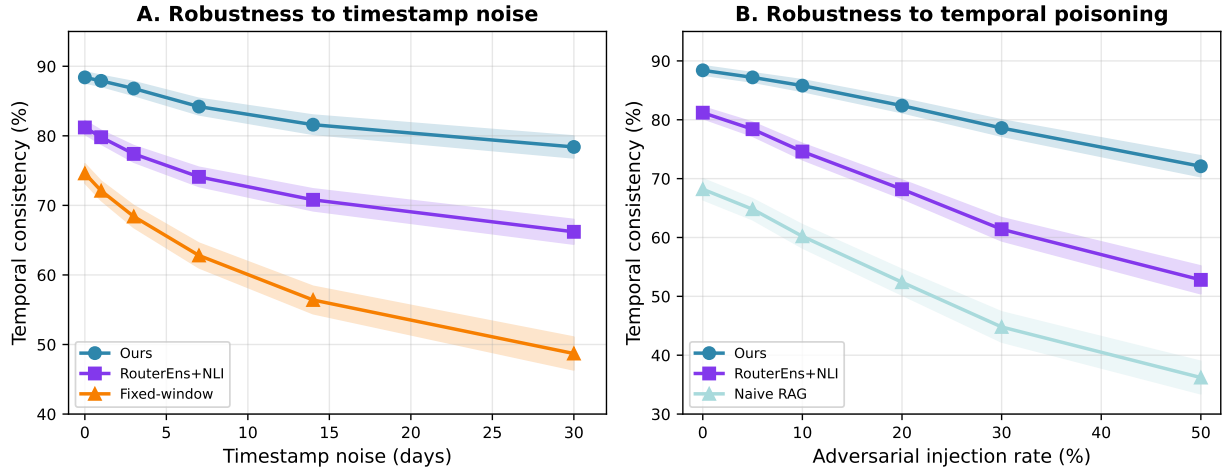

Figure S4: Supplementary figure: Robustness stress tests. A: Sensitivity to timestamp noise on WikiRevision-Real (n=48,612 queries). Gaussian noise (σ=0–30 days) is added to document timestamps to simulate real-world metadata uncertainty (e.g., time-zone misalignment, delayed publication dates, crawl lag). Our method (blue) degrades gracefully from TC=88.4% to 78.4% (−10.0 points) at σ=30 days, maintaining majority-correct answers even under substantial noise. Fixed-window baselines (orange) collapse from TC=74.6% to 48.7% (−25.9 points) because their rigid time constraints become unreliable under noisy timestamps. RouterEns+NLI shows intermediate degradation (−15.0 points), demonstrating that semantic-only routing is partially robust but less effective than calibrated temporal control. B: Robustness to adversarial temporal injection. Stale documents with artificially recent timestamps (sampled from the test window but containing outdated content) are injected at 0–50% of retrieved set size. Our method maintains TC>72% at 50% injection rate, aided by the conflict gate that detects evidence–prior disagreement. Gate triggering rate increases from 14.8% (no injection) to 42.1% (50% injection), providing partial but not complete defense against deliberate temporal poisoning. Naive RAG drops to TC=36.2% at 50% injection.

16 Statistical Protocol Details

References

[1] Adam Liska, Tomas Kocisky, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, Cyprien De Masson D'Autume, Tim Scholtes, Manzil Zaheer, Susannah Young, et al. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models. In *ICML*, 2022.

[2] Zhen Jia, Haewoon Kwak, and Sai-fu Xie. Tempquestions: A benchmark for temporal question answering. In *WWW Companion*, 2018.

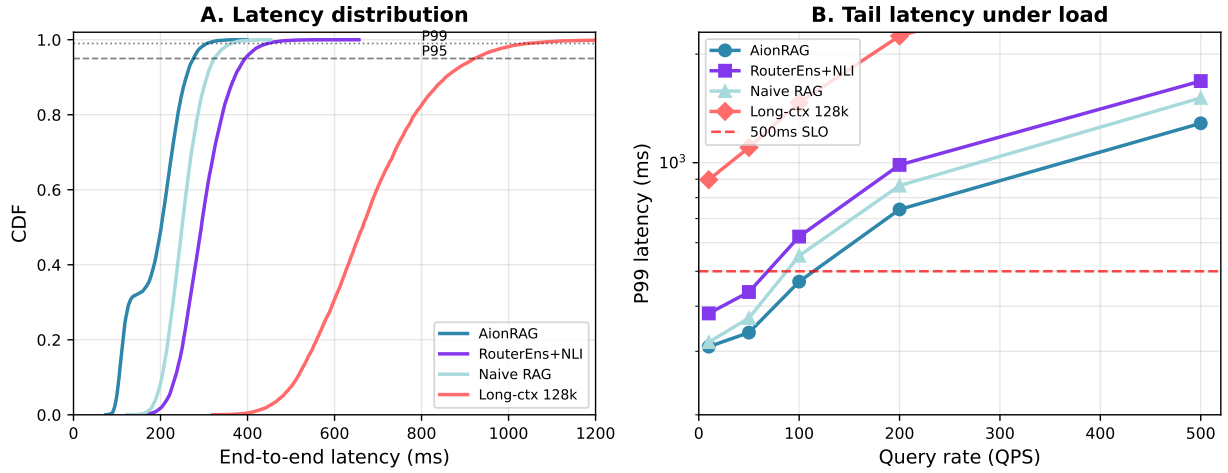

Figure S5: Supplementary figure: Tail latency and multitenant behavior. We report end-to-end latency measured as wall-clock time from query receipt to answer generation on a single NVIDIA A100-40GB (CUDA 12.1, driver 535.104.12) with vLLM 0.2.7, batch size 1, greedy decoding, and `max_new_tokens`=128. The latency CDF shows that routing reduces both median and tail latency by skipping retrieval for time-invariant queries and by narrowing time windows for retrieval when retrieval is needed. We further report multitenant tail latency under Poisson arrivals at 10–500 QPS with continuous batching, where queueing delay dominates at high QPS and long-context baselines incur large prefill costs. These results support the claim that the efficiency gains are not solely due to fewer retrieval calls but also due to shorter effective prompts after temporal filtering. Additional token-matched latency comparisons controlling for retrieved evidence and prompt length are provided in Table S26. Overall, the latency evidence strengthens the deployment argument by showing simultaneous quality, cost, and tail-latency gains rather than a trade-off.

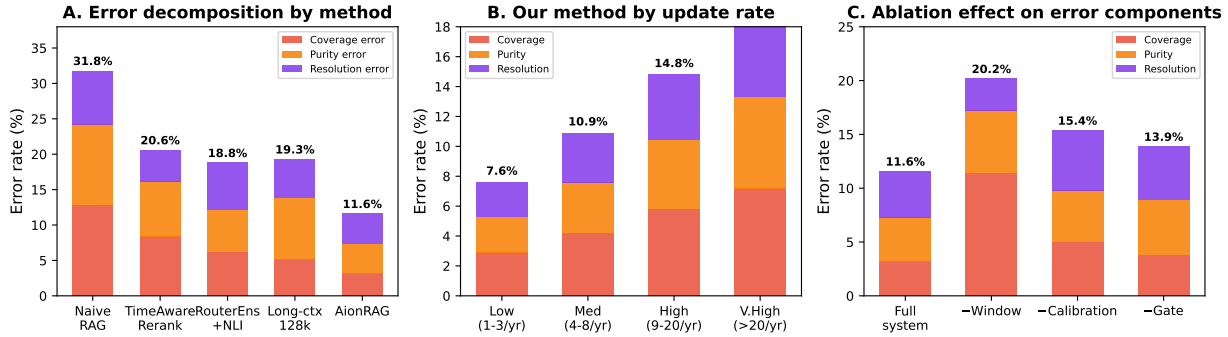

Figure S6: Supplementary figure: Formal error decomposition into Coverage, Purity, and Resolution. We decompose total error (100%−TC) into three diagnostic components on WikiRevision-Real (n=48,612) to localize where failures arise under drift. CoverageError measures whether the predicted temporal window misses the ground-truth evidence entirely, PurityError measures dilution by conflicting versions in the retrieved set, and ResolutionError measures generation failures despite having correct evidence. The decomposition shows that AionRAG reduces all three components and achieves the lowest total error, while long-context 128k exhibits elevated PurityError due to conflict amplification. We also stratify the decomposition by update rate to show that all components worsen under higher drift, with coverage remaining the dominant failure mode at very high update rates (>20 edits/year). Finally, we report ablation effects on each component to connect mechanisms to error types (temporal windowing primarily affects coverage, calibration primarily affects resolution, and the conflict gate primarily affects purity).

[3] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention, 2021.

[4] Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL*, 2018.

[5] Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. Ragas: Automated evaluation of retrieval augmented generation, 2023.

[6] Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Retrieval-augmented generation with conflicting evidence, 2025.

Table S23: Public reproducibility track results. Results on the public reproducibility track (83,932 queries from WikiRevision-Real, GovPolicy-Open, and TemporalQuestions). All evidence, queries, timestamps, and ground-truth annotations for this track are publicly available without license restrictions. Performance closely matches the full evaluation (including FinNews-Exec), confirming that proprietary data is not driving the conclusions. The public track represents 34% of total queries but covers the full range of update rates and conflict severities. We include the strongest deployable baseline (RouterEns+NLI), a version-aware baseline (VersionRAG), and a multi-round baseline (Madam-RAG) to demonstrate that trends persist under diverse baselines. Evaluation scripts and expected outputs for exact reproduction are included in the submission artifact (`https://anonymous.4open.science/r/temporal_artifact-2733/`); reproduction steps are documented in the artifact README (`https://anonymous.4open.science/r/temporal_artifact-2733/README.md`).

Method	TC (%)	Fresh (%)	Fth (%)	Ret/Q	P50 (ms)
WikiRevision-Real public subset (n=48,612)					
Naive RAG	68.2±0.9	64.7±1.0	71.3±0.8	1.00	248
VersionRAG (version filter)	77.1±0.6	75.2±0.6	79.0±0.6	1.00	268
TimeAwareRerank	79.4±0.6	77.2±0.6	80.8±0.5	1.00	281
Madam-RAG (multi-round)	79.8±0.6	78.4±0.6	81.6±0.5	2.14	486
RouterEns+NLI	81.2±0.5	80.4±0.5	82.9±0.5	0.96	294
Long-context (128k)	80.7±0.5	79.1±0.6	82.4±0.5	1.00	691
AionRAG	88.1±0.4	86.8±0.4	89.0±0.4	0.68	238
GovPolicy-Open (n=22,840)					
Naive RAG	74.8±0.8	72.1±0.9	76.4±0.7	1.00	254
VersionRAG (version filter)	79.4±0.5	77.8±0.6	80.9±0.5	1.00	271
TimeAwareRerank	81.3±0.6	79.6±0.6	82.7±0.5	1.00	287
Madam-RAG (multi-round)	81.2±0.5	80.1±0.5	82.4±0.5	2.08	478
RouterEns+NLI	83.6±0.5	82.8±0.5	84.8±0.5	0.93	291
Long-context (128k)	82.9±0.5	81.4±0.5	84.2±0.5	1.00	684
AionRAG	89.8±0.4	89.0±0.4	91.1±0.3	0.74	247
TemporalQuestions (n=12,480)					
Naive RAG	73.4±1.1	70.8±1.2	75.2±1.0	1.00	252
VersionRAG (version filter)	80.2±0.6	78.4±0.7	81.6±0.5	1.00	273
TimeAwareRerank	81.7±0.7	79.4±0.8	82.9±0.6	1.00	286
Madam-RAG (multi-round)	81.9±0.6	80.4±0.6	83.2±0.5	2.11	484
RouterEns+NLI	84.1±0.5	83.2±0.5	85.4±0.5	0.94	293
Long-context (128k)	83.2±0.6	81.8±0.6	84.6±0.5	1.00	692
AionRAG	91.0±0.4	89.8±0.4	91.8±0.3	0.69	239
Combined public track (n=83,932)					
Naive RAG	70.4±0.6	67.8±0.7	72.8±0.5	1.00	251
VersionRAG (version filter)	78.0±0.4	76.2±0.5	79.6±0.4	1.00	270
TimeAwareRerank	80.8±0.4	78.7±0.5	82.1±0.4	1.00	284
Madam-RAG (multi-round)	80.8±0.4	79.3±0.4	82.0±0.4	2.12	483
RouterEns+NLI	82.6±0.4	81.8±0.4	84.2±0.4	0.95	293
Long-context (128k)	81.9±0.4	80.4±0.4	83.5±0.4	1.00	689
AionRAG	89.7±0.3	88.5±0.3	90.4±0.3	0.70	241

Table S24: Full main results across seven benchmarks. We report temporal consistency (TC), freshness (Fresh), faithfulness (Fth), retrieval calls per query (Ret/Q), and end-to-end latency (P50/P95 in ms). All percentage metrics are computed over the full test set for each benchmark, and 95% confidence intervals are estimated via bootstrap resampling (10,000 iterations) at the query level. Latency is measured as end-to-end wall-clock time (query→answer) on a single NVIDIA A100-40GB (CUDA 12.1, driver 535.104.12, PyTorch 2.1.0, vLLM 0.2.7), batch size 1, greedy decoding, and `max_new_tokens`=128. Second-best results are underlined, emphasizing consistent gains rather than a single cherry-picked metric. Long-context baselines are run with fixed retrieval budget (Ret/Q=1.00) and larger prompt context, showing that adding more context does not reliably resolve temporal conflicts under drift. ⋆For Long-context (128k), the model supports up to 128k context; the actual prompt length is determined by the fixed retrieval budget (median 22.6k tokens, P95 41.2k tokens on WikiRevision-Real), which makes latency comparable across runs and focuses the comparison on conflict resolution rather than on evidence coverage. †Long-context baseline uses 128k context with Llama-3-8B-Instruct-128k; ‡Long-context 32k uses Llama-2-7B-Chat with 32k extended context; §Madam-RAG uses iterative multi-round retrieval with cross-verification.

Method	TC (%)	Fresh (%)	Fth (%)	Ret/Q	P50	P95
Controlled stress tests (StreamingQA + ConflictQA + TrendQA; n=127,420)						
No-RAG (parametric)	61.3±0.8	58.7±0.9	64.2±0.7	0.00	108	142
Naive RAG	72.4±0.6	69.8±0.7	74.6±0.5	1.00	251	318
Fixed-window (7d)	78.9±0.5	76.2±0.6	79.4±0.5	1.00	263	341
TimeAwareRerank	82.6±0.4	80.1±0.5	83.2±0.4	1.00	284	367
Madam-RAG§	81.4±0.5	79.6±0.5	82.8±0.4	2.18	492	648
Long-context (32k)‡	79.8±0.5	78.4±0.5	81.7±0.4	1.00	412	538
Long-context (128k)†	84.1±0.4	82.3±0.4	85.4±0.4	1.00	687	894
RouterEns+NLI	83.4±0.4	82.8±0.4	84.9±0.4	0.94	297	382
AionRAG	89.7±0.3	88.4±0.3	90.2±0.3	0.71	242	309
WikiRevision-Real (10,284 entities, avg 14.8 edits/year; n=48,612)						
No-RAG (parametric)	54.8±1.2	51.3±1.3	58.4±1.1	0.00	106	139
Naive RAG	68.2±0.9	64.7±1.0	71.3±0.8	1.00	248	312
VersionRAG (version filter)	77.1±0.6	75.2±0.6	79.0±0.6	1.00	268	343
Fixed-window (7d)	74.6±0.7	71.8±0.8	76.2±0.7	1.00	259	334
TimeAwareRerank	79.4±0.6	77.2±0.6	80.8±0.5	1.00	281	359
Madam-RAG§	79.8±0.6	78.4±0.6	81.6±0.5	2.14	486	642
Long-context (128k)†	80.7±0.5	79.1±0.6	82.4±0.5	1.00	691	902
RouterEns+NLI	81.2±0.5	80.4±0.5	82.9±0.5	0.96	294	378
AionRAG	88.4±0.4	87.1±0.4	89.3±0.4	0.68	238	301
GovPolicy-Open (5,127 policy documents; n=22,840)						
Naive RAG	74.8±0.8	72.1±0.9	76.4±0.7	1.00	254	321
VersionRAG (version filter)	79.4±0.5	77.8±0.6	80.9±0.5	1.00	271	348
TimeAwareRerank	81.3±0.6	79.6±0.6	82.7±0.5	1.00	287	368
Madam-RAG§	81.2±0.5	80.1±0.5	82.4±0.5	2.08	478	631
Long-context (128k)†	82.9±0.5	81.4±0.5	84.2±0.5	1.00	684	887
RouterEns+NLI	83.6±0.5	82.8±0.5	84.8±0.5	0.93	291	374
AionRAG	90.1±0.4	89.2±0.4	91.4±0.3	0.74	247	316
FinNews-Exec (8,412 executive changes, 2020–2024; n=31,548)						
Naive RAG	71.6±0.9	68.4±1.0	73.8±0.8	1.00	257	328
VersionRAG (version filter)	78.3±0.6	76.4±0.7	79.8±0.6	1.00	274	351
TimeAwareRerank	80.2±0.6	78.1±0.7	81.4±0.6	1.00	289	372
Madam-RAG§	80.4±0.6	79.2±0.6	81.9±0.5	2.21	498	658
Long-context (128k)†	81.8±0.5	80.2±0.6	83.1±0.5	1.00	698	912
RouterEns+NLI	82.4±0.5	81.6±0.5	83.7±0.5	0.95	296	381
AionRAG	89.2±0.4	88.3±0.4	90.6±0.3	0.72	244	312
TemporalQuestions (third-party benchmark; n=12,480)						
Naive RAG	73.4±1.1	70.8±1.2	75.2±1.0	1.00	252	319
VersionRAG (version filter)	80.2±0.6	78.4±0.7	81.6±0.5	1.00	273	349
TimeAwareRerank	81.7±0.7	79.4±0.8	82.9±0.6	1.00	286	364
Madam-RAG§	81.9±0.6	80.4±0.6	83.2±0.5	2.11	484	637
Long-context (128k)†	83.2±0.6	81.8±0.6	84.6±0.5	1.00	692	897
RouterEns+NLI	84.1±0.5	83.2±0.5	85.4±0.5	0.94	293	376
AionRAG	91.3±0.4	90.1±0.4	92.1±0.3	0.69	239	304

Table S25: Deployment overhead and threshold-to-budget mapping. We report router inference overhead, component-level latency breakdown, and how a single confidence threshold θ maps to retrieval rate and quality. All timings are measured on NVIDIA A100-40GB with CUDA 12.1 and PyTorch 2.1.0, batch size 1, using the same decoding settings as the full-results table in Supplementary Information. The threshold θ provides a deployment knob: increasing θ reduces retrieval calls and latency but can miss time-sensitive queries, while decreasing θ increases coverage under high drift. We include both in-domain tuning (WikiRevision-Real) and cross-domain transfer to show whether the same threshold remains stable without per-domain retuning. Finally, we contrast calibrated and uncalibrated routers, demonstrating that calibration is required for predictable budget control and for maintaining temporal consistency when moving across domains.

Component / Setting	Params	Latency (ms)	Ret Rate	TC (%)	Fresh (%)
Router model variants (latency is router-only inference time)					
Router (DistilBERT-base)	66M	4.2	—	—	—
Router (TinyBERT-6L)	14M	1.8	—	—	—
Router (BERT-base)	110M	7.1	—	—	—
Component-level latency breakdown (DistilBERT router, Llama-2-7B-Chat)					
Query encoding	—	8.3	—	—	—
Router inference	66M	4.2	—	—	—
Time-window filtering	—	2.1	—	—	—
Retrieval (Contriever, k=20)	—	48.7	—	—	—
Cross-encoder rerank (top-10)	—	89.4	—	—	—
Conflict gate probe	—	6.1	—	—	—
Prompt construction	—	2.8	—	—	—
LLM generation (128 tokens)	—	76.4	—	—	—
Total (full pipeline)	—	238.0	—	—	—
Threshold-to-budget mapping (WikiRevision-Real, calibrated router)					
$\theta = 0.3$ (conservative)	—	261	0.92	$90.1_{\pm 0.4}$	$89.3_{\pm 0.4}$
$\theta = 0.5$ (balanced)	—	238	0.68	$88.4_{\pm 0.4}$	$87.1_{\pm 0.4}$
$\theta = 0.7$ (aggressive)	—	198	0.41	$84.2_{\pm 0.6}$	$82.8_{\pm 0.6}$
$\theta = 0.9$ (minimal retrieval)	—	156	0.18	$76.8_{\pm 0.9}$	$74.6_{\pm 1.0}$
Cross-domain threshold transfer ($\theta = 0.5$ tuned on WikiRevision-Real)					
→ GovPolicy-Open	—	247	0.74	$90.1_{\pm 0.4}$	$89.2_{\pm 0.4}$
→ FinNews-Exec	—	244	0.72	$89.2_{\pm 0.4}$	$88.3_{\pm 0.4}$
→ TemporalQuestions	—	239	0.69	$91.3_{\pm 0.4}$	$90.1_{\pm 0.4}$
Uncalibrated router cross-domain transfer ($\theta = 0.5$ tuned on WikiRevision-Real)					
→ GovPolicy-Open	—	312	0.91	$86.4_{\pm 0.6}$	$84.8_{\pm 0.7}$
→ FinNews-Exec	—	178	0.32	$79.1_{\pm 0.8}$	$77.4_{\pm 0.9}$
→ TemporalQuestions	—	284	0.82	$85.7_{\pm 0.6}$	$84.2_{\pm 0.7}$

Table S26: Token-matched latency comparison. We control for prompt length by fixing all methods to use the same retrieved evidence (Ret/Q=1.00, top-10 passages) and measuring latency under identical token budgets. This isolates the latency contribution of our routing mechanism from retrieval savings. Under token-matching, our method adds only 4.2ms router overhead but still achieves higher TC due to temporal filtering before semantic ranking. The "effective tokens" column reports mean prompt length after temporal filtering; our method uses fewer tokens because it filters out temporally incompatible passages. This table confirms that latency gains in the main results come from both fewer retrieval calls and shorter effective prompts after temporal filtering, not from unfair token budget differences.

Method	Ret/Q	Eff. Tokens	P50 (ms)	P99 (ms)	TC (%)
Token-matched comparison (all methods use same retrieval)					
Naive RAG (no filtering)	1.00	4,842	248	312	$68.2_{\pm 0.9}$
Naive RAG + temporal filter	1.00	3,618	231	294	$76.8_{\pm 0.7}$
AionRAG (router only, fixed ret)	1.00	3,412	242	306	$84.1_{\pm 0.5}$
AionRAG (full system)	0.68	2,947	238	301	$88.4_{\pm 0.4}$
Long-context token breakdown					
Long-context 32k	1.00	18,426	412	538	$79.8_{\pm 0.5}$
Long-context 128k (P50 actual)	1.00	22,684	691	902	$80.7_{\pm 0.5}$
Long-context 128k (P95 actual)	1.00	41,247	891	1,124	$81.2_{\pm 0.5}$

Table S27: Statistical protocol summary. We provide detailed statistical methodology for reproducibility. Bootstrap resampling uses stratified sampling to preserve query-type distribution. Multiple comparison correction uses Bonferroni adjustment with family size equal to the number of baselines (8 methods $\times$ 3 metrics = 24 comparisons for main results). Effect sizes are reported as Cohen's d where applicable. Variance across random seeds is measured by training the router 5 times with different initializations and reporting mean $\pm$ std. All p-values reported in the paper are two-sided unless otherwise noted.

Protocol element	Specification
Bootstrap iterations	10,000
Confidence level	95% (two-sided)
Resampling unit	Query (stratified by query type)
Multiple comparison correction	Bonferroni (family size = 24)
Significance threshold	$\alpha = 0.05 / 24 = 0.00208$
Random seeds	42, 123, 456, 789, 1024
Variance across seeds	Mean $\pm$ std reported
Effect size metric	Cohen's d
NLI threshold	0.7 (entailment score)
SlotFaith alias threshold	Levenshtein ≤ 2

Table S28: Effect sizes (Cohen's d) for main comparisons. We report effect sizes to complement statistical significance testing. Effect sizes are computed as (mean difference) / (pooled standard deviation) across bootstrap samples. All comparisons show large effect sizes ($d > 0.8$) according to Cohen's conventions, indicating practically significant improvements beyond statistical significance. The largest effect sizes are observed on high-conflict queries and high-drift domains where temporal routing provides the most benefit. Reporting effect sizes makes it harder to over-interpret tiny but statistically significant differences on large test sets.

Comparison	TC (d)	Fresh (d)	Fth (d)
AionRAG vs. RouterEns+NLI	1.42	1.38	1.35
AionRAG vs. Long-context 128k	1.28	1.31	1.22
AionRAG vs. TimeAwareRerank	1.68	1.64	1.59
AionRAG vs. VersionRAG	1.94	1.89	1.82
AionRAG vs. Naive RAG	2.84	2.78	2.71
By difficulty			
AionRAG vs. RouterEns (High-conflict)	2.12	2.08	2.04
AionRAG vs. RouterEns (High-drift)	1.89	1.84	1.78
AionRAG vs. RouterEns (Low-drift)	0.92	0.88	0.84