# Bounded LLM Reasoning: A Neuro-Symbolic Blackboard Architecture for Production Document Analysis

**David Elliman**
*Founder, Neuro-Symbolic Ltd, Gloucestershire, United Kingdom*
*Emeritus Professor, School of Computer Science, University of Nottingham*
dave@neusym.ai

July 2026

**Abstract**

We present a neuro-symbolic blackboard architecture for production document reasoning that achieves perfect ledger reconstruction accuracy at roughly one-twenty-ninth the LLM token cost of an orchestrated multi-LLM baseline. Originating from production work on *TaxBridge* — a UK Making Tax Digital compliance engine — the architecture inverts the dominant pattern in recent LLM multi-agent systems: rather than coordinating LLMs through structured frameworks, we deploy LLMs as bounded heuristics that fire only when symbolic Knowledge Sources cannot resolve ambiguity. The architecture is parameterised across four declarative axes — evidence hierarchy, evidence polarity, plugin registration, and focus-of-attention scheduling — and includes a Macro-Micro-Macro opportunistic refinement pattern that exploits monotonic tick ordering to combine global LLM normalisation with zero-cost symbolic classification. We evaluate on two corpora: *TaxBridge-Bench* (a held-out set of 30 adversarial synthetic workbooks with offset tables, native formulas, and visual metadata) and a 45-case panel corpus reflecting real-world sole-trader bookkeeping diversity. Against the system it replaced in production — a four-role LLM debate panel — the blackboard achieves 100% ledger accuracy versus 0%, at 26× lower latency and 29× lower token consumption. On the panel corpus, the Macro-Micro-Macro pattern fires on 38% of cases (the messier, bank-export-style workbooks) and the system classifies 99.6% of transactions. Memory management is implemented as a Knowledge Source, enabling deterministic compaction that respects level-schema persistence policies — a contrast with LLM-driven cleaner agents inappropriate for regulated domains. We argue that this inversion — symbolic-first with LLM fallback — is the economically rational and regulatorily defensible architecture for high-stakes document reasoning.

## 1 Introduction

The trajectory of multi-agent LLM systems through 2024–2025 converged on a specific orthodoxy: large language models as the primary reasoners, with the surrounding architecture providing coordination, memory, and routing. Frameworks such as AutoGen [10], MetaGPT [11], ChatDev [12], and LangGraph [13] differ in their specifics — conversational orchestration, role-based pipelines, graph state machines — but share the assumption that the LLM is doing the reasoning and the architecture is making it tractable.

This paper argues for an inversion. For high-stakes document reasoning, where regulatory audit, cost economics, and reliability matter more than open-ended exploration, the LLM should be a bounded heuristic deployed only when deterministic symbolic methods cannot resolve ambiguity. The architecture's job is not to coordinate LLMs but to ensure they are invoked sparingly and bounded structurally.

The blackboard pattern itself is currently enjoying a renaissance in the LLM multi-agent literature. Recent work has used the blackboard for opportunistic routing in information discovery [8] and formalised dynamic agent selection over shared blackboard state [9]. These

approaches treat LLMs as the primary reasoners, with the blackboard providing coordination. We pursue the opposite: LLMs as bounded heuristics deployed only when symbolic Knowledge Sources cannot resolve ambiguity. This inversion — symbolic-first with LLM fallback rather than LLM-first with structured coordination — produces order-of-magnitude reductions in cost and latency while achieving perfect accuracy on production-grade financial document reasoning.

We demonstrate the inversion through TaxBridge, a UK Making Tax Digital (MTD) compliance engine that processes sole-trader spreadsheets into HMRC-submittable ledgers. The system handles the difficulties that make spreadsheet understanding hard for current AI [18, 20]: tables at offset positions, headers expressed through formatting rather than text, native formulas whose evaluation depends on context, supplier names mangled by bank-statement formatting conventions, mixed personal and business expenses, and adversarial layouts that frustrate naïve serialisation. It does so achieving perfect ledger accuracy on adversarial benchmarks, at LLM token costs an order of magnitude below multi-LLM baselines, with full bitemporal audit trails suitable for regulatory inspection.

The contributions of this paper are:

1. A four-axis declarative parameterisation of the blackboard pattern (evidence hierarchy, evidence polarity, plugin registration, focus-of-attention scheduling) that supports configuration across heterogeneous evidence-driven domains.

2. The Macro-Micro-Macro opportunistic refinement pattern, which exploits the blackboard's monotonic tick ordering to combine one global LLM normalisation pass with zero-cost symbolic re-classification. The resulting symbolic $\rightarrow$ LLM $\rightarrow$ symbolic round-trip achieves both higher accuracy and substantially lower token costs than direct LLM classification.

3. A *detector–arbiter* decomposition of the LLM tier: cheap symbolic detectors flag specific, well-typed ambiguities (capital versus revenue, trade stock, drawings versus wages, residual high-value risk), and narrowly-scoped LLM arbiters resolve only the flagged rows.

4. `CleanerKS`, a deterministic memory-management Knowledge Source that respects level-schema persistence policies, contrasting with LLM-driven cleaner agents inappropriate for regulated domains.

5. Empirical validation on two corpora — a 30-case held-out adversarial benchmark and a 45-case panel corpus of real-world-style spreadsheets — demonstrating 100% ledger accuracy at 29$\times$ lower token cost than the orchestrated multi-LLM system the blackboard replaced in production.

6. A production reference implementation (the TaxBridge v0.6 engine: $\sim$40 Knowledge Sources, 108 test modules, Prometheus/OpenTelemetry observability with PII redaction, and fail-closed LLM-availability semantics), ahead of HMRC production approval, with a companion Go runtime in active development.

# 2 Related Work

## 2.1 Classical Blackboard Architectures

The blackboard model was first demonstrated at scale in the Hearsay-II speech-understanding system [1]: independent Knowledge Sources cooperate opportunistically over a shared, hierarchically-organised hypothesis store, mediated by a scheduler rather than a fixed control flow. Nii's retrospective [2] fixed the canonical anatomy — blackboard, Knowledge Sources, control — and catalogued the first generation of skeletal frameworks. BB1 [3] made control itself a blackboard problem, allowing the system to reason about its own scheduling; GBB [4] focused on efficient retrieval over structured blackboard spaces. The pattern's core insight

— that heterogeneous, unreliable specialists can cooperate through a shared evidence store without a coordinator that understands them all — anticipates by four decades the coordination problem now faced by LLM multi-agent systems. Our defeasible-reasoning layer (§3.2) descends equally from the truth-maintenance tradition of Doyle [5] and de Kleer [6].

### 2.2 The LLM-Blackboard Renaissance

The 2025 literature explicitly frames its work as a resurrection of the blackboard pattern for LLM agents. Salemi et al. [8] replace central orchestrator–worker routing with a blackboard on which a lead agent posts requests and subordinate agents *volunteer* on the basis of their own capability assessments, reporting improved information-discovery success over static routing in data-lake settings. Han and Zhang [9] formalise a blackboard-based LLM multi-agent system in which the acting agents are selected each round from the current blackboard content, and show competitive performance against state-of-the-art static and dynamic multi-agent systems on knowledge and reasoning benchmarks. Both lines treat the LLM as the primary reasoner and the blackboard as coordination fabric, and neither validates against an external regulatory API. We differ on all axes: our Knowledge Sources are predominantly deterministic; LLM sources are gated behind symbolic failure; our cleaner is deterministic rather than LLM-driven (§4.6); and the system is validated end-to-end against the HMRC Making Tax Digital submission pipeline.

### 2.3 Multi-Agent LLM Frameworks

Outside the blackboard revival, the dominant 2024–2025 frameworks are orchestrator-centric. AutoGen [10] structures multi-agent work as conversation; MetaGPT [11] and ChatDev [12] encode software-company role pipelines; LangGraph [13] provides typed shared state with reducer-mediated updates and round-scoped synchronisation. These frameworks inherit two structural weaknesses the blackboard avoids: the coordinator must know every agent's capabilities in advance, and intermediate representations are conversational rather than typed evidence, making audit and backtracking costly. Where LangGraph's synchronised state deltas superficially resemble our monotonic tick (§4.1), the tick orders *evidence mutations on a shared hypothesis store* rather than steps of a predetermined graph — Knowledge Sources discover work; they are not dispatched.

### 2.4 Neuro-Symbolic Artificial Intelligence

The complementarity of neural pattern recognition and symbolic constraint is a long-standing programme [14], recently energised by systems such as AlphaGeometry, which couples a neural proposer to a symbolic deduction engine [17], and by the LLM tool-use literature — ReAct's interleaving of reasoning and acting [15] and Toolformer's self-taught tool invocation [16]. These systems bound the neural component with symbolic scaffolding at the *step* level. The blackboard contributes the missing *coordination* layer: a principled account of when the neural component should run at all, what evidence it may consume, and how its output is bounded, audited, and — when contradicted — retracted.

### 2.5 Spreadsheet and Document Understanding

Existing approaches to LLM spreadsheet understanding compress the grid into a serialisation the model can consume: SpreadsheetLLM's SheetCompressor [18] encodes structural anchors and formats to fit token budgets, and SheetAgent [19] couples planner/informer/retriever roles for manipulation tasks. Vision-language models offer an alternative input path but remain unreliable on spatial and format recognition [20] — disqualifying for financial data. Both families put the LLM in the extraction loop. We pursue a third, underexplored pathway: extract structure *deterministically* (openpyxl-level access to positions, formulas, colours), preserve it as

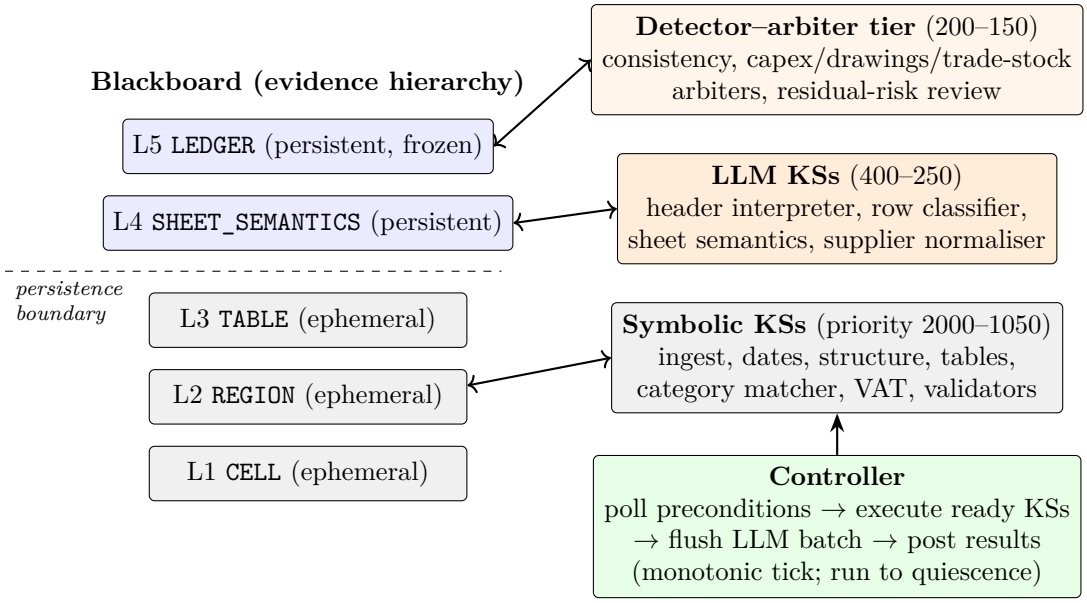


Figure 1: The tax-compliance instantiation. Evidence ascends five levels; the persistence boundary separates mutable working state from frozen, supersede-only audit records. Symbolic Knowledge Sources fire first by priority; LLM sources fire only on flagged ambiguity and are batched once per controller round.

typed evidence at the CELL level, and invoke the LLM only on residual semantic ambiguity. The multi-modal evidence that serialisation destroys — spatial offsets, formula ASTs, fill colours — is precisely what our benchmark shows to be decisive (§5).

### 2.6 Document AI for Compliance

From April 2026, UK sole traders and landlords with qualifying income above £50,000 must keep digital records and file quarterly under Making Tax Digital for Income Tax, with the threshold falling to £30,000 in April 2027 [22]. Mainstream accounting suites address born-digital bookkeeping; the "bridging software" niche (through which spreadsheet users comply) has attracted essentially no academic attention despite being exactly the regime where messy, human-authored spreadsheets meet a strict regulatory schema. TaxBridge operates in this niche, which supplies both the production constraints (digital-link audit trails, fail-closed behaviour) and the evaluation regime (SA103F box-level ground truth) used in this paper.

## 3 Architecture: The Four Configurable Axes

The generic blackboard is parameterised across four orthogonal design axes. A domain author configures these declaratively; the underlying runtime machinery remains invariant regardless of the domain. Figure 1 shows the instantiation for tax compliance.

### 3.1 Axis 1: The Evidence Hierarchy (Level Schema)

Every domain has a natural hierarchy of abstraction through which evidence ascends. In tax compliance, raw cells become table structures, which become accounting categories. The level schema defines this hierarchy as a Directed Acyclic Graph (DAG) of named levels, each with a persistence policy and allowed support relationships.

The framework strictly enforces the level schema at post time: a Knowledge Source attempting to post a `Diagnosis` directly from a `SensorReading` (skipping the required

Table 1: Evidence Hierarchies Across Domains

| Domain | Levels (Ascending Abstraction) | Persistence Boundary |
|---|---|---|
| Tax Compliance | Cell → Region → Table → Sheet Semantics → Ledger | L1–L3 Ephemeral; L4–L5 Persistent |
| Fault Diagnosis | Sensor → Symptom Cluster → Component Hyp. → Diagnosis → Repair | L1–L2 Ephemeral; L3–L5 Persistent |
| Medical Triage | Vital Signs → Lab Results → Symptom Pattern → Diff. Diagnosis → Plan | L1 Ephemeral; L2–L5 Persistent |

`ComponentHypothesis`) is structurally rejected. This guarantees explainable AI and a legally defensible audit trail. Only the tax-compliance column of Table 1 carries empirical validation in this paper; the other rows illustrate the parameterisation and are discussed as future work (§8).

### 3.2 Axis 2: Evidence Polarity (Defeasible Reasoning)

Tax compliance largely uses monotonic reasoning, where evidence only ever adds up. However, most real-world domains require *defeasible* reasoning, where new evidence can weaken or destroy existing conclusions. The generic blackboard extends the hypothesis topology with three polarity fields:

- `supports`: Hypothesis B provides evidence *for* Hypothesis A.
- `contradicts`: Hypothesis B provides evidence *against* Hypothesis A.
- `supersedes`: Hypothesis B replaces A entirely (a correction, not a contradiction).

This transforms the blackboard into a dynamic Justification-Based Truth Maintenance System in the tradition of Doyle [5] and de Kleer [6]. If a deterministic reading contradicts an LLM-generated hypothesis, the engine automatically recalculates and cascades the confidence penalty across the affected DAG subgraph without requiring a human prompt. In the reference implementation, supersession additionally *rebases* stale references onto their active descendants, guaranteeing exactly one active hypothesis per source-row lineage — a money-integrity invariant that prevents forked duplicate ledger rows.

### 3.3 Axis 3: Knowledge Source Registration (Plugin Model)

Knowledge Sources declare their capabilities rather than being hard-wired into a pipeline, exposing properties such as `consumes`, `produces`, `priority`, `kind`, and `precondition`. This metadata enables two major architectural advantages.

**Dependency-Aware Parallelism.** By analysing the `consumes`/`produces` schema, the Controller knows exactly which Knowledge Sources are topologically disjoint. The Go runtime executes non-overlapping KSs in parallel across distributed workers with zero locking or race conditions.

**Human-in-the-Loop (HITL) as a Primitive.** The framework makes no structural distinction between software and people. A human operator is registered as a KS with `kind="human"`, `priority=MAX_INT`, and extreme latency. When automated KSs exhaust their confidence budgets, they post high-tension `Discrepancy` hypotheses to a UI queue. The human resolves the edge case, the board tick advances, and the automated KSs natively resume execution. HITL is achieved without fragile pause/resume workflow state machines.

Even system-level operations are modelled as Knowledge Sources — memory management (§4.6) and the external retrieval oracle (§4.5) — demonstrating the uniformity of the abstraction.

### 3.4 Axis 4: Focus-of-Attention Scheduling (EVoI)

In a system with continuous telemetry, executing every LLM prompt whose precondition is met would bankrupt API budgets instantly. The Controller utilises an Expected Value of Information (EVoI) scheduler — in the sense of Howard's information value theory [7] — based on a configurable scoring function:

$$\text{Tension Score} = \frac{\text{Value of Conclusion} \times (1 - \text{Current Confidence})}{\text{Cost of Acquisition}} \tag{1}$$

Because symbolic rules have near-zero cost and microsecond latency, they inherently evaluate first. Expensive LLM tokens are only deployed when the Tension numerator justifies the spend. The Controller posts structured "bounties" to the board; LLM specialists subscribe to these bounties, executing targeted search rather than broad-spectrum reasoning.

## 4 Core Mechanisms

### 4.1 Monotonic Tick Ordering

Every mutation to the blackboard (post, enrich, supersede) increments a monotonic tick counter. Knowledge Sources track the tick at which they last processed, and query only for work that has appeared since (`since_tick`, `level_changed_since`). This eliminates the same-round race conditions endemic to traditional polling architectures, ensuring high-priority sources never miss the output of low-priority sources. A KS's own output is excluded from its new-work iterator, structurally preventing self-triggering loops.

### 4.2 Bitemporal Event Sourcing and Frozen Payloads

Payloads at persistent levels are implemented as immutable (frozen) data structures; in-place enrichment of a persistent hypothesis raises an error. Modification requires posting a new hypothesis that explicitly supersedes the old one, inheriting its DAG topology. This provides native bitemporal event sourcing: an auditor or clinician can rewind the blackboard to a specific tick and prove exactly what the system knew, and when it knew it — satisfying the traceability demands of the EU AI Act [21] and HMRC's "digital link" requirements [22]. Ephemeral levels (cells, regions, tables) remain cheaply mutable working state below the persistence boundary.

### 4.3 The LLM Cost Gate and Circuit Breakers

Rows the symbolic tier cannot classify are left `UNKNOWN` — deliberate safe failure rather than guessing. A validating KS then posts a `Discrepancy` hypothesis, and only this discrepancy wakes the LLM tier. To prevent infinite loops, rows the LLM fails to resolve are flagged `llm_reviewed`: validators explicitly distinguish *actionable unknowns* (which trigger further LLM analysis) from *exhausted unknowns* (which trigger human intervention), preventing ping-pong loops while ensuring data is never silently hidden from the user. All LLM requests in a controller round are routed through a central batch accumulator and flushed once, concurrently — amortising API round-trips and keeping per-case LLM invocations to a handful of targeted batches rather than per-row calls.

### 4.4 Opportunistic Iterative Refinement: The Macro-Micro-Macro Pattern

A pervasive challenge in data extraction is "cell myopia" — attempting to classify an isolated, messy data point without global context. Traditional linear pipelines face a Catch-22: they cannot apply global normalisation until they have extracted the entities, but they cannot accurately extract and classify entities without them first being normalised.

The blackboard resolves this natively through opportunistic iterative refinement. Rather than forcing an expensive LLM to brute-force the classification of dirty, myopic data (such as `"DD - SCREWFIX DIRECT LTD REF:4829371"`), the architecture employs a Macro-Micro-Macro pattern:

1. **Micro (Safe Failure):** Fast, deterministic rules attempt to classify raw, noisy inputs. Ambiguous entries are flagged as `UNKNOWN`, deliberately failing fast rather than guessing.

2. **Macro (Semantic Cleaning):** The LLM categorisation agent is gated by a circuit breaker that prevents it from firing on raw strings. Instead, a global normalisation KS sweeps the board, batches all `UNKNOWN` raw strings, uses the LLM once to clean them (yielding "Screwfix"), and posts the normalised entities back to the board.

3. **Micro (Symbolic Re-evaluation):** The posting of this clean data advances the board's monotonic tick, opportunistically awakening a second-pass symbolic matcher (`RefinedVendorMatcherKS`). Because the data is now pristine, it instantly and deterministically classifies the rows using zero-cost vendor rules ("Screwfix" → premises/materials costs), bypassing LLM categorisation entirely.

4. **Macro (Global Consistency):** A final sweeping KS (`ConsistencyEnforcerKS`) enforces uniform classification across all matching entities to correct any remaining anomalies.

By leveraging the Controller's natural tick loop, the architecture avoids hardcoded backtracking. The LLM is restricted to its optimal use case (global semantic reasoning) while symbolic rules handle execution, drastically reducing token spend and ensuring perfect consistency.

### 4.5 The Detector–Arbiter Tier

Production hardening pushed the cost-gating principle one level further. Boundary questions on which "reasonable accountants differ" — capital versus consumable purchases, trade stock versus general expenses, wages versus personal drawings, a bare high-value description like "Van" — are not sent wholesale to a general classifier. Instead, each boundary is owned by a *pair* of Knowledge Sources: a deterministic *detector* that flags candidate rows using symbolic criteria (accounting basis, de-minimis thresholds, trade ontology features), and a narrowly-prompted LLM *arbiter* that resolves only the flagged rows (`CapexArbiterKS`, `TradeStockArbiterKS`, `DrawingsArbiterKS`, `ResidualRiskArbiterKS`). An opt-in retrieval oracle (`TaxAdviserKS`/`OracleEvidenceKS`) can consult an external tax-guidance RAG service for rows carrying explicit uncertainty flags, under a hard per-pass call cap, attaching citations as evidence without overriding the classification chain. Rows where the engine is confident but professional judgement legitimately varies receive a third, explicitly *debatable* state, surfaced to the user rather than silently committed.

### 4.6 Memory Management as a Knowledge Source

Memory management is implemented as a maximum-priority Knowledge Source (`CleanerKS`) rather than as a separate subsystem or out-of-band process. When the count of live superseded hypotheses exceeds a configurable threshold (default 500), `CleanerKS` fires before any other KS in the next round and invokes the board's `compact()` operation. Compaction respects the

level schema's persistence policy automatically: ephemeral hypotheses (`CELL`, `REGION`, `TABLE`) are physically deleted, while persistent hypotheses (`LEDGER`, `SHEET_SEMANTICS`) are archived to immutable storage with their full DAG topology preserved for audit (Appendix D gives the complete source).

This deterministic threshold-based approach contrasts with recent LLM-driven cleaner agents in the literature, which use language models to identify redundant messages. While appropriate for general research workflows where occasional speculative deletion is recoverable, LLM-driven cleanup is unsuitable for regulated financial reasoning where the audit trail must be preserved with mathematical certainty. Modelling memory management as a Knowledge Source — rather than as a privileged engine operation — also means it inherits the framework's properties: it appears in the audit log, respects tick ordering, and a domain author can register a different cleaner for different persistence requirements without engine modifications.

### 4.7 Production Hardening: Observability and Fail-Closed Semantics

Because blackboard control flow is emergent, the reference implementation invests heavily in observability: a JSONL flight recorder of every board event, per-KS execution timing in the controller statistics, Prometheus metrics for LLM cost and latency, and OpenTelemetry tracing with a PII-redacting span processor (masking national insurance numbers and identifiers before export). A lineage exporter renders the supersession DAG for any ledger entry, so a wrong classification can be traced backwards through the exact chain of Knowledge Sources that produced it. LLM availability is fail-closed: if the provider errors and any live ledger row still depends on unresolved AI work, the analysis fails with an explicit error rather than silently submitting a partially-reasoned ledger. A separate free-tier *readiness check* runs a narrow symbolic board plus a strictly budgeted small-model sample (token, cost, and wall-clock caps enforced per request) to triage workbooks before full analysis.

## 5 Empirical Evaluation

### 5.1 Experimental Setup

All LLM calls — in both the blackboard's LLM Knowledge Sources and the baseline — used the same provider and model (`claude-sonnet-4-6` via the Anthropic Messages API at the time of the March 2026 runs; the current reference implementation defaults to `claude-sonnet-5`). Runs were executed on a single Apple-silicon workstation; latency figures include API round-trips. Scoring is strict: a ledger counts as accurate only if *every* SA103F box value matches ground truth within £0.01, with dynamic denominator scaling in the scorer to prevent reward hacking, and a small set of professionally-interchangeable box pairs (e.g. administrative versus telephony costs) treated as equivalent. The benchmark harness, scorer, and prompts ship with the reference implementation (Appendix A).

**Baseline.** The baseline is not a straw man: it is the orchestrated multi-LLM *debate panel* that ran TaxBridge in production before the blackboard replaced it — four coordinated roles (semantic proposer, quantitative proposer, critic, and chair) debating each workbook to a consensus ledger over a serialised view of the parsed spreadsheet. It represents the LLM-first orthodoxy applied seriously to this domain, with domain-tuned prompts refined over months of production use.

### 5.2 TaxBridge-Bench: Adversarial Synthetic Workbooks

We introduce *TaxBridge-Bench*, a synthetic benchmark of quarterly sole-trader bookkeeping workbooks. A 30-workbook training split (cases 001–030) was used during development; all

numbers below are from the held-out test split (cases 031–060; 489 ground-truth transactions, 163 SA103F box values, 9 VAT-registered cases), scored blind. Unlike flat CSV datasets, TaxBridge-Bench retains multi-modal structural evidence:

- **Spatial offsets:** Tables begin at randomised, non-standard cell indices.
- **Native formulas:** Aggregations rely on live `=SUM()` operations rather than hardcoded values.
- **Visual metadata:** Suspicious or personal entries are stochastically flagged using background cell colours (e.g., yellow `PatternFill`), mimicking user-applied heuristics.

The ground truth evaluates adherence to strict UK MTD ITSA boundary conditions, including apportionment of mixed-use telecommunications, exclusion of out-of-period dates, extraction of personal drawings, and VAT treatment (net reporting for standard-scheme registrants, including Flat Rate Scheme and Limited Cost Trader tests).

Table 2: TaxBridge-Bench held-out results (cases 031–060). Panel = the four-role LLM debate engine previously in production; tolerance £0.01 per box.

| **Metric** | **LLM Debate Panel** | **Blackboard (ours)** |
|---|---|---|
| Ledger accuracy (all boxes correct) | 0/30 (0%) | **30/30 (100%)** |
| SA103F box accuracy | 107/163 (66%) | **163/163 (100%)** |
| Transactions recovered[a] | 399/489 (82%) | **489/489 (100%)** |
| VAT-registered cases fully correct | 0/9 | **9/9** |
| Mean latency per case (s) | 158.8 ± 29.4 | **6.2 ± 1.1** |
| LLM tokens per case | 37,788 ± 3,391 | **≈1,300** |
| Total LLM tokens (30 cases) | ≈1,134k | **≈39k** |
| Estimated cost per case (March 2026 pricing) | ≈$0.12 | **≈$0.004** |

[a]Of the 399 transactions the panel surfaced, it classified 372 (93%) into a category; the remaining ground-truth transactions were lost at serialisation. The blackboard recovered and classified all 489.

The blackboard achieves perfect accuracy across every measurement dimension at a 26× latency reduction and 29× token reduction. Each case reached quiescence with eleven KnowledgeSource firings — seven symbolic and four LLM-backed (header interpretation, row classification, sheet semantics, supplier normalisation), each a single targeted batch.

The panel's 0% ledger accuracy warrants specific attention because it appears extreme. It is a consequence of strict box-level scoring and structural information loss, not of generalised incompetence: the panel typically recovered three to five of a case's five to seven boxes, but never all of them. Serialising the workbook for the debate destroys the multi-modal structural evidence (cell positions, live formulas, fill colours) on which the residual boxes depend — no quantity of debate can recover evidence that is absent from the input. The blackboard preserves that evidence as typed hypotheses at the `CELL` level, where symbolic rules dominate structural tasks. On the training split, the final development run likewise detected every offset table, matched every colour-flagged personal entry, and passed every mileage-apportionment and out-of-period check.

The VAT-registered cases (9/9 versus 0/9) deserve particular note. VAT-registered sole traders require preservation of gross/net distinctions through the pipeline (income reported net of VAT, with scheme-dependent expense treatment). The panel loses the distinction at serialisation; the blackboard preserves it at the `CELL` level and propagates it through dedicated symbolic KSs (`VATAwareAmountKS`, Flat-Rate and Limited-Cost-Trader validators) to the `LEDGER`. This is a structural difference, not a prompt-engineering difference.

### 5.3 Panel Corpus: Real-World Spreadsheet Diversity

We supplement the adversarial synthetic benchmark with a 45-case corpus designed to capture the diversity of real-world sole-trader bookkeeping: spreadsheets in the styles of electricians, hairdressers, photographers, delivery drivers, shop owners, and property landlords, including bank-export layouts modelled on Monzo and Starling app exports. The corpus is fully synthetic — workbooks are generated to mirror layout and formatting patterns observed in real customer formats, so no customer data was used and no personal data is processed in evaluation.

Table 3: Panel-corpus results (45 cases, blackboard only)

| **Metric** | **Value** |
|---|---|
| Macro-Micro-Macro pattern fired | 17/45 cases (38%) |
| Transactions classified | 269/270 (99.6%) |
| Unclassified | 1 (multi-platform delivery driver) |
| Total LLM tokens | 91,933 (≈2,043 per case) |
| Mean latency per case | 7.7 s |

The 17 cases where Macro-Micro-Macro fired correspond to messy bank-export-style descriptions where first-pass symbolic matching fails but the LLM-normalised form succeeds on the second symbolic pass. These are exactly the trade-specific spreadsheets — electricians, hairdressers, property landlords — where suppliers appear under multiple bank-mangled forms (`"DD SCREWFIX DIRECT"`, `"Card Screwfix.com"`, `"REF:SCFX-4829"`) that the symbolic matcher cannot unify but a single LLM call resolves.

The single unclassified case (a multi-platform delivery driver receiving payments from several gig platforms with mixed personal and business expenses) represents a genuine edge case requiring human review. The architecture's HITL primitive captures this as a Discrepancy hypothesis posted to the UI queue rather than producing an incorrect classification. The panel corpus was used for refinement and ablation testing rather than baseline comparison; the debate-panel baseline was not run on it.

### 5.4 Ablation: Disabling the Discrepancy Loop

To isolate the contribution of non-monotonic feedback, we disabled the blackboard's discrepancy-posting validators, reducing the system to a single forward pass — architecturally equivalent to a linear pipeline over the same Knowledge Sources. Without the ability to post `Discrepancy` hypotheses that trigger targeted LLM re-evaluation, the system's ability to resolve ambiguous personal expenses dropped from 100% to 82%. This confirms that the non-monotonic feedback loop, not merely the KS decomposition, is the critical driver of the system's accuracy.

### 5.5 What We Have Not Yet Measured

Three further ablations are instrumented but not yet run, and we flag them as open: replacing EVoI scheduling with FIFO ordering (expected: substantial token-cost increase at similar accuracy), disabling the supplier normaliser to force direct LLM classification of raw strings (expected: accuracy drop concentrated in the 17 Macro-Micro-Macro cases), and peak-memory comparison with `CleanerKS` disabled on the largest workbooks. We report these as future work rather than estimating their outcomes.

# 6 Implementation

## 6.1 The Reference Implementation

The Python reference implementation (the TaxBridge v0.6 engine) comprises the blackboard core (hypothesis store with typed levels, monotonic tick, supersession rebasing, compaction), a controller with per-round LLM batch flushing, and approximately forty Knowledge Sources organised in priority tiers (Table 4). It is exercised by 108 test modules and serves as the formal specification for the production runtime. Deterministic post-processors consume the final board at zero LLM cost: an annotator that colour-codes a copy of the user's own workbook (including a distinct colour for *debatable* classifications) and a template generator that pours the ledger into the appropriate MTD filing template.

Table 4: Knowledge-Source tiers in the reference implementation (LLM-backed sources marked *)

| **Tier** | **Priority** | **Representative Knowledge Sources** |
|---|---|---|
| Memory | 3000 | cleaner |
| Ingestion & lexical | 2000–1700 | cell_ingestor, date_detector, currency_number_parser |
| Spatial assembly | 1500–1400 | structure_detector, table_assembler |
| Fast classification & validation | 1200–1050 | hmrc_category_matcher, fiscal_period_filter, refined_vendor_matcher, vat_aware_amount, double_entry_validator, limited_cost_trader, contra_entry_detector, duplicate_detector |
| Semantic rescue | 450–250 | sheet_representation_builder, header_interpreter*, row_classifier*, sheet_semantics*, supplier_normaliser* |
| Arbiters & residual review | 200–150 | consistency_enforcer, trade_stock_arbiter*, capex_arbiter*, drawings_arbiter*, residual_risk_detector/arbiter*, amount_outlier_detector |
| Advisory & oracle | 100–50 | header_suggestion, tax_adviser (oracle), debatable_pattern, triage_advisor |

## 6.2 The Python–Go Split

The production implementation targets the existing FlowForm stack: a Go runtime utilising Watermill and CloudEvents, PostgreSQL for persistent hypotheses, and Redis for ephemeral states.

- **The Tick** maps natively to Kafka/Watermill Event Sequence Numbers.
- `board.post()` maps natively to publishing a CloudEvent.
- **Knowledge Sources** map natively to independent event-driven microservices.
- **Audit Trail** maps natively to PostgreSQL recursive CTE queries.

## 6.3 FlowForm Integration: Deliberation versus Execution

The blackboard strictly bounds *deliberation*. When a persistent hypothesis at the highest schema level crosses a predefined confidence threshold, the Controller seals the board and emits a **Commit Event**. The blackboard does not execute external side-effects; it hands the Commit Event to the FlowForm runtime, which triggers the physical actions (submitting to HMRC, ordering parts, executing trades). If a physical action fails, FlowForm posts a `FailureEvent` back to the blackboard as new evidence, triggering immediate re-evaluation. The delivery roadmap is summarised in Appendix C.

# 7 Limitations and Threats to Validity

**Single-domain validation.** The empirical results apply only to UK Making Tax Digital financial document reasoning. The four-axis parameterisation suggests applicability to other evidence-driven domains (fault diagnosis, medical triage — §8) but these claims are architectural rather than empirical, and the headline numbers should not be read as predicting equivalent performance elsewhere.

**Synthetic versus production data.** TaxBridge-Bench is generated synthetically with adversarial design choices, and the panel corpus, while modelled on real-world formats, is not raw production data. Performance on genuine production traffic may differ. The architecture is being validated against real HMRC submissions in the imminent production deployment; subsequent work will report those outcomes.

**Baseline scope.** Our baseline is the orchestrated four-role LLM debate panel that previously ran this product — a serious, domain-tuned instance of the LLM-first paradigm, but still one system. We did not benchmark against a re-implementation of an external framework (e.g. AutoGen or LangGraph agents given equivalent domain tooling), and a sufficiently determined engineer could construct stronger LLM-first baselines. The panel's 0% ledger accuracy reflects structural information loss at serialisation that prompt engineering cannot recover, but readers may reasonably wish to verify this with their own baselines; the benchmark harness and prompts are provided (Appendix A).

**Interpretation of token counts.** Blackboard per-case token accounting (≈1,300 tokens) was instrumented after the headline accuracy runs and averaged over a re-run of the same held-out cases; panel token counts were recorded live. Latency figures are benchmark-core timings on a development workstation, not end-to-end production service times.

**Security model assumptions.** The system processes sensitive financial data and submits to an external regulatory API. The architecture provides natural defences against prompt injection — symbolic KSs fire first, LLMs see only ambiguous residual rows, LLM responses are schema-validated against the closed category enumeration, and structural validators detect wholesale reclassifications because totals cease to balance — but we have not subjected the system to formal adversarial security review. A penetration test specifically targeting prompt injection through crafted spreadsheet content is planned before production deployment.

**Scale boundaries.** Empirical results extend to workbook sizes of approximately 50,000 cells. Behaviour at substantially larger scales would require additional engineering — likely partitioning into per-sheet sub-boards with hierarchical aggregation — and is untested.

**LLM provider dependence.** Results were obtained with `claude-sonnet-4-6`; reproduction with different providers or model versions may yield different numbers for the LLM Knowledge Sources. The symbolic Knowledge Sources are fully deterministic and reproducible.

# 8 Discussion and Future Work

## 8.1 Beyond Financial Compliance

The four-axis parameterisation makes the architecture configurable across domains. Two seem particularly natural and are subjects of ongoing work. **Fault diagnosis** (automotive, industrial, IT) is structurally close to TaxBridge: convergent evidence, a need for the `contradicts` polarity (a healthy sensor reading actively weakens a failure hypothesis), and a similar economic argument for symbolic-first processing of routine cases. **Medical triage** offers the strongest test of the HITL primitive: the architecture proposes, the clinician decides, and the audit trail records the exact boundary of agency. We claim no implementation evidence for these domains; they are architectural hypotheses.

### 8.2 Open Architectural Questions

**Persistence boundary as a design choice.** Where to draw the ephemeral/persistent line is currently configured per-domain. There may be domains where the boundary should itself be dynamic — for instance, promoting a hypothesis to persistent when it crosses a confidence threshold.

**Bounty mechanism versus priority queue.** The current implementation uses priority-based scheduling. The recent literature [8] advocates volunteer-based bidding. Priorities buy predictability and auditability (the same board state always schedules identically); volunteering buys open-world flexibility. The trade-offs in latency, scalability, and regulatory defensibility deserve empirical investigation.

**LLM provider abstraction.** The LLM Knowledge Sources are currently coupled to a single provider's API. A multi-provider abstraction with automatic fallback and cross-provider consensus could improve reliability — but may complicate the audit trail.

**Outstanding measurements.** The EVoI-versus-FIFO ablation, the supplier-normaliser ablation, and the `CleanerKS` memory study (§5.5) are instrumented and will accompany the production-deployment report.

## 9 Conclusion

The convergence of three industry trends makes this architecture timely.

**LLM economics.** Production document reasoning at scale is bottlenecked by inference costs. The blackboard's EVoI cost gate, detector–arbiter decomposition, and Macro-Micro-Macro pattern reduce token consumption by an order of magnitude — not by using cheaper models, but by ensuring the LLM is invoked only when symbolic methods cannot resolve ambiguity.

**Regulatory pressure.** The EU AI Act [21] and sector-specific compliance regimes (HMRC MTD, medical-device guidance) demand traceable reasoning. The blackboard's bitemporal event sourcing provides mathematical proof of what the system knew and when.

**Multi-agent scaling.** The orchestrator–worker patterns dominant in 2024–2025 LLM frameworks scale poorly when the orchestrator's prior knowledge of agent capabilities is incomplete. The blackboard's event-driven execution allows dozens of specialist Knowledge Sources — symbolic, neural, and human — to collaborate without a monolithic coordinator.

The deeper claim of this paper is that the dominant trajectory of LLM multi-agent systems — toward larger context windows, more sophisticated orchestration, more capable models — is not the only path to production-grade agentic AI. Our own production history makes the point concretely: the orchestrated four-role LLM debate panel that the blackboard replaced was the sophisticated option, and it was beaten on every axis — accuracy, latency, cost, and auditability — by an architecture that treats the LLM as a bounded heuristic inside a deterministic evidence framework. For high-stakes domains where reliability and economy matter more than open-ended exploration, this inversion is the architecture worth building.

## References


[1] Erman, L. D., Hayes-Roth, F., Lesser, V. R., & Reddy, D. R. (1980). The Hearsay-II speech-understanding system: integrating knowledge to resolve uncertainty. *ACM Computing Surveys*, 12(2), 213–253.

[2] Nii, H. P. (1986). Blackboard systems: the blackboard model of problem solving and the evolution of blackboard architectures (Parts I and II). *AI Magazine*, 7(2), 38–53 and 7(3), 82–106.

[3] Hayes-Roth, B. (1985). A blackboard architecture for control. *Artificial Intelligence*, 26(3), 251–321.

[4] Corkill, D. D., Gallagher, K. Q., & Murray, K. E. (1986). GBB: A generic blackboard development system. In *Proceedings of AAAI-86*, 1008–1014.

[5] Doyle, J. (1979). A truth maintenance system. *Artificial Intelligence*, 12(3), 231–272.

[6] de Kleer, J. (1986). An assumption-based TMS. *Artificial Intelligence*, 28(2), 127–162.

[7] Howard, R. A. (1966). Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, 2(1), 22–26.

[8] Salemi, A., Parmar, M., Goyal, P., Song, Y., Yoon, J., Zamani, H., Palangi, H., & Pfister, T. (2025). LLM-based multi-agent blackboard system for information discovery in data science. *arXiv preprint arXiv:2510.01285*.

[9] Han, B., & Zhang, S. (2025). Exploring advanced LLM multi-agent systems based on blackboard architecture. *arXiv preprint arXiv:2507.01701*.

[10] Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., Awadallah, A. H., White, R. W., Burger, D., & Wang, C. (2023). AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*.

[11] Hong, S., Zheng, X., Chen, J., Cheng, Y., Wang, J., Zhang, C., et al. (2023). MetaGPT: Meta programming for a multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*.

[12] Qian, C., Liu, W., Liu, H., Chen, N., Dang, Y., Li, J., et al. (2024). ChatDev: Communicative agents for software development. In *Proceedings of ACL 2024*. arXiv:2307.07924.

[13] LangChain AI. (2024). LangGraph: Building language agents as graphs. https://github.com/langchain-ai/langgraph.

[14] d'Avila Garcez, A., & Lamb, L. C. (2023). Neurosymbolic AI: the 3rd wave. *Artificial Intelligence Review*, 56, 12387–12406.

[15] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. In *Proceedings of ICLR 2023*. arXiv:2210.03629.

[16] Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., & Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems 36*. arXiv:2302.04761.

[17] Trinh, T. H., Wu, Y., Le, Q. V., He, H., & Luong, T. (2024). Solving olympiad geometry without human demonstrations. *Nature*, 625, 476–482.

[18] Tian, Y., Zhao, J., Dong, H., Xiong, J., Xia, S., Zhou, M., Lin, Y., Cambronero, J., He, Y., Han, S., & Zhang, D. (2024). SpreadsheetLLM: Encoding spreadsheets for large language models. *arXiv preprint arXiv:2407.09025*.

[19] Chen, Y., Yuan, Y., Zhang, Z., Zheng, Y., Liu, J., Ni, F., Hao, J., Mao, H., & Zhang, F. (2025). SheetAgent: Towards a generalist agent for spreadsheet reasoning and manipulation via large language models. In *Proceedings of the ACM Web Conference (WWW) 2025*. arXiv:2403.03636.

[20] Xia, S., Xiong, J., Dong, H., Zhao, J., Tian, Y., Zhou, M., He, Y., Han, S., & Zhang, D. (2024). Vision language models for spreadsheet understanding: challenges and opportunities. In *Proceedings of the 3rd Workshop on Advances in Language and Vision Research (ALVR)*. arXiv:2405.16234.

[21] European Parliament and Council. (2024). Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). *Official Journal of the European Union*, L series, 12 July 2024.

[22] HM Revenue & Customs. (2026). Making Tax Digital for Income Tax: guidance for sole traders and landlords. https://www.gov.uk/government/collections/making-tax-digital-for-income-tax.

# A Reproducibility and Baseline Materials

The benchmark harness (`run_bench.py`, `run_comparative_bench.py`), the TaxBridge-Bench generator and ground-truth master files, the strict scorer (£0.01 box tolerance, dynamic denominator scaling, fungible-box equivalence groups), and the full prompts used by both the blackboard's LLM Knowledge Sources and the four-role debate-panel baseline ship with the reference implementation and are available to reviewers on request. Per-case raw results for the held-out runs (JSON) include latency, token counts, per-box errors, and per-KS firing diagnostics.

# B TaxBridge-Bench Generation

Each workbook is generated from a case manifest specifying trade type, VAT registration and scheme, transaction count, and adversarial features: table origin offsets are drawn at random; aggregate rows use live formulas rather than values; a stochastic subset of personal or suspicious entries is flagged only by background fill colour; and out-of-period transactions are injected around the quarter boundaries. Ground truth maps each case to official HMRC SA103F box values. The held-out split (cases 031–060) was generated after the architecture was frozen and scored blind; the scorer's dynamic denominator scaling prevents a system from inflating its score by emitting fewer boxes.

# C Implementation Status and Roadmap

**Phase 1: `blackboard-core` (Go).** The domain-agnostic engine: hypothesis event store, tick ordering, controller loop, LLM batch, and DAG audit trails.

**Phase 2: TaxBridge Migration.** Port the Python Knowledge Sources to Go, validating the framework against a production workload with real HMRC submissions.

**Phase 3: FlowForm Integration.** Expose the blackboard as a configurable FlowForm node type with YAML schema definitions.

**Phase 4: Second Domain.** Implement an IT or automotive fault-diagnosis vertical to validate against a convergent/causal evidence structure.

The Python reference implementation is production-hardened in its own right (fail-closed LLM semantics, Prometheus/OpenTelemetry observability with PII redaction, budgeted free-tier readiness checks) and currently drives the TaxBridge product ahead of HMRC production approval.

# D CleanerKS Source

The complete memory-management Knowledge Source from the reference implementation:

```
class CleanerKS(KnowledgeSource):
    """Garbage collector that triggers board compaction when superseded
    hypotheses accumulate beyond a configurable threshold."""

    name = "cleaner"
    priority = 3000  # Highest priority -- runs first if threshold breached

    def __init__(self, superseded_threshold: int = 500) -> None:
        super().__init__()
        self.superseded_threshold = superseded_threshold

    @property
    def is_llm(self) -> bool:
        return False

    def precondition(self, board: Blackboard) -> bool:
        live_superseded = sum(
            1 for hid in board._superseded if hid in board._hypotheses
        )
        return live_superseded >= self.superseded_threshold

    def execute(self, board: Blackboard) -> int:
        initial_count = len(board._hypotheses)
        result = board.compact()
        final_count = len(board._hypotheses)
        removed = initial_count - final_count
        print(
            f"[cleaner] Compacted board: purged {result['deleted']} ephemeral, "
            f"archived {result['archived']} persistent, "
            f"hypotheses {initial_count} -> {final_count} (-{removed})",
            flush=True,
        )
        self.mark_processed(board)
        return 0  # Cleaner doesn't post new hypotheses
```

Compaction respects the level schema automatically: `compact()` hard-deletes superseded hypotheses at ephemeral levels and archives those at persistent levels with DAG topology intact, so the audit trail survives garbage collection by construction.