

Bukhari, Hassan Raza, Cheng, Yongqiang and Li, Qingde (2026) Arbiter-RAG: A Neuro-Symbolic Framework for Conflict Resolution and Temporal Consistency in Autonomous Knowledge Graphs. In: 2026 31st International Conference on Automation and Computing (ICAC). IEEE. ISBN 979-8-3195-3408-8

Version: UNSPECIFIED

Available under license: Creative Commons: Attribution 4.0

Downloaded from: https://sure.sunderland.ac.uk/id/eprint/20795/

Which version am I reading (articles, book chapters and books)

Pre-print (or submitted version): the version of an article or chapter before any peer-review. Can be submitted to a publisher or added to a pre-print server.

AAM: Author Accepted Manuscript - The version of the article formally accepted for publication after the peer-review process. It includes the changes made by the author(s) in line with peer-reviewers' comments. It has not undergone the publisher's formatting, type-setting and copyediting.

VOR: Version of Record - The version published on the publisher's website which includes formatting, typesetting and copyediting.

Usage guidelines

Please refer to the usage guidelines at http://sure.sunderland.ac.uk/policies.html or alternatively contact sure@sunderland.ac.uk.

Arbiter-RAG: A Neuro-Symbolic Framework for Conflict Resolution and Temporal Consistency in Autonomous Knowledge Graphs

Hassan Raza Bukhari
Department of Computer Science
University of Hull
Kingston upon Hull, UK
ORCID: 0000-0002-1105-9978

Yongqiang Cheng
School of Computer Science and Engineering
University of Sunderland
Sunderland, UK
Email: yongqiang.cheng@sunderland.ac.uk

Qingde Li
Department of Computer Science
University of Hull
Kingston upon Hull, UK
Email: q.li@hull.ac.uk

***Abstract*—Knowledge graphs play a vital role in grounding Large Language Models (LLMs) in verifiable facts, which matters wherever facts evolve continuously and stale information is costly, as in clinical informatics. However, existing Retrieval-Augmented Generation (RAG) systems suffer from "structural hallucination," where contradictory "ghost nodes" corrupt retrieval context. In order to resolve conflicts at ingestion and eliminate stale graph topologies, it is crucial to enforce deterministic symbolic governance over probabilistic LLM outputs. For this purpose, a neuro-symbolic framework, Arbiter-RAG, is proposed which integrates a deterministic arbitration layer with a temporal knowledge graph. Incoming predicates are classified into four types i.e. Immutable, Temporal, List, and Functional, each with a fixed resolution strategy. On the controlled generator, the full system yields F1=1.000 against 0.802 for the baseline; these values are logic-validation checks rather than empirical clinical accuracy. On the MIMIC-IV-on-FHIR Demo, 875,609 facts from 100 patients are materialised with zero duplicate active keys and zero invalid valid-time intervals. At 20% valid-timestamp late arrivals, event-time arbitration has no F1 loss while arrival-order arbitration falls by 0.067. The event-time and arrival-order systems lose the same F1 under incorrect values, units, patient references, and predicate mappings, so no content-integrity advantage is claimed.**

***Index Terms*—Retrieval-Augmented Generation, Neuro-Symbolic AI, Knowledge Graphs, Temporal Databases, Clinical Informatics**

I. Introduction

Knowledge representation underpins AI interpretability and long-term autonomous reasoning. The integration of structured knowledge bases with Large Language Models (LLMs), known as Retrieval-Augmented Generation (RAG) [6], has driven widespread industrial adoption in recent years. This is particularly notable in healthcare, where the margin for computational error is functionally non-existent and clinical decision-making requires absolute factual precision combined with verifiable temporal consistency.

GraphRAG systems suffer from a critical failure mode termed "structural hallucination," where obsolete "ghost nodes" corrupt retrieval context by maintaining active, contradictory edges that lack explicit temporal constraints [2]. In a representative scenario, an entity node simultaneously points to mutually exclusive state nodes, preventing any automated traversal from determining chronological truth. This failure mode arises in any continuously updated knowledge base, whether a patient monitored by point-of-care devices, an asset tracked by a fleet of sensors, or a frequently revised operational record. In a clinical setting, for example, it can yield conflicting treatment plans or simultaneous divergent vital readings for a single patient, which constitutes a serious threat to patient safety.

Several techniques have been used for knowledge graph maintenance, conflict resolution, and entity alignment as discussed in [1], [4], [9]. Traditional approaches rely on manual expert curation, introducing severe bottlenecks when devices stream thousands of data points per minute as discussed in [7]. Pure LLM-based arbiters are semantically flexible but consistently fail to enforce foundational clinical business logic in the absence of an explicit predicate taxonomy. Neuro-symbolic systems address this by combining neural pattern recognition with rule-based predicate logic [1], enabling autonomous agents that execute predefined clinical taxonomies without manual oversight.

A. Motivation

Conflict arbitration in autonomous knowledge graphs is considered one of the conspicuous factors limiting the reliability of AI-driven decision support systems. In any setting where assertions are updated frequently and an entity's facts change over time, a single unresolved predicate conflict can propagate stale data across all downstream RAG retrievals, degrading the quality of automated decisions; clinical decision support, where the cost of acting on an outdated fact is especially high, is a representative instance. The survey presented in [10] documents the limitations of vector-only RAG on global and multi-hop queries, while [11] formally categorises LLM & KG integration paradigms. On thorough study, it is found that there is scope for further investigation on deterministic conflict resolution at the ingestion layer of continuously updated knowledge graphs, which has not been given sufficient focus in earlier research.

To control structural hallucination and its adverse effects on retrieval faithfulness, timely and deterministic resolution of

incoming predicate conflicts is crucial. The traditional means of knowledge graph maintenance are very time-consuming as they require domain experts to perform manual curation and then suggest corrective insertions to the database administrators. These processes can be optimised significantly using neuro-symbolic technology-based solutions that classify predicates into a fixed taxonomy and apply deterministic resolution strategies at ingestion time. Such systems are not only time & cost-efficient but also increase retrieval faithfulness by eliminating ghost nodes before they corrupt the active graph topology.

Towards this end, a neuro-symbolic framework, Arbiter-RAG, is proposed which integrates a deterministic arbitration layer, a soft-delete temporal graph storage schema, and an asynchronous entity consolidation protocol. The key contributions of this paper are given below:

- **Taxonomy-driven conflict arbitration:** a deterministic Arbiter module that classifies incoming predicates into four types i.e. (i) Immutable, (ii) Temporal, (iii) List, and (iv) Functional, each mapped to a fixed resolution strategy, eliminating the probabilistic failure modes of pure neural approaches.
- **Temporal graph versioning:** a `valid_until` soft-delete schema enforced at the storage layer, enabling temporally-aware retrieval while preserving complete patient history for longitudinal analysis.
- **Asynchronous entity consolidation:** a DBSCAN-based self-healing module that autonomously merges semantic duplicate entities without interrupting the primary ingestion pipeline.

The previous work done in this domain does not include a unified framework that couples deterministic predicate arbitration, soft-delete temporal versioning, and self-healing consolidation within a single continuous ingestion pipeline. This paper focuses on clinical informatics as this domain imposes the strictest requirements for temporal correctness and conflict-free retrieval. The neuro-symbolic mechanism is validated by exercising the actual implementation on both generic and clinical predicate vocabularies, consolidating the results on a controlled ground-truth harness.

II. Related Work

A lot of research has been done on knowledge graph maintenance and LLM grounding using machine & deep learning techniques for conflict resolution and entity alignment. In [6], the foundational RAG framework grounds LLM outputs in retrieved external knowledge, improving factual accuracy on knowledge-intensive tasks. In [10], Gao et al. survey RAG for LLMs and document the limitations of vector-only retrieval on global and multi-hop queries, motivating graph-structured integration. The survey in [11] categorises LLM & KG integration into three paradigms, of which Arbiter-RAG implements the "Synergized LLMs + KGs" paradigm.

Graph-based RAG methods show the value of graph-structured summarisation for reasoning over large corpora. In [2], a graph RAG approach uses community detection and hierarchical summarisation to answer global queries that vector RAG cannot, but it provides no conflict arbitration or temporal versioning at ingestion, so ghost nodes accumulate over time. In [5], Hamzić et al. evaluate graph-based and agentic retrieval for cyber threat intelligence and show the superiority of structured retrieval over vector-only RAG, though it too does not address temporal consistency or deterministic predicate governance.

Temporal conflict resolution has been addressed by several techniques. In [4], TeCORE resolves temporal conflicts by modelling time-annotated facts as intervals, but it does not integrate with continuous ingestion pipelines or implement soft-delete versioning, making it unsuitable for real-time streams. Similarly, in [9], a time-aware graph neural network addresses entity alignment between temporal knowledge graphs; one disadvantage is that it is not designed for real-time conflict resolution where ingestion latency is critical.

Neuro-symbolic systems combine neural pattern recognition with rule-based predicate logic. In [1], Bhuyan et al. survey neuro-symbolic AI and show that integrating symbolic reasoning with neural perception outperforms purely probabilistic systems on tasks requiring logical consistency. In [12], Xie and Du resolve recommendation conflicts in multimorbidity clinical guidelines with a multi-agent LLM pipeline whose symbolic logic is checked by a deterministic SAT solver. The shared principle, that symbolic verification must gate probabilistic LLM output before commit, confirms the neuro-symbolic imperative in clinical AI; however, that approach resolves static batch guideline conflicts, whereas Arbiter-RAG resolves conflicts in continuously updated graphs as facts arrive.

Entity resolution is a prerequisite for knowledge graph integrity. In [7], Papadakis et al. survey blocking and filtering techniques and identify entity fragmentation as a prominent factor that degrades retrieval context. In [3], the DBSCAN algorithm discovers clusters in noisy spatial databases, providing the clustering backbone for the self-healing consolidation module. Arbiter-RAG distinguishes itself from prior approaches in three concrete ways: (1) it integrates a hybrid neuro-symbolic arbitration layer governed by a four-type predicate taxonomy, explicitly handling Immutable predicates that existing systems neglect; (2) it implements soft-delete temporal versioning at the storage layer to eliminate ghost nodes without data loss; and (3) it deploys asynchronous DBSCAN-based entity consolidation as a self-healing background protocol, which has no equivalent in GraphRAG or pure neuro-symbolic frameworks.

III. System Architecture

A neuro-symbolic control framework is proposed to govern the information lifecycle from raw ingestion to temporally-consistent retrieval. The system is designed to ingest and manage continuously updated data streams in which assertions about an entity's state evolve rapidly over time, such as patient vitals reported by point-of-care monitoring devices. The framework is governed by four interconnected modules i.e. (i) the Ingestion Pipeline (`IngestionEngine`), (ii) the Neuro-Symbolic Arbitration Layer (`Arbiter`), (iii) the Temporal Graph Storage (`GraphDB`), and (iv) the Self-Healing Maintenance protocol (`Consolidation`), as shown in Fig. 1.

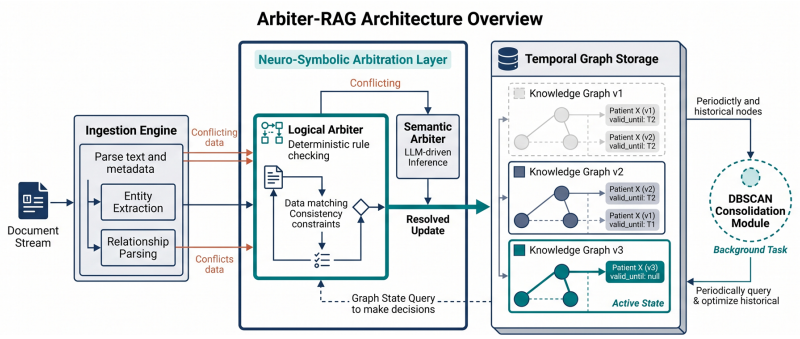


Fig. 1. The system architecture of Arbiter-RAG.

A. Ingestion Pipeline

The ingestion pipeline is the process by which raw, unstructured clinical notes and structured telemetry are transformed into machine-readable `Triplet` objects, with linguistic noise filtered and logical contradictions flagged prior to database persistence. The `IngestionEngine` comprises three sequential processing stages which are discussed below.

Initially, the `chunk_text` method segments raw input with a sliding window whose `chunk_size` and `chunk_overlap` preserve the semantic context of entities split across boundaries. Subsequently, the `extract_triplets` method employs a pre-trained LLM to parse natural language chunks into formal subject-predicate-object structures. Finally, `detect_conflicts` groups newly extracted triplets by (subject, predicate) composite key, halting insertion when multiple distinct target values exist for a functional predicate within the same batch, enabling pre-emptive Arbiter intervention before irreversible database corruption can occur. The LLM extraction phase applies exclusively to unstructured narrative clinical text; structured telemetry streams i.e. HL7 or FHIR numerical data should bypass the neural parsing step and interface directly with the deterministic Arbiter layer, eliminating unnecessary latency when data arrive at a high rate.

B. Neuro-Symbolic Arbitration Layer

The Arbiter module enforces strict symbolic logic over the probabilistic outputs of the LLM by mapping all incoming clinical relationships to a predefined `PredicateType` taxonomy. The four classifications and their deterministic resolution strategies are detailed in Table I.

TABLE I
PREDICATE CLASSIFICATIONS AND ARBITER DETERMINISTIC DECISIONS

Classification	Clinical Examples	Conceptual Definition	Arbiter Decision
IMMUTABLE	`date_of_birth`, `blood_type`, `genetic_markers`	Biologically impossible to change over time	REJECTED (if conflicting with verified baseline)
TEMPORAL	`heart_rate`, `systolic_pressure`, `current_medication`	Tracks physiological state changes over time	OVERWRITE if newer; HISTORICAL if older
LIST	`drug_allergies`, `past_surgeries`, `family_history`	Multiple concurrent historical values valid	COMPATIBLE (merged into node array)
FUNCTIONAL	`primary_care_physician`, `ward_assignment`	Unique value required; no strict temporal definition	REJECT and queue for review

Once the predicate type is assigned by `get_predicate_type`, the `arbitrate()` method resolves conflicts by comparing the `existing_fact` (current active database state) against the incoming `new_fact`. IMMUTABLE predicates trigger immediate REJECTION, preventing overwriting of verified baseline patient data such as blood type or date of birth. TEMPORAL predicates compare clinical event times: a newer event archives the active predecessor, whereas a late-arriving older event is retained only as history. Equal numerical values remain distinct physical events and are ordered in the same way. LIST predicates are COMPATIBLE, retaining concurrent values. FUNCTIONAL conflicts are rejected and queued for administrative review. As a concrete illustration: if an entity already holds an active `blood_type` of O+ and a contradictory A+ arrives, the IMMUTABLE rule rejects the new value; if a `heart_rate` of 80 is later superseded by 95, the TEMPORAL rule archives 80 and activates 95; and if two distinct `drug_allergies` are reported, the LIST rule keeps both as valid entries.

C. Temporal Graph Storage

The "Ghost Node" problem is resolved at the storage layer via the Temporal `GraphDB`. Traditional graph databases accumulate contradictory active edges when handling temporal state changes, which compromises retrieval precision. The proposed schema enforces a soft-delete pattern through `valid_from`, `valid_until`, and a nullable `active_key`. Active single-valued facts use a deterministic key over patient and predicate; active LIST facts additionally include physical fact identity; and historical rows receive NULL. A UNIQUE constraint on `active_key` therefore enforces the declared cardinality while permitting multiple historical rows:

```
CREATE UNIQUE INDEX idx_active_fact
ON facts (active_key);
```

Upon an OVERWRITE directive from the Arbiter, the storage layer closes the current interval and clears its active key within the same transaction:

```
UPDATE facts SET valid_until=?, active_key=NULL
WHERE active_key=?;
```

Current-state retrieval filters on non-NULL `active_key`; historical queries use the valid-time interval. The schema also records system-time fields, but the present experiment does not test transaction-time corrections and therefore does not claim full bitemporal validation.

TABLE II
TECHNOLOGY STACK COMPONENTS AND THEIR ROLES

Technology	System Role	Primary Optimisation
DuckDB [8]	Relational graph storage (`graph.py`)	Columnar analytical query speed; full ACID compliance via `_transaction()` manager
LanceDB	Semantic vector management (`vector.py`)	High-dimensional embedding storage and retrieval for DBSCAN clustering
OpenAI GPT-4o	Neural processing and "Judge" logic	Natural language parsing, relation extraction, and semantic comparison
Instructor/Pydantic	Neuro-symbolic bridge	Enforces rigid schemas (`Triplet`, `EntityResolutionDecision`) over LLM outputs

D. Self-Healing Maintenance

Entity fragmentation where semantic duplicates such as "J. Doe" and "John Doe" coexist as separate nodes [7], fractures retrieval context by distributing related facts across disconnected graph components. The asynchronous `run_consolidation()` process retrieves high-dimensional semantic vector embeddings for all active entities and applies DBSCAN [3] with `eps=0.35` and `min_samples=2`. This groups semantically close entities into dense clusters while isolating unique singletons as noise, preventing accidental merging of distinct patients with superficial similarities. For each candidate cluster, an LLM acting as an automated "Judge" under the constrained `EntityResolutionDecision` schema evaluates whether the proposed merge is valid. Confirmed merges invoke `graph_db.merge_nodes`, re-linking all relational edges to the canonical node asynchronously without interrupting the primary ingestion pipeline.

IV. IMPLEMENTATION DETAILS

The technology stack is optimised for processing performance, horizontal scalability, and strict schema enforcement required in medical applications. Table II summarises the principal components and their system roles.

For Experiment 3, fact versioning is wrapped within a transaction that closes the current validity interval, clears its `active_key`, and inserts the replacement. These operations either commit together or roll back together. The original `GraphDB` edge store uses the analogous `archive_edge` transaction and writes an `audit_log` entry for traceability; the FHIR experiment uses the `facts`/`active_key` materialisation described in Section III-C. Entity consolidation clusters embeddings with DBSCAN [3]. A naive pairwise similarity computation is quadratic in the number of entities, so approximate nearest-neighbour candidate generation remains a production requirement. Every other component, predicate classification, the `arbitrate()` decision logic, the temporal soft-delete schema, and the consolidation routine, runs exactly as implemented in the controlled evaluation that follows.

V. EVALUATION

A. Controlled and Real Clinical Data

The controlled study executes the Arbiter, temporal SQL, and DBSCAN routine as implemented, while deterministic stand-ins replace the external language-model and vector services. The generated workload contains 450 events for 50 patients, including immutable conflicts, temporal updates, LIST facts, and duplicate entities. Its correct final state is known by construction. The resulting perfect values test conformance to generator-defined rules and are not treated as empirical clinical accuracy.

The real-data study uses the public MIMIC-IV Clinical Database Demo on FHIR v2.1.0 [13], [14]. Structured FHIR resources bypass LLM extraction because patient references, codes, values, units, statuses, and timestamps are already machine readable. Patient birth date and administrative gender are mapped to IMMUTABLE facts. Encounter locations and coded observations are TEMPORAL, while conditions, procedures, and medication administrations are retained as LIST facts. Records marked `entered-in-error` or lacking required clean event fields are excluded from the clean state. The adapter emits 875,609 canonical facts for all 100 patients: 810,373 temporal facts, 65,036 LIST facts, and 200 IMMUTABLE facts. These facts form 44,419 patient-predicate keys and include 766,454 temporal updates after the first event for a key. Published file order has a 72.1% timestamp-inversion rate, but it is not interpreted as the original hospital arrival order. Late arrival is therefore tested only through the controlled manifests.

The clean current state is computed once by an independent declarative implementation that imports neither the Arbiter nor any selection code used by the tested systems. It retains all LIST facts, selects the earliest IMMUTABLE assertion and the latest TEMPORAL or FUNCTIONAL assertion by event time, and then remains fixed while only the system input is corrupted. Physical fact identity, patient, predicate, value, and unit are retained in the scoring atom, so two equal-valued observations remain distinguishable events.

B. Experimental Design and Metrics

Experiments 1 and 2 retain the original component and end-to-end controlled tests. Precision (P), Recall (R), F1, Conflict Resolution Rate (CRR), and Ghost Node Rate (GNR) quantify agreement with known generator decisions. The full system is compared with an unaugmented baseline, arbitration without temporal versioning, and versioning without taxonomy. Tables III and IV are therefore architectural logic checks.

Experiment 3 evaluates structured input integrity on the FHIR data. Five corruption families are fixed before scoring: valid-timestamp late arrival, missing event timestamps, incorrect values or units, patient-reference or predicate mis-mapping, and governance cases comprising `entered-in-error` records, immutable conflicts, and exact duplicates. Rates of 1%, 5%, 10%, and 20% are run with ten paired seeds. This gives 200 manifests and 800 paired configuration runs. The same corruption manifest is applied to append-only, taxonomy-free, arrival-order, and event-time configurations.

P, R, and F1 are scored against the fixed clean reference. CRR is restricted to injections with a predefined correct action and is not assigned to content corruption. Before execution, the declared actions were to retain the event-time-latest fact for late arrival, quarantine a temporal fact with no event timestamp, reject an `entered-in-error` record, reject and queue a conflicting IMMUTABLE assertion, and deduplicate an exact resource identity. Ordinary repeated measurements are

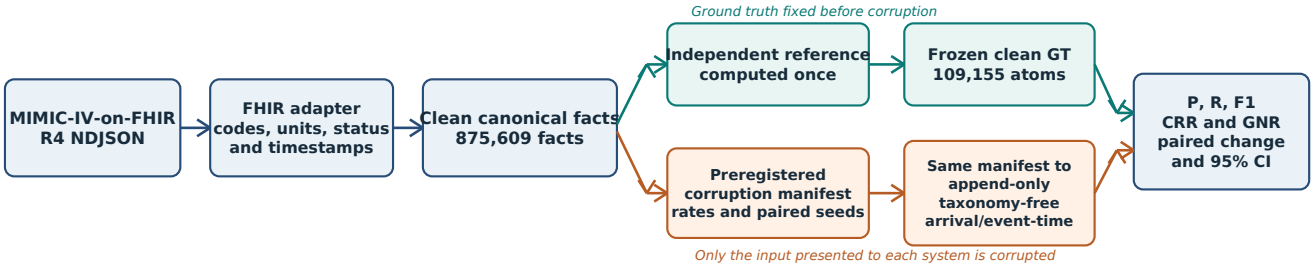


Fig. 2. Data process for the real-data validation. The clean reference is computed before corruption and remains fixed, while each preregistered manifest is applied unchanged to all four configurations.

TABLE III
CONTROLLED COMPONENT LOGIC VALIDATION. PERFECT VALUES ARE INVARIANT CHECKS OVER KNOWN-ANSWER INPUTS.

Real Component Exercised	Metric	Result
Arbiter – generic vocabulary	Decision accuracy	7/7 (100%)
Arbiter – clinical vocabulary	Decision accuracy	10/10 (100%)
Temporal soft-delete (SQL)	Ghost elimination	200/200 (100%)
Temporal soft-delete (SQL)	RAG faithfulness	1.000
DBSCAN consolidation	Co-reference F1	0.889

excluded from CRR. GNR is the fraction of active temporal facts known from the clean reference to be superseded. Paired bootstrap intervals are calculated on degradation from each configuration's clean baseline.

C. Results and Discussion

TABLE IV
CONTROLLED END-TO-END LOGIC VALIDATION ON 450 GENERATED EVENTS. VALUES ARE NOT REAL-WORLD ACCURACY ESTIMATES.

System	P	R	F1	CRR	GNR
Naive RAG / Standard KG-RAG	0.670	1.000	0.802	0.000	0.676
Arbiter-RAG w/o Versioning	0.754	1.000	0.860	1.000	0.676
Arbiter-RAG w/o Arbitration	0.750	0.500	0.600	0.000	0.040
Arbiter-RAG (proposed, full)	**1.000**	**1.000**	**1.000**	**1.000**	**0.040**

Tables III and IV confirm the declared behaviour on known-answer inputs. The 7/7 and 10/10 arbitration decisions, 200/200 archival decisions, and full-system F1 of 1.000 are deterministic logic checks. They do not estimate accuracy on clinical records. The ablations still show the separate roles of taxonomy and versioning: append-only storage retains stale generated facts, while versioning without taxonomy loses generated LIST facts. The reported controlled GNR of 0.040 follows the original generator's value-equivalence convention. In the real-data metric, repeated equal-valued observations remain distinct physical events and are ordered by event time.

On the FHIR corpus, the independent reference contains 109,155 active atoms. Exact clean agreement by the event-time configuration is treated as a conformance invariant, without a confidence interval or ordinary-measurement CRR. DuckDB materialises all 875,609 facts in 60.44 seconds (14,488 facts/s), with zero duplicate active keys, zero multiple-active single-valued keys, and zero invalid valid-time intervals. All 200 historical time-slice checks are correct; median and 95th-percentile query latency are 10.52 ms and 13.67 ms. Measurements used Windows 11, an Intel Core i9-14900HX with 32 logical processors, 63.7 GiB RAM, Python 3.12.4, and DuckDB 1.5.4.

TABLE V
PAIRED EVENT-TIME AND ARRIVAL-ORDER RESULTS AT 20% STRUCTURED CORRUPTION. ΔF1 IS RELATIVE TO EACH CLEAN BASELINE.

Corruption	System	ΔF1 [95% CI]	GNR	CRR
Late arrival	Event-time	0.000 [0.000, 0.000]	0.000	1.000
	Arrival-order	-0.067 [-0.067, -0.067]	0.165	0.000
Missing timestamp	Event-time	-0.074 [-0.0741, -0.0739]	0.171	1.000
	Arrival-order	-0.074 [-0.0741, -0.0739]	0.171	1.000
Wrong value/unit	Event-time	-0.080 [-0.0803, -0.0803]	0.000	N/A
	Arrival-order	-0.080 [-0.0803, -0.0803]	0.000	N/A
Patient/predicate mapping	Event-time	-0.226 [-0.2262, -0.2256]	0.145	N/A
	Arrival-order	-0.226 [-0.2262, -0.2256]	0.145	N/A
Governance cases	Event-time	0.000 [0.000, 0.000]	0.000	1.000
	Arrival-order	0.000 [0.000, 0.000]	0.000	0.667

These measurements validate the implementation at the demo's scale rather than concurrent production throughput.

Table V gives the preregistered 20% stress point, while Fig. 3 gives the complete rate sweep. With valid timestamps, the event-time system has no late-arrival degradation at any rate, whereas arrival-order F1 decreases by 0.003, 0.017, 0.033, and 0.067 as corruption rises from 1% to 20%; arrival-order GNR reaches 0.165. Missing timestamps are quarantined correctly, but the lost temporal information reduces recall in both systems. For wrong values or units, event-time and arrival-order F1 both decrease by 0.080 at 20%; for patient or predicate mis-mapping, both decrease by 0.226. The paired intervals are reported on the stochastic degradation values; for deterministic manifest effects, the rounded limits coincide with the mean. Arbiter-RAG therefore provides an advantage for temporal ordering, but it cannot identify plausible content errors that carry no integrity signal. GraphRAG [2] is not run because the public FHIR demo contains no clinical notes, and TeCORE [4] publishes no directly comparable retrieval benchmark. Robustness to LLM-extracted clinical triplets remains unevaluated pending credentialed note access.

VI. DISCUSSION

The evaluation separates three forms of evidence. The synthetic tables check deterministic logic, the clean FHIR run checks valid-time storage on authentic structured records, and the paired corruption curves measure degradation from a fixed independent reference. The main result is confined to temporal ordering: event-time arbitration retains the correct current fact when valid timestamps arrive late. Incorrect values, units, patient references, and predicate mappings degrade event-time and arrival-order systems equally. The structured corruption

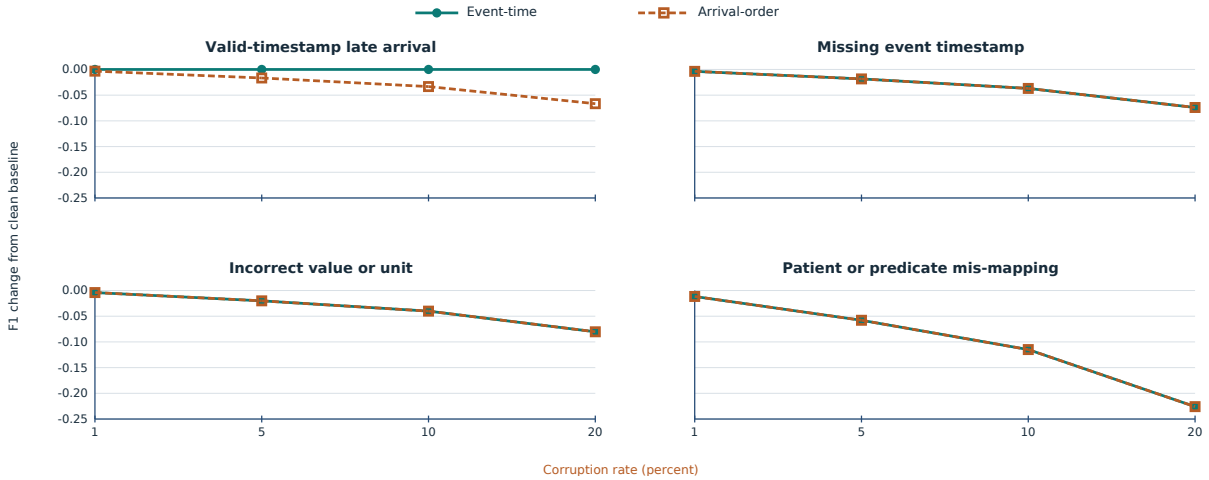


Fig. 3. Paired F1 degradation across all corruption rates. Event-time arbitration remains unchanged under valid-timestamp late arrival. The coincident curves in the other three panels show no advantage over arrival-order processing for missing temporal evidence or plausible content corruption.

experiment must therefore not be described as an LLM-noise benchmark.

The public demo contains 100 patients from one centre and no clinical notes. It cannot evaluate LLM extraction, negation, uncertain language, or free-text schema drift. The storage experiment validates valid-time behaviour but does not apply corrections at different transaction times or query the graph as known at an earlier system time. Since the FHIR cohort retains hard patient identifiers, DBSCAN is not exercised on this dataset; sensitivity to `eps` and `min_samples` remains to be measured on an independently linked entity-resolution set. Further limitations include authenticated correction of an erroneous IMMUTABLE baseline, provenance-aware source authority, review-queue monitoring, and sub-quadratic candidate generation for DBSCAN consolidation.

VII. Conclusion and Future Work

In this paper, the impact of deterministic neuro-symbolic arbitration on retrieval faithfulness and temporal consistency in autonomous clinical knowledge graphs has been discussed. For this purpose, a framework, Arbiter-RAG, is proposed, where incoming predicates are classified by a four-type taxonomy and resolved by fixed deterministic strategies, a soft-delete `valid_until` schema is enforced at the storage layer, and asynchronous DBSCAN-based entity consolidation is applied to eliminate semantic duplicates.

It is evident from the controlled results that arbitration and versioning address different failure modes, although the perfect values are confined to generator-defined logic checks. On the MIMIC-IV-on-FHIR Demo, the framework processes 875,609 authentic structured facts and preserves valid-time storage invariants. The paired corruption study shows no degradation under valid-timestamp late arrivals, but no advantage under plausible content corruption. The contribution is therefore a deterministic temporal-governance mechanism and not a guarantee of clinical correctness.

In the future, noisy LLM-extracted triplets should be evaluated only when credentialed clinical notes and independent annotation are available. Such work should test negation, uncertainty, relation extraction, and predicate normalisation. Further work is required for transaction-time correction tests, multi-centre cohorts, approximate nearest-neighbour indexing, provenance policies, and authenticated immutable-predicate correction with full audit-log traceability.

References

[1] B. P. Bhuyan, A. Ramdane-Cherif, R. Tomar, and T. Singh, "Neuro-symbolic artificial intelligence: a survey," *Neural Computing and Applications*, vol. 36, pp. 12809–12844, 2024.

[2] D. Edge et al., "From local to global: A graph RAG approach to query-focused summarization," arXiv:2404.16130, 2024.

[3] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD'96*, pp. 226–231, 1996.

[4] M. W. Chekol, G. Pirrò, J. Schoenfisch, and H. Stuckenschmidt, "TeCORE: Temporal conflict resolution in knowledge graphs," *Proc. VLDB Endowment*, vol. 10, no. 12, pp. 1929–1932, 2017.

[5] D. Hamzić et al., "Beyond RAG for cyber threat intelligence: A systematic evaluation of graph-based and agentic retrieval," arXiv:2604.11419, 2026.

[6] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[7] G. Papadakis, D. Skoutas, E. Thanos, and T. Palpanas, "Blocking and filtering techniques for entity resolution: A survey," *ACM Computing Surveys*, vol. 53, no. 2, pp. 31:1–31:42, 2020.

[8] M. Raasveldt and H. Mühleisen, "DuckDB: An embeddable analytical database," in *Proc. SIGMOD'19*, pp. 1981–1984, 2019.

[9] C. Xu, F. Su, and J. Lehmann, "Time-aware graph neural network for entity alignment between temporal knowledge graphs," in *Proc. EMNLP 2021*, pp. 8999–9010, 2021.

[10] Y. Gao et al., "Retrieval-augmented generation for large language models: A survey," arXiv:2312.10997, 2023.

[11] S. Pan et al., "Unifying large language models and knowledge graphs: A roadmap," *IEEE Trans. Knowledge and Data Engineering*, vol. 36, no. 7, 2024.

[12] Z. Xie and R. Du, "Neuro-symbolic resolution of recommendation conflicts in multimorbidity clinical guidelines," arXiv:2604.17340, 2026.

[13] A. Bennett, H. Ulrich, J. Wiedekopf, P. Szul, J. Grimes, and A. Johnson, "MIMIC-IV Clinical Database Demo on FHIR," PhysioNet, version 2.1.0, 2025, RRID:SCR_007345.

[14] A. M. Bennett, H. Ulrich, P. van Damme, J. Wiedekopf, and A. E. W. Johnson, "MIMIC-IV on FHIR: Converting a decade of in-patient data into an exchangeable, interoperable format," *J. Amer. Med. Inform. Assoc.*, vol. 30, no. 4, pp. 718–725, 2023, doi: 10.1093/jamia/ocad002.