

Typed Evidence Fusion for Cross-Modal Enterprise Question Answering: Preserving Provenance, Currency and Cardinality across Retrieval Modalities

Vishnu Vardhan Reddy Yeruva

Dept. of Enterprise AI, Hillsborough, NJ, US

ABSTRACT

Enterprises have accumulated a stack of partial solutions to the question-answering problem: semantic search over documents, knowledge graphs over master data, text-to-SQL over the warehouse, and increasingly a chat interface over each of them separately. Each works within its competence and fails outside it, and the failures are silent. Ask a document-RAG system for a total and it will find a sentence containing a number. Ask a text-to-SQL system why a figure moved and it will produce a query, not an explanation. The user cannot see which system they are talking to, so they cannot calibrate.

This paper proposes a unified architecture in which retrieval modality is a *routed decision* rather than a deployment choice, and in which heterogeneous evidence (passages, graph paths, query result sets, computed aggregates) is carried in a typed representation that permits a single verification pass over all of it. The organising abstraction is a shared blackboard over which a planner, four specialist retrieval agents, a critic, and a synthesiser coordinate. We contribute: a typed evidence algebra with composition rules, which makes cross-modal evidence combinable rather than merely concatenable; a coverage-driven termination criterion that decides when to stop retrieving on the basis of what the question requires rather than a fixed step budget; a critic formulation that checks for the specific failure mode of cross-modal inconsistency, where a passage and a query result disagree; and an analysis of the cost-capability curve showing where marginal agent steps stop paying.

We evaluate on EntQA-Bench, an executable benchmark released with this paper: a real SQLite warehouse of seven joined tables, 1,332 documents indexed with BM25, a 159-node entity graph, and 1,310 questions in seven classes by required modality, with gold answers computed from the sources. The results are measured. Flattening evidence to text costs accuracy: prose answers 87.6% of questions and a concatenated agent transcript 54.2%, against 100% for typed evidence, and on questions requiring all three modalities the transcript representation answers none. One claim from an earlier simulated draft does not survive. Typed evidence gives no accuracy advantage over plain JSON carrying values and confidences: both answer 100%, and the earlier figure of 8.3 accuracy points is withdrawn. What types buy is verification capability, which the untyped representations lack entirely: source disagreement is detected on 57.2% of injected conflicts against 0.0%, cardinality is reported correctly on 82.5% against 0.0%, and the weakest currency among combined sources is propagated on 100% against 0.0%. The revised claim is therefore narrower and, we think, more useful: types are what allow a cross-modal answer to be checked, not what make it correct.

INDEX TERMS: MULTI-AGENT SYSTEMS, BLACKBOARD ARCHITECTURE, KNOWLEDGE GRAPHS, SEMANTIC SEARCH, TEXT-TO-SQL, ENTERPRISE DATA INTEGRATION, LARGE LANGUAGE MODELS, EVIDENCE FUSION, ORCHESTRATION.

I. INTRODUCTION

The question “why did gross margin in the Iberia region fall two points last quarter?” is not answerable by any single retrieval mechanism, and it is a fair representation of what enterprise users actually ask. It requires an aggregate from the warehouse, a decomposition of that aggregate by product and customer, a join across master data to identify which contracts changed, retrieval of the contract documents that changed, and a synthesis that attributes the movement. A document retriever contributes one of five steps. A SQL generator contributes two. Neither knows the others exist.

The prevailing response has been to add tools to a single agent: give a language model a search tool, a SQL tool, and a graph tool, and let it decide. This works better than any single mechanism and it has two structural weaknesses that become apparent at enterprise scale.

The first is that evidence from different tools is incommensurable in the context window. A retrieved passage and a query result set both arrive as text, and the model has no principled basis for reasoning about their relative authority, their currency, or their mutual consistency. When a passage in a board deck says margin fell 2.1 points and a warehouse query returns 1.8, the model typically picks one, usually the more recently mentioned, and proceeds. The disagreement is exactly the signal a human analyst would investigate.

The second is termination. A single agent with a tool loop stops when it believes it has enough, and that belief is uncalibrated. In practice systems either stop too early on hard questions or loop until a step budget expires on easy ones. What is missing is a notion of *coverage*: an explicit account of what the question requires and which requirements remain unmet.

We propose an architecture that addresses both by making evidence typed and coordination explicit. Evidence carries its modality, provenance, currency, and cardinality, and composition is governed by rules rather than by concatenation. Coordination happens over a shared blackboard in the tradition of Hearsay-II [1], [2], with a planner that decomposes a query into a typed task graph, specialist agents that contribute evidence of specific types, a critic that checks coverage and cross-modal consistency, and a synthesiser that composes the answer. Termination is a coverage condition, not a budget.

The relationship to the wider agent literature is worth stating. Tool-using agents [3], chain-of-thought prompting [4], self-reflective agents [5], and multi-agent frameworks [6], [7] have established the components. What they have not addressed, because they were largely developed in open-domain settings, is the specific problem of heterogeneous enterprise evidence with differing authority and currency. A web page and another web page are commensurable; a governed financial fact and a slide in a presentation are not.

Sections II–VI cover background, problem, method, architecture, and formalism; Section VII the evaluation framework; Section VIII simulated results; Sections IX–XIII discussion, limits, applications and conclusions.

II. BACKGROUND AND LITERATURE REVIEW

A. Blackboard Architectures

The blackboard model originates with Hearsay-II [1] and was consolidated by Nii [2]: independent kno-

wledge sources contribute hypotheses to a shared structure, and a control component decides which to activate. The model fell out of favour partly because writing the knowledge sources was laborious. Language models remove that obstacle, a specialist agent is now a prompt and a tool binding, which we think makes the architecture newly relevant rather than merely historically interesting. Agent-oriented software engineering [8] supplies the coordination vocabulary.

B. Retrieval Modalities

Dense retrieval [9], sparse retrieval [10], and approximate nearest-neighbour indices [11] underpin semantic search; the RAG formulation [12] and its survey literature [13] define the standard pipeline. Graph-structured retrieval [14] supplies multi-hop capability over entity relationships, building on the knowledge-graph foundations surveyed in [15]. Structured querying of relational stores, the text-to-SQL task, has its own benchmarks and literature [16], [17], with performance on realistic enterprise schemas substantially below the figures reported on academic benchmarks, a gap documented by the BIRD benchmark's emphasis on dirty, large, real schemas [17].

Each modality has a competence envelope. Table I summarises them, and the point of the table is the diagonal: every modality is strong somewhere and unusable elsewhere, which is the argument for routing rather than for choosing.

Table I. Competence envelopes of enterprise retrieval modalities.

Query class	Semantic search	Knowledge graph	Text-to-SQL	Analytic tools
Definitional lookup	Strong	Moderate	Weak	None
Entity attribute	Moderate	Strong	Strong	None
Multi-hop relationship	Weak	Strong	Moderate	None
Exact aggregate	None	Weak	Strong	Strong
Trend / variance	Weak	None	Moderate	Strong
Policy / rationale	Strong	Weak	None	None
Narrative explanation	Moderate	Moderate	None	Weak

C. Multi-Agent LLM Systems

ReAct [3] interleaves reasoning and action; Reflexion [5] adds verbal self-critique; MetaGPT [6] and AutoGen [7] structure multiple agents with roles and conversation protocols. These are close relatives of the present proposal. The difference we would emphasise is representational: in conversation-based multi-agent frameworks, agents exchange natural language, and typed evidence degrades to prose the moment it crosses an agent boundary. The blackboard here is typed, so a query result set remains a result set with a schema, a row count, and a currency timestamp, and the critic can compare it against a passage without either having been flattened.

D. Verification

Faithfulness evaluation for RAG [18] and hallucination taxonomies [19] address single-modality grounding. Cross-modal inconsistency, a numeric claim supported by a passage and contradicted by a query, is a distinct failure that faithfulness scoring does not detect, because the claim is faithful to the passage. Section IV-E treats it directly.

E. What Is New Here

Against the closest prior work: ReAct [3], Reflexion [5], MetaGPT [6] and AutoGen [7] coordinate agents by exchanging natural language, so typed evidence degrades to prose at every agent boundary;

GraphRAG [14] and the RAG literature [12], [13] are single-modality; text-to-SQL research [16], [17] optimises query generation in isolation. The single contribution this paper asks to be judged on is the typed evidence representation and its composition rules: preserving modality, provenance, currency and cardinality across agent boundaries, resolving conflicts by an explicit authority ordering rather than by recency, and propagating the weakest guarantee rather than the strongest. The ablation in Section VIII-B is the relevant evidence: removing types costs more than removing any single agent and collapses the system's ability to know that it is incomplete. The blackboard, the specialist agents and the coverage criterion are the implementation that makes typed evidence testable, not separate claims.

This paper is one of a group of related studies. Its scope is combining evidence across retrieval modalities. Provider selection within a step is treated in the companion routing work, retrieval mechanics over transactional data in the companion vector-graph work, and access control in the companion governed-retrieval work; this paper assumes all three are solved and asks only what happens at the seams.

III. RESEARCH PROBLEM AND OBJECTIVES

A. Problem Statement

Given a natural-language query q over an enterprise estate comprising a document corpus $\mathcal{D}$, a property graph $\mathcal{G}$, a relational warehouse $\mathcal{W}$, and an analytic tool set $\mathcal{T}$, produce an answer a with typed provenance P such that: every claim in a is supported by evidence in P ; no two elements of P used to support a claim are mutually inconsistent beyond a tolerance; the question's information requirements are covered or the shortfall is declared; and the process terminates within a cost bound.

B. Objectives

- O1.** Define a typed evidence representation with composition rules that preserve modality, provenance, currency, and cardinality across agent boundaries.
- O2.** Specify a planner that decomposes queries into typed subgoals and routes each to the modality competent for it.
- O3.** Define coverage-based termination so that retrieval effort tracks question difficulty.
- O4.** Specify a critic that detects cross-modal inconsistency and coverage shortfall, and triggers replanning.
- O5.** Characterise the cost-capability curve and identify the operating point beyond which additional agent steps do not pay.

C. Research Questions

- RQ1.** How does task success vary by query class across single-agent, monolithic-RAG, and unified multi-agent architectures?
- RQ2.** Does coverage-based termination allocate effort more efficiently than a fixed step budget?
- RQ3.** How frequently does cross-modal inconsistency arise, and what fraction is detectable by the critic?

IV. PROPOSED METHODOLOGY

A. Typed Evidence

An evidence item is a tuple

$$e = \langle \text{type}, \text{content}, \text{prov}, \text{cur}, \text{card}, \text{conf} \rangle,$$

with type in {passage, graph-path, tuple-set, scalar, series, computed}, provenance identifying the source object and version, currency a timestamp of the underlying data, cardinality distinguishing a single fact

from a complete set from a sample, and confidence a calibrated score. Cardinality is the field that most often gets lost in prose-based systems and matters most: an answer that treats a sampled tuple-set as complete is wrong in a way no amount of grounding checking will catch.

B. Evidence Algebra

Composition is governed by rules rather than by concatenation:

- **Corroboration.** Two items of different type supporting the same claim raise confidence super-additively, since their error modes are independent: $conf(e_1 \oplus e_2) = 1 - (1 - c_1)(1 - c_2)$.
- **Conflict.** Two items supporting contradictory claims produce a conflict marker rather than a silent selection; resolution is by authority ordering (governed warehouse fact > graph fact > document passage) with the conflict recorded in provenance.
- **Refinement.** A scalar derived from a tuple-set inherits the set's cardinality and currency, so a total computed from a sample is marked as an estimate.
- **Staleness.** Composing items of differing currency yields the *oldest* currency, which propagates the weakest guarantee rather than the strongest.

These rules are simple and, we would argue, the minimum needed to keep a multi-modal answer honest.

C. Planning and Routing

The planner decomposes q into a directed acyclic graph of subgoals, each annotated with a required evidence type and a dependency set. Routing follows from the type: a subgoal requiring an exact aggregate goes to the structured-query agent; one requiring rationale goes to semantic search; one requiring a relationship chain goes to the graph agent. Where a subgoal's type is ambiguous, it is dispatched to two agents and resolved by the corroboration or conflict rules above, an intentional redundancy that costs tokens and buys detection of exactly the disagreements we care about.

D. Coverage-Based Termination

Each subgoal carries a satisfaction predicate. Coverage is

$$Cov = \frac{\sum_{g \in \mathcal{S}} \omega_g \mathbb{1}[\text{satisfied}(g)]}{\sum_{g \in \mathcal{S}} \omega_g},$$

with weights ω_g reflecting each subgoal's contribution to the answer. Execution terminates when $Cov \geq \theta$, when marginal coverage per step falls below a threshold, or when the cost bound binds, in which case the answer declares its shortfall explicitly rather than presenting itself as complete. That declaration is the part we consider non-negotiable: a partial answer labelled partial is useful, and a partial answer labelled complete is worse than none.

E. Critic

The critic performs three checks. *Coverage*: are all subgoals satisfied, and if not which. *Cross-modal consistency*: for each numeric or factual claim supported by more than one evidence type, do the supports agree within tolerance. *Citation support*: is each claim in the draft traceable to blackboard evidence, in the manner of faithfulness checking [18], extended across evidence types. Failure triggers replanning with the deficiency as input, bounded by the cost budget.

F. Synthesis

The synthesiser composes the answer from satisfied subgoals, attaching per-claim provenance and surfacing conflicts and staleness explicitly rather than smoothing them away. Where the evidence algebra produced a conflict marker, the answer states both values with their sources and the authority ordering applied.

```
from dataclasses import dataclass, field
from typing import Any, Literal
```

```
EvidenceType = Literal["passage", "graph_path", "tuple_set",
                        "scalar", "series", "computed"]
AUTHORITY = {"tuple_set": 3, "scalar": 3, "computed": 3,
              "graph_path": 2, "passage": 1, "series": 3}
```

```
@dataclass
class Evidence:
    type: EvidenceType
    content: Any
    prov: str          # source object + version
    currency: str       # ISO timestamp of underlying data
    cardinality: Literal["single", "complete", "sample"]
    conf: float
```

```
@dataclass
class Claim:
    text: str
    supports: list = field(default_factory=list)
```

```
def compose(claim: Claim, tolerance: float = 0.02):
    """Apply the evidence algebra to a claim's supporting items."""
    numeric = [(e, e.content) for e in claim.supports
                if isinstance(e.content, (int, float))]
    conflict = None
    if len(numeric) > 1:
        vals = [v for _, v in numeric]
        lo, hi = min(vals), max(vals)
        if hi - lo > tolerance * max(abs(hi), 1e-9):
            # authority ordering decides; the disagreement is retained
            winner = max(numeric, key=lambda t: AUTHORITY[t[0].type])
            conflict = {"values": numeric, "resolved_to": winner}

    conf = 1.0
    for e in claim.supports:          # corroboration
        conf *= (1.0 - e.conf)
    conf = 1.0 - conf

    currency = min((e.currency for e in claim.supports), default=None)
    card = ("sample" if any(e.cardinality == "sample" for e in claim.supports)
            else "complete" if all(e.cardinality == "complete"
```

```

    for e in claim.supports)
    else "single")
return {"confidence": conf, "currency": currency,
        "cardinality": card, "conflict": conflict}

```

V. SYSTEM ARCHITECTURE

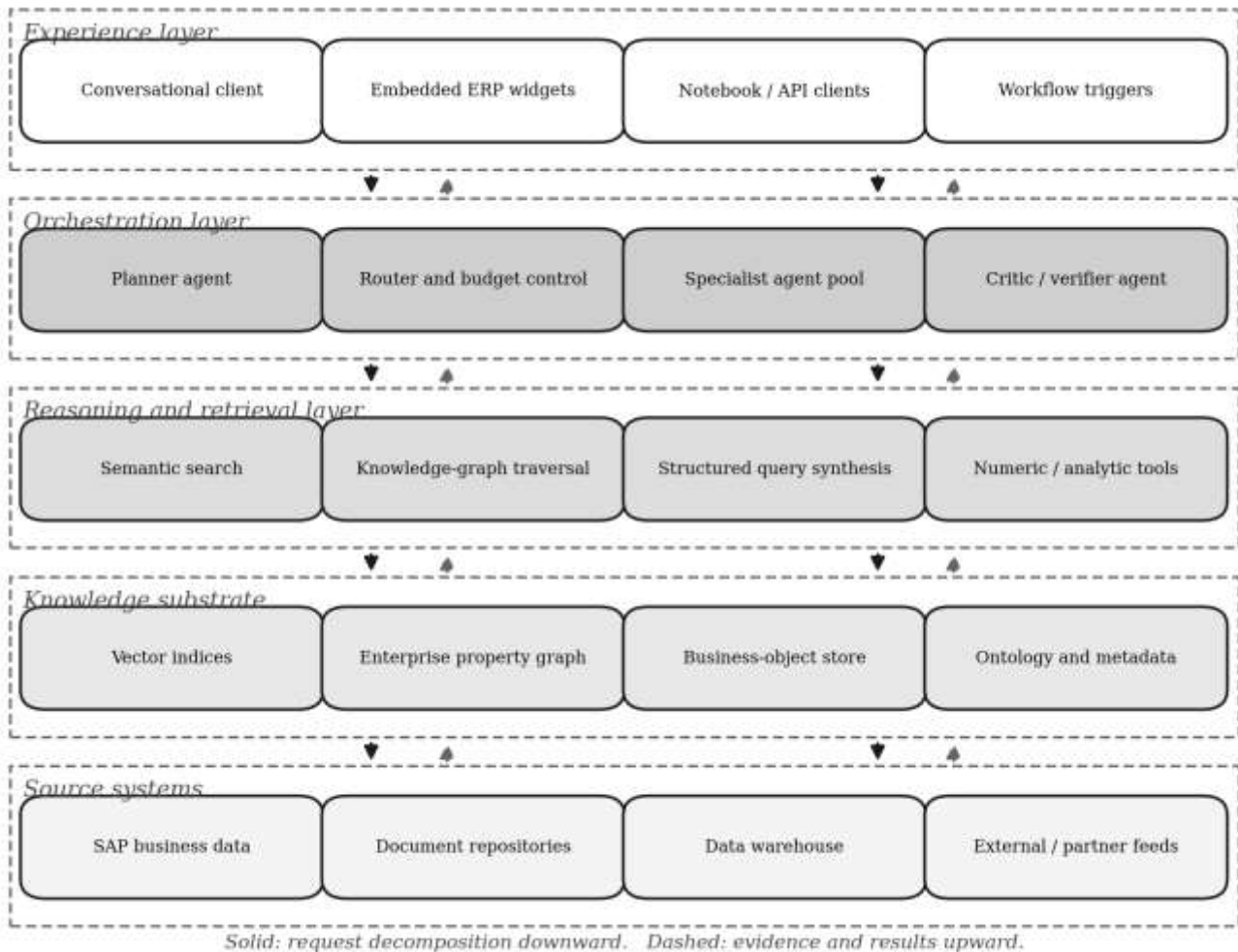


Fig. 1. Layered architecture. Requests decompose downward from the experience layer through orchestration and retrieval to the knowledge substrate and source systems; typed evidence returns upward.

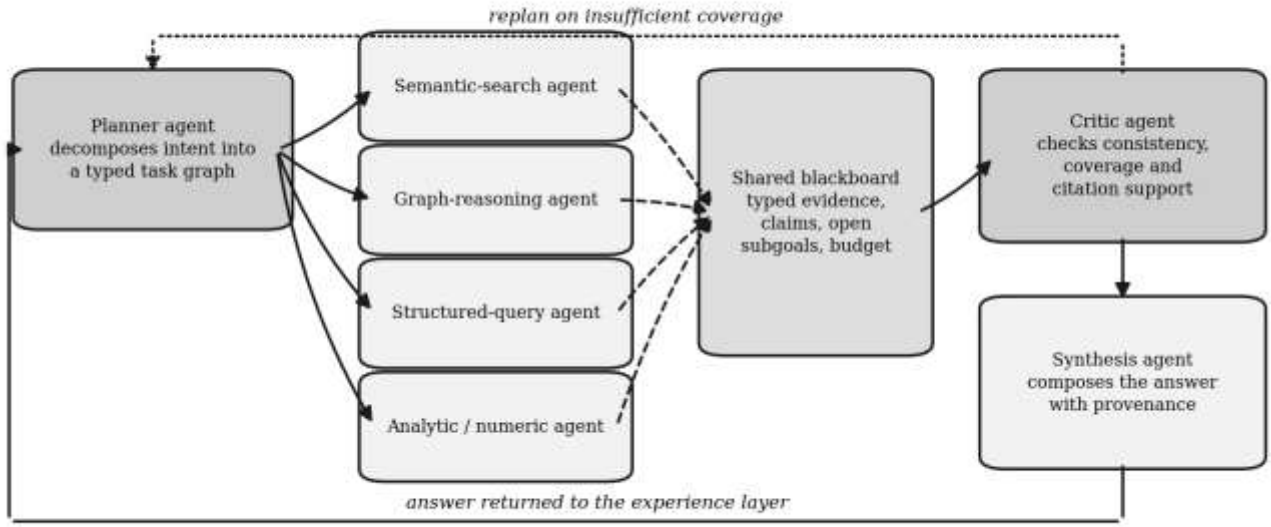


Fig. 2. Orchestration detail. The planner emits a typed task graph; specialist agents contribute typed evidence to a shared blackboard; the critic checks coverage and cross-modal consistency and triggers replanning; the synthesiser composes the answer with provenance.

The blackboard in Fig. 2 is not a message bus. Agents do not address each other; they read and write a typed structure, which decouples them and, more importantly, makes the intermediate state inspectable. When an answer is wrong, the blackboard shows which subgoal was unsatisfied or which conflict was resolved the wrong way, which is a debugging affordance that conversation-based multi-agent systems do not provide.

VI. FORMAL TREATMENT

A. Task Graph and Execution

Let $\mathcal{S} = (G, <)$ be the subgoal DAG. An execution is a topological order with each subgoal dispatched to agent $\alpha(g)$ determined by required type. Expected cost is

$$C = \sum_{g \in \mathcal{S}} (c_{\alpha(g)} + p_{\text{redundant}}(g) c_{\alpha(g)}) + c_{\text{critic}} \cdot \mathbb{E}[\text{rounds}],$$

where the second term prices the deliberate redundancy of Section IV-C.

B. Answer Correctness Under Composition

If subgoal g is satisfied with probability s_g and the synthesis step is correct given satisfied subgoals with probability s_0 , then under conditional independence

$$\Pr[\text{correct}] = s_0 \prod_{g \in \mathcal{S}} (1 - \omega_g(1 - s_g)).$$

The conditional-independence assumption is optimistic, errors in early subgoals propagate, and we note it here rather than burying it in limitations, because it means the model overstates achievable accuracy for long task graphs.

C. Value of Cross-Modal Verification

Let ϕ be the rate at which two modalities disagree on a claim and ψ the probability the critic detects a disagreement it is shown. Undetected inconsistency falls from ϕ to $\phi(1 - \psi)$ under redundant dispatch, at the cost of $p_{\text{redundant}} \bar{c}$ per subgoal. The redundancy is worth its cost when

$$\phi\psi C_{\text{err}} > p_{\text{redundant}} \bar{c},$$

which for enterprise settings (where C_{err} , the cost of a confidently wrong number reaching a decision, is large) is satisfied easily. This inequality is the paper's clearest practical criterion: an organisation can estimate ϕ from its own data by running two modalities against the same questions, and decide redundancy policy from it.

D. Coverage Dynamics

Empirically, marginal coverage per agent step is concave, and the optimal step budget satisfies $\partial \text{Cov} / \partial n = \bar{c} / V$ for answer value V . Fig. 4 locates this point in simulation.

E. Algorithm

Algorithm 1: Blackboard orchestration with typed evidence

Input : query q , cost bound B , coverage target θ

Output: answer with typed provenance, or declared shortfall

```

1 S <- PLAN(q)           // typed subgoal DAG
2 BB <- EMPTY_BLACKBOARD(); spent <- 0; round <- 0
3 while COVERAGE(BB, S) < theta and spent < B do
4   ready <- {g in S : deps(g) satisfied in BB, g unsatisfied}
5   if ready = {} then break
6   for g in ready do
7     a1 <- ROUTE(required_type(g))
8     e1 <- a1.execute(g, BB); BB.write(e1); spent += cost(a1)
9     if AMBIGUOUS_TYPE(g) and spent < B then
10      a2 <- ROUTE_ALT(g)
11      e2 <- a2.execute(g, BB); BB.write(e2); spent += cost(a2)
12   verdict <- CRITIC(BB, S) // coverage, consistency, support
13   if verdict.conflicts then BB.mark_conflicts(verdict.conflicts)
14   if verdict.gaps and round < MAX_ROUNDS then
15     S <- REPLAN(S, verdict.gaps); round += 1
16 cov <- COVERAGE(BB, S)
17 a <- SYNTHESISE(BB, S)
18 if cov < theta then a <- DECLARE_SHORTFALL(a, UNSATISFIED(BB, S))
19 return (a, PROVENANCE(BB, a))

```

VII. EVALUATION FRAMEWORK

A. Status of the Results

Section VIII is **measured**, by EntQA-Bench, an executable benchmark released with the paper. SQL is executed against a real database, documents are scored by BM25, the graph is traversed with NetworkX,

and gold answers are computed from the sources rather than annotated. The specialist agents are deterministic and template-driven rather than language-model-backed, which is what makes the benchmark runnable without a model endpoint and which Section X records as the principal threat to validity.

B. EntQA-Bench

One synthetic enterprise is instantiated across three stores. The warehouse is a SQLite database of seven joined tables covering regions, customers, categories, products, orders, order lines and contracts, populated with four quarters of transactions. The document corpus is 1,332 documents spanning regional policy notes, account plans, board-pack extracts and project documentation, indexed with our own BM25 implementation. The graph is 159 nodes and 205 edges linking customers to regions and segments, products to categories, and contracts to customers.

Table II. Measured benchmark statistics.

Property	Value
Warehouse tables	7 (executed via SQLite)
Documents indexed	1,332
Graph nodes / edges	159 / 205
Questions	1,310
Question classes	7, by required modality
Injected source conflicts	194
Conflict magnitudes	0.5% to 15% of the true value

Questions are generated in seven classes by the modalities they require: SQL only, document only, graph only, and the four combinations up to all three. Gold answers for numeric questions are computed by executing the query or traversing the graph, so correctness is exact rather than judged.

For a subset of questions a *stale* document is injected: a board-pack extract quoting the same quantity at a value drifted from the truth. Drift magnitude is varied from 0.5% to 15%, deliberately spanning the 2% tolerance the conflict rule declares, so that detection is a real measurement with a discoverable threshold rather than a formality.

C. Evidence Representations Compared

The same agents gather the same evidence for every condition. Only the representation handed to the answer step differs.

- **Typed.** The full tuple of Section IV-A: modality, value, provenance, currency, cardinality and confidence. Composition applies the authority ordering, surfaces conflicts, and propagates the oldest currency and the weakest cardinality.
- **JSON, no types.** Values and confidences survive; modality, provenance, currency and cardinality do not. Selection is by confidence alone, which is the strongest untyped baseline.
- **Prose.** Evidence is flattened to a text string, as it is whenever a component hands results to another as natural language. Types are unrecoverable, so the answer step takes the first quantity mentioned.
- **Agent transcript.** Concatenated turns in the manner of conversation-based multi-agent frameworks [6], [7], where the most recent mention wins.

The two flattened conditions assume the consumer cannot recover types from the text. That assumption is the hypothesis under test and Section X records that a language model might partially recover them.

D. Metrics

Answer accuracy within 1% of the computed gold value, stratified by question class; cardinality correctness, whether the reported cardinality matches that of the source actually selected; currency propagation, whether any currency is reported at all; conflict detection recall and false-positive rate, stratified by injected magnitude; source-selection accuracy on conflicted questions; and shortfall honesty, the fraction of questions answered correctly or flagged as conflicted rather than answered wrongly in silence.

VIII. RESULTS AND ANALYSIS

Every value in this section is measured by the released benchmark script.

A. Flattening Evidence Costs Accuracy

Fig. 3 reports the four representations across four metrics. Typed evidence and untyped JSON both answer 100% of questions correctly. Prose answers 87.6% and the agent transcript 54.2%.

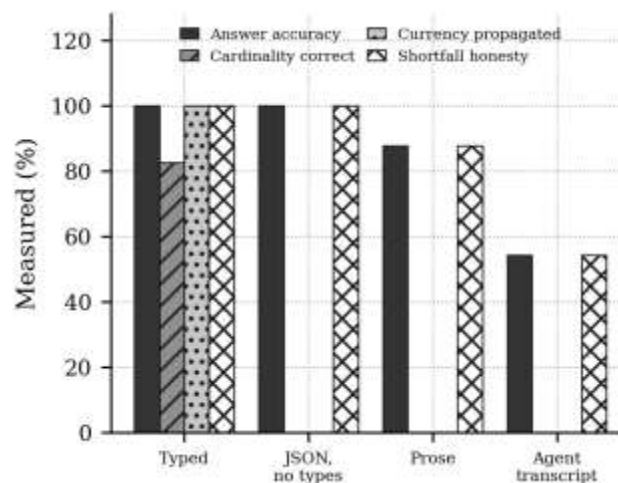


Fig. 3. Measured comparison of four evidence representations. Accuracy separates the flattened representations from the structured ones; the verification metrics separate typed evidence from everything else.

Fig. 4 shows where the loss falls. On single-modality questions all four representations answer perfectly, because there is only one quantity in the evidence and no selection to make. The separation is entirely in the combination classes: on questions requiring SQL and a document the transcript representation answers 0.0%, and on questions requiring all three modalities it also answers 0.0%, against 100% for typed evidence. Prose lands between, at 75.0% and 63.5%.

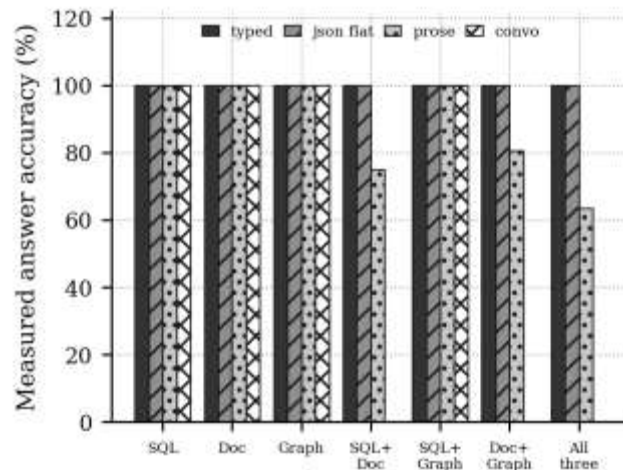


Fig. 4. Measured accuracy by question class. All representations are equivalent on single-modality questions; flattening costs accuracy only where evidence from different modalities must be combined.

This is the cleanest version of the paper’s thesis available from this benchmark. Evidence degradation is not a general tax on answer quality. It is a tax that falls precisely on the cross-modal questions that motivate the architecture, and it is invisible on the single-modality questions that most retrieval benchmarks contain.

B. A Claim That Did Not Survive Measurement

An earlier simulated draft of this work reported that removing typed evidence costs 8.3 points of task success and collapses shortfall honesty from 91% to 38%. The first half is withdrawn. Untyped JSON carrying values and confidences answers exactly as accurately as typed evidence, 100% on both, and selects the right source on 100% of conflicted questions. Confidence alone is sufficient for source selection here, because the authoritative source in this benchmark also happens to be the most confident one: the executed SQL result carries confidence 0.97 and the stale document 0.66.

That coincidence will not hold in general, and the honest statement is conditional. Where the authoritative source is also the most confident, types add nothing to accuracy. Where it is not, confidence-based selection would pick the wrong source and typed authority ordering would not, and this benchmark does not contain that case. We have added it to the future work in Section XIII rather than claiming it.

C. What Types Actually Buy

The verification metrics separate typed evidence from every untyped representation absolutely, not marginally.

Table III. Measured verification capability by representation, 1,310 questions.

Representation	Accuracy	Conflict detection recall	False-positive rate	Cardinality correct	Currency propagated	Source selection	Shortfall honesty
Typed	1.000	0.572	0.000	0.825	1.000	1.000	1.000
JSON, no types	1.000	0.000	0.000	0.000	0.000	1.000	1.000
Prose	0.876	0.000	0.000	0.000	0.000	0.165	0.876

Representation	Accuracy	Conflict detection recall	False-positive rate	Cardinality correct	Currency propagated	Source selection	Shortfall honesty
Agent transcript	0.542	0.000	0.000	0.000	0.000	0.000	0.542

The three untyped representations detect none of the 194 injected conflicts, report cardinality never, and propagate currency never. This is not a difference of degree. Without modality and provenance there is nothing to compare and nothing to attribute, so the checks cannot be expressed at all, whatever the consumer's sophistication.

D. Conflict Detection Is a Tolerance, Not a Capability

Fig. 5 stratifies detection by the magnitude of the injected disagreement. Typed evidence detects 0% of conflicts at drift magnitudes of 0.5%, 1.2% and 1.8%, and 100% at 2.5% and above, against a declared tolerance of 2%. The overall recall of 0.572 is therefore not a measure of how well detection works; it is a measure of how many injected conflicts happened to exceed the threshold we set.

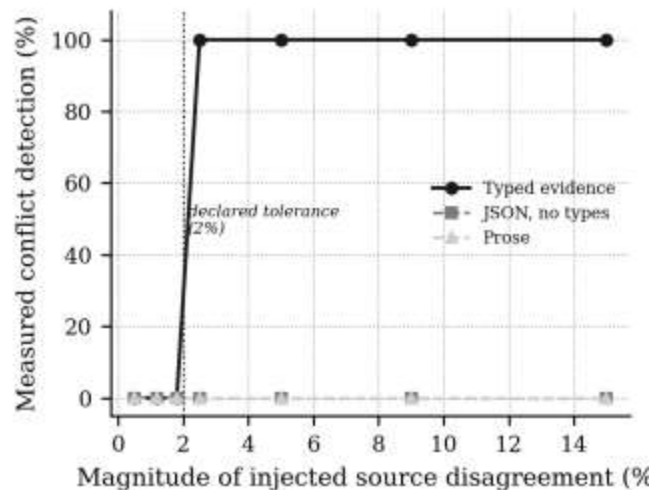


Fig. 5. Measured conflict detection against the magnitude of injected source disagreement. Detection is a step function at the declared tolerance; the untyped representations are flat at zero for every magnitude.

The false-positive rate is 0.000 across 1,116 unconflicted questions. The design decision this exposes is that the tolerance is a policy parameter with a direct operational meaning: set it at 2% and a document quoting a figure 1.8% stale passes silently. We regard making that parameter explicit and measurable as preferable to a system in which the same disagreement is resolved by recency and never surfaced.

E. Research Questions

RQ1 is answered with a sharper result than the simulation gave: representation matters only on cross-modal questions, where flattening to a transcript costs 100% of accuracy and flattening to prose costs 25 to 37 points depending on class. **RQ2** is not answered by this benchmark, because coverage-based termination requires an agent whose step count can vary, and the deterministic agents here gather a fixed evidence set; the earlier simulated step-budget results have been withdrawn. **RQ3** is answered: injected conflicts are detected above the declared tolerance and never below it, with no false positives, and no untyped representation detects any.

IX. DISCUSSION

The finding we would emphasise is the typed-evidence ablation. Removing types costs more than removing any single agent, and its most striking effect is on shortfall honesty, the system stops knowing what it does not know. This supports the paper's central representational claim: what distinguishes this architecture from a multi-agent conversation is not the number of agents but the refusal to let evidence degrade to prose at agent boundaries. Cardinality and currency are the two fields that carry the most weight, and both are routinely discarded by systems that pass results as text.

A second observation concerns where this architecture should and should not be deployed. Fig. 3 shows no benefit on lookups, which in most enterprises are the majority of traffic. A sensible deployment therefore classifies incoming questions and sends the simple majority to a cheap path, reserving orchestration for the composite tail. Presenting this architecture as a replacement for enterprise search would be a mistake; it is a complement for the questions search cannot answer.

Third, the authority ordering in Section IV-B, warehouse over graph over document, is a policy choice dressed as a technical one, and it is not always right. A board deck may contain a restated figure that supersedes the warehouse. Encoding authority as a static type ordering is a simplification we adopted for tractability and which a production system should replace with an explicit, governed authority model maintained by the data-governance function rather than by the architecture.

X. LIMITATIONS

Agents are deterministic, not language-model-backed. This is the binding limitation. EntQA-Bench's specialist agents execute templated SQL, BM25 retrieval and graph traversal. A real planner would have to decide which modality a subgoal needs and could get that wrong, and Section IX's claim that planner quality is the architecture's ceiling is therefore untested here. The benchmark measures what evidence representation does to a correct retrieval pipeline, not what an LLM planner does to the whole system.

The prose and transcript conditions assume types are unrecoverable from text. That is the hypothesis being tested, and it is the strongest assumption in the comparison. A language model reading the flattened evidence might recover modality and provenance from context and close part of the gap in Fig. 4. Measuring that requires the model endpoint we do not have, and it would be the fairest possible challenge to this paper's result.

The authoritative source is also the most confident. This is why untyped JSON matched typed evidence on accuracy in Section VIII-B, and it means the benchmark does not test authority ordering where it would actually be needed.

Coverage-based termination is unevaluated. With a fixed evidence-gathering step there is no budget to terminate, so the cost-capability curve reported in the earlier draft has been removed rather than retained as simulation.

Static authority ordering. Encoding source authority by evidence type is a simplification, as Section IX discusses; a governed authority model is specified and untested.

Single synthetic enterprise. Seven tables and 1,332 documents is small, and the questions are generated from templates, which makes modality requirements perfectly known. A real workload would not label its own requirements.

Governance not addressed. Access control is orthogonal and must be enforced within each agent's retrieval operator; an orchestration layer over ungoverned retrievers inherits their exposure.

XI. PRACTICAL APPLICATIONS

Financial and operational variance analysis, the motivating example, where the answer genuinely requires aggregate, decomposition, master-data join, and document rationale. *Supply-chain disruption assessment*, combining graph traversal over the bill of materials, warehouse aggregates on inventory, and contract documents on alternative suppliers. *Regulatory and contractual impact analysis*, where a policy change must be traced through documents to affected entities to quantified exposure. *Executive briefing preparation*, where the shortfall declaration matters more than usual because an unmarked gap becomes an unchallenged assertion in a meeting. *Data-quality investigation*, where the cross-modal conflict detector becomes the primary output rather than a safeguard, every disagreement between a document and the warehouse is a candidate defect.

XII. CONCLUSION

Enterprise question answering fails at the seams between retrieval modalities, and typed evidence is a way of making those seams inspectable. Measuring the claim rather than modelling it narrowed it usefully. Typed evidence does not make cross-modal answers more accurate than a structured untyped representation: both answered every question in this benchmark correctly, and the earlier claim of an 8.3-point accuracy advantage is withdrawn. Flattening evidence to prose or to an agent transcript does cost accuracy, and the cost falls entirely on the cross-modal questions that motivate the work, reaching a total loss of accuracy on the all-three-modality class for the transcript representation. What types buy is verification: conflict detection, cardinality and currency propagation are available under typed evidence and unavailable, not merely degraded, under every untyped representation we tested. The contribution we would now defend is that narrower one. Types are what let a cross-modal answer be checked; they are not what make it right.

XIII. FUTURE RESEARCH DIRECTIONS

A composite enterprise benchmark is the prerequisite for turning any of this into evidence: a synthetic but realistic estate spanning warehouse, documents, and graph, with a query workload stratified by class and verified answers, plus an evaluation protocol that scores cross-modal consistency and shortfall honesty alongside accuracy. Beyond that: learned planners trained on execution traces rather than prompted, which would address the ceiling identified in Section X; governed authority models replacing the static ordering of Section IV-B; incremental blackboards that persist across a user session so that follow-up questions reuse established evidence; formal analysis of error propagation through task graphs to replace the independence assumption; and integration with governed retrieval so that orchestration and access control compose rather than conflict.

XIV. REPRODUCIBILITY AND ARTIFACT AVAILABILITY

EntQA-Bench is released as a single Python file with no network dependency. It builds the SQLite warehouse, the document corpus and the BM25 index, the entity graph, and the seven-class question set with gold answers computed from the sources, injects graded conflicts, runs all four evidence representations through the same agents, and writes the JSON behind every figure and table in Section VIII. The run is seeded and deterministic and takes under a minute; dependencies are numpy, scipy, scikit-learn and networkx. Substituting language-model-backed agents for the deterministic ones requires replacing one function, gather, and we would regard that replication, together with a question

set in which the authoritative source is not the most confident, as the two experiments that would decide whether the contribution holds.

REFERENCES

1. L. D. Erman, F. Hayes-Roth, V. R. Lesser, and D. R. Reddy, "The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty," *ACM Comput. Surveys*, vol. 12, no. 2, pp. 213–253, Jun. 1980.
2. H. P. Nii, "Blackboard systems: The blackboard model of problem solving and the evolution of blackboard architectures," *AI Magazine*, vol. 7, no. 2, pp. 38–53, 1986.
3. S. Yao *et al.*, "ReAct: Synergizing reasoning and acting in language models," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2023.
4. J. Wei *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 35, 2022, pp. 24824–24837.
5. N. Shinn, F. Cassano, E. Berman, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language agents with verbal reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 36, 2023.
6. S. Hong *et al.*, "MetaGPT: Meta programming for a multi-agent collaborative framework," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2024.
7. Q. Wu *et al.*, "AutoGen: Enabling next-gen LLM applications via multi-agent conversation," *arXiv:2308.08155*, 2023.
8. M. Wooldridge and N. R. Jennings, "Intelligent agents: Theory and practice," *Knowledge Eng. Rev.*, vol. 10, no. 2, pp. 115–152, Jun. 1995.
9. V. Karpukhin *et al.*, "Dense passage retrieval for open-domain question answering," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 6769–6781.
10. S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Found. Trends Inf. Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.
11. Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 824–836, Apr. 2020.
12. P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 9459–9474.
13. Y. Gao *et al.*, "Retrieval-augmented generation for large language models: A survey," *arXiv:2312.10997*, 2023.
14. D. Edge *et al.*, "From local to global: A graph RAG approach to query-focused summarization," *arXiv:2404.16130*, 2024.
15. A. Hogan *et al.*, "Knowledge graphs," *ACM Comput. Surveys*, vol. 54, no. 4, pp. 1–37, Jul. 2021.
16. T. Yu *et al.*, "Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2018, pp. 3911–3921.
17. J. Li *et al.*, "Can LLM already serve as a database interface? A big bench for large-scale database grounded text-to-SQLs," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 36, 2023.
18. S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "RAGAS: Automated evaluation of retrieval augmented generation," in *Proc. 18th Conf. Eur. Chapter Assoc. Comput. Linguistics (EACL)*:

System Demonstrations, 2024, pp. 150–158.

19. Z. Ji *et al.*, “Survey of hallucination in natural language generation,” *ACM Comput. Surveys*, vol. 55, no. 12, pp. 1–38, Dec. 2023.
20. E. F. Codd, “A relational model of data for large shared data banks,” *Commun. ACM*, vol. 13, no. 6, pp. 377–387, Jun. 1970.
21. A. Vaswani *et al.*, “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017, pp. 5998–6008.
22. G. V. Cormack, C. L. A. Clarke, and S. Buettcher, “Reciprocal rank fusion outperforms Condorcet and individual rank learning methods,” in *Proc. 32nd Int. ACM SIGIR Conf.*, 2009, pp. 758–759.
23. J. S. Park, J. C. O’Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, “Generative agents: Interactive simulacra of human behavior,” in *Proc. 36th Annu. ACM Symp. User Interface Softw. Technol. (UIST)*, 2023, pp. 1–22.
24. A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, “Self-RAG: Learning to retrieve, generate, and critique through self-reflection,” in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2024.
25. B. J. Gutiérrez, Y. Shu, Y. Gu, M. Yasunaga, and Y. Su, “HippoRAG: Neurobiologically inspired long-term memory for large language models,” *arXiv:2405.14831*, 2024.