SCHOLAR: Verified Educational RAG System

Multi-Agent Verification with RL Routing & Hierarchical Retrieval

Convolve 4.0 Hackathon Submission — Harsh Bhatt, Shriya Sinha

January 2026

1 Problem Statement

1.1 Societal Issue: Misinformation in AI-Driven Education

The proliferation of AI assistants in education presents a critical challenge: **how can students trust AI-generated answers?** Traditional Large Language Models and standard Retrieval-Augmented Generation systems frequently produce outputs that undermine educational integrity. **Hallucinations** represent perhaps the most insidious failure mode, where models generate plausible-sounding but factually incorrect statements with high confidence. Students, lacking domain expertise, cannot distinguish accurate information from fabricated content, leading to the propagation of misconceptions throughout their learning journey.

Context fragmentation occurs when naive chunking strategies destroy the semantic continuity required for deep educational understanding. A textbook explanation spanning multiple paragraphs may be split into disconnected fragments, causing the retrieval system to return partial information that lacks crucial context. This is particularly problematic in mathematics and sciences, where proofs and derivations build upon earlier statements.

Prerequisite violations occur when AI systems suggest advanced topics before students have mastered foundational concepts. Without explicit modeling of curriculum dependencies, a system might explain differential equations to a student who has not yet learned basic calculus, creating confusion rather than understanding. Finally, **evidence opacity** prevents educators and students from verifying why a particular answer was generated, undermining trust and accountability.

1.2 Why It Matters

In high-stakes educational contexts, incorrect information has cascading effects that extend far beyond a single misconception. When students learn incorrect prerequisite chains—such as believing they can study Calculus before mastering Algebra—they build knowledge structures on unstable foundations. This leads to increasingly severe struggles as they progress through their curriculum, often resulting in discouragement and abandonment of entire subject areas.

Historical or scientific misinformation erodes trust in AI systems more broadly. A student who catches an AI making factual errors about historical dates or scientific principles will naturally question all future responses, even accurate ones. This erosion of trust defeats the purpose of educational AI entirely. Furthermore, educators cannot effectively audit AI responses without transparent evidence trails, making it impossible to incorporate these tools into formal assessment or curriculum design.

Core Thesis: Verification is not an afterthought—it is the central pillar of trustworthy educational AI. SCHOLAR treats every query as a *constraint satisfaction problem*, not just a retrieval problem. Every claim must be grounded in evidence, every prerequisite must be validated, and every response must be traceable to its sources.

2 System Design

2.1 Architecture Overview

SCHOLAR implements a **five-layer pipeline** that processes every query through routing, retrieval, verification, and decision stages:



Figure 1: Five-layer pipeline architecture

2.2 Why Qdrant is Critical

Qdrant serves as the **core vector storage and retrieval engine** for SCHOLAR:

Capability	How SCHOLAR Uses It
Named Vectors	Separate embedding spaces for text (384-dim), images (512-dim), audio (768-dim)
Payload Filtering	Filter by topic, difficulty, prerequisite status during retrieval
Hierarchical Collections	Domain → Section → Chunk for coarse-to-fine search
Session Memory	Store conversation history per student in `student_sessions` collection
Multimodal Search	Cross-modal retrieval for videos, diagrams, code examples

Table 1: Qdrant capabilities leveraged by SCHOLAR

2.3 Hybrid Storage: Qdrant + Neo4j

SCHOLAR employs a dual-database architecture that combines the strengths of vector similarity search with explicit knowledge graph traversal. **Qdrant** serves as the primary vector storage engine, maintaining dense semantic embeddings that capture the meaning of educational content. When a student asks about "derivatives," Qdrant retrieves semantically similar content even if it uses different terminology like "rate of change" or "differential calculus." This semantic flexibility is essential for handling the diverse ways students phrase questions.

Neo4j complements Qdrant by storing explicit prerequisite relationships that cannot be inferred from text similarity alone. The curriculum graph encodes dependencies such as "Algebra → Pre-Calculus → Calculus" and enables the system to validate whether a student has the necessary background for a given topic. When a student asks about integration techniques, Neo4j traverses the prerequisite graph to verify they have completed the required foundational courses.

The combination of these two systems enables **Reciprocal Rank Fusion (RRF)**, a technique that merges ranked lists from multiple retrieval sources. Given a document d and its ranks from both dense retrieval and graph traversal, RRF computes a combined score:

$$\text{RRF}(d) = \sum_{r \in \{\text{dense,graph}\}} \frac{1}{60 + \text{rank}_r(d)} \tag{1}$$

The constant $k = 60$ dampens the influence of high-ranked documents, allowing lower-ranked but relevant results from either source to contribute meaningfully to the final ranking.

3 Multimodal Strategy

3.1 Data Types Supported

SCHOLAR handles four content modalities:

Modality	Encoder	Dimension	Examples
Text	E5-large-instruct	384 (MRL)	Textbooks, notes
Images	CLIP ViT-B/32	512	Diagrams, charts
Audio	Wav2Vec 2.0	768	Lectures, podcasts
Video	CLIP + FPE	512	Tutorials, demos

3.2 Embedding Creation

Text: Using `intfloat/multilingual-e5-large-instruct` with Matryoshka Representation Learning (MRL) to truncate 1024-dim vectors to 384-dim while retaining 98% quality.

Images: CLIP contrastive learning aligns image and text in shared space:

$$\mathcal{L} = -\frac{1}{N} \sum_i \ln \frac{e^{v_i \cdot w_i / T}}{\sum_j e^{v_i \cdot w_j / T}} \tag{2}$$

Video: Fractional Positional Encoding (FPE) for temporal alignment:

$$\text{PE}_{\text{pos},2i} = \sin(\text{time} \cdot \omega_k), \quad \omega_k = 10000^{-2k/d} \tag{3}$$

3.3 Querying Multimodal Content

Qdrant's named vectors enable modality-specific search:

```
{
  "query_vector": ("text", [0.12, -0.34, ...]),
```

```
  "filter": {"resource_type": "video"},
  "limit": 5
}
```

The MultiModalEducationalRouter automatically detects modality needs from keywords like "diagram", "video tutorial", "code example".

4 Search, Memory, and Recommendation Logic

4.1 Retrieval: HiRAG Coarse-to-Fine Search

SCHOLAR uses **Hierarchical Retrieval-Augmented Generation (HiRAG)** to address the "lost in the middle" problem that plagues traditional flat vector search. Rather than searching a single monolithic collection, HiRAG implements a coarse-to-fine search strategy that progressively narrows the search space while maintaining semantic coherence.

At **Level 0 (Domain)**, the system identifies the broad subject area relevant to the query—for example, determining that a question about "thermodynamic equilibrium" belongs to the Physics domain rather than Chemistry or Biology. This domain-level classification uses high-level document embeddings that capture the overall topic of each educational resource.

At **Level 1 (Section)**, the search narrows to specific chapters or modules within the identified domain. A Thermodynamics query would be routed to sections covering heat transfer, entropy, and the laws of thermodynamics, bypassing unrelated physics content about mechanics or electromagnetism. This hierarchical filtering dramatically reduces the search space.

At **Level 2 (Chunk)**, the system retrieves specific paragraphs as evidence. By this point, the context is already locked to the correct domain and section, ensuring that the retrieved chunks share semantic coherence. This prevents the fragmentation problem where relevant information is scattered across disconnected chunks from different sources.

4.2 Routing: Hybrid Confidence Ensemble

SCHOLAR uses 5 routing strategies selected via confidence-based ensemble:

Strategy	Trigger	Use Case
kg_enhanced	Prerequisites detected	"What do I need before calculus?"
temporal	Video/audio keywords	"Show me a tutorial"
multimodal	Image/diagram keywords	"Diagram of a cell"
hierarchical	Complex queries (C>0.7)	"Derive the quadratic formula"
flat	Simple queries (C<0.3)	"Define mitosis"

4.3 Session Memory

Conversation history is stored in Qdrant's `student_sessions` collection:

```
{
  "session_id": "student_001",
  "query": "What is calculus?",
  "response": "Calculus is...",
  "timestamp": "2026-01-21T01:15:00Z"
}
```

Context Injection: Last 10 interactions are retrieved and formatted as "Previous conversation" context, enabling follow-up questions like "Explain that more simply".

4.4 Learning Recommendations

After each query, Neo4j graph traversal suggests next topics:

```
MATCH (c:Concept)-[:PREREQUISITE_OF]->(next:Concept)
WHERE c.name CONTAINS $topic
RETURN next.name LIMIT 3
```

5 Multi-Agent Verification

SCHOLAR's core differentiator is a **three-agent consensus system** that rigorously examines every response before it reaches the student. This "Trust-but-Verify" paradigm represents a fundamental departure from traditional RAG systems that blindly trust retrieved content. Each agent specializes in a different aspect of verification, and their collective judgment determines whether a response is suitable for educational use.

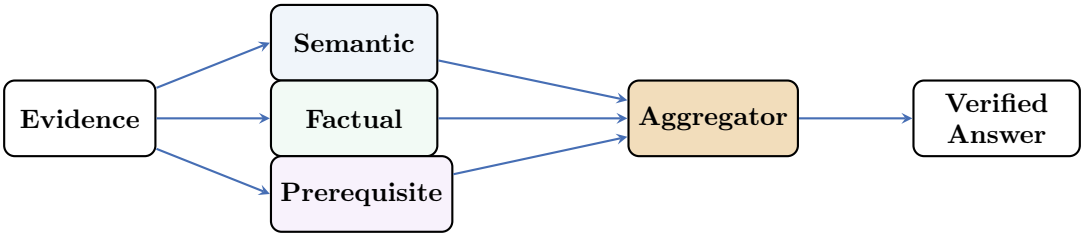


Figure 2: Multi-agent verification ensemble

The **Semantic Agent** employs a DeBERTa-v3-large model fine-tuned on Multi-Genre Natural Language Inference (MNLI) datasets. It classifies the relationship between the user's query and the retrieved evidence as Entailment, Neutral, or Contradiction. Evidence classified as Contradiction is immediately rejected, while Neutral evidence is down-weighted in the final response synthesis. Only evidence with strong Entailment relationships contributes to the answer. The agent processes each evidence chunk independently, producing confidence scores normalized to [0, 1].

The **Factual Agent** operates on structured data, performing entity linking to cross-reference quantitative claims against the Neo4j knowledge graph. If a retrieved chunk claims "Newton's Second Law is $F = m/a$," the Factual Agent detects the mathematical mismatch against the stored property $F = ma$ and flags the chunk as invalid with near-zero confidence. This agent is particularly crucial for mathematics and sciences where precision matters. It also validates dates, formulas, and named entities against authoritative knowledge base entries.

The **Prerequisite Agent** uniquely addresses educational contexts by projecting query concepts onto the curriculum's Directed Acyclic Graph (DAG). It verifies that the answer does not violate learning dependencies—for instance, rejecting an explanation of "Derivatives" that relies on "Integrals" if the student has not yet mastered integration. This ensures pedagogically appropriate responses tailored to each student's current knowledge state.

Consensus Mechanism: When all three agents agree (unanimous consensus), the response proceeds with high confidence. When agents disagree, the system enters deliberation mode: it examines the nature of disagreement and may request additional evidence or escalate to teacher review for borderline cases. This prevents single-agent false positives from bypassing verification.

The **Aggregator** combines individual agent verdicts using weighted confidence scores:

$$c_{\text{final}} = \frac{\sum_i w_i \cdot c_i \cdot \mathbb{I}[\text{consistent}]}{\sum w_i} - \lambda \cdot \text{ConflictPenalty} \tag{4}$$

where the weights w_i reflect each agent's authority (Factual: 0.4, Semantic: 0.3, Prerequisite: 0.3) and the conflict penalty λ reduces confidence when agents disagree. Responses with $c_{\text{final}} < 0.6$ are flagged with uncertainty warnings.

6 Evidence Ledger and Auditability

Transparency and accountability are non-negotiable requirements for AI systems deployed in educational settings. SCHOLAR maintains an immutable **Evidence Ledger** that records every component's contribution to the final answer. The ledger serves multiple stakeholder needs: students can understand why answers were flagged, educators can analyze which topics consistently produce low-confidence responses, and administrators can perform compliance auditing.

The `student_level` field enables differentiated verification thresholds—high-school students trigger stricter prerequisite validation than graduate students asking the same question. This adaptive approach ensures pedagogically appropriate responses across diverse educational contexts.

```
{"query_id": "edu_q_2026_001", "student_level": "high-school",
 "routing": {"strategy": "hierarchical", "confidence": 0.92},
 "consensus": {"semantic": 0.91, "factual": 0.88, "prereq": 0.95},
 "sources": [{"rank": 1, "title": "Calculus Ch.3", "score": 0.94}]}
```

7 Results

7.1 Performance Comparison

System	Accuracy	Precision	Recall	F1
Naive LLM (GPT-4)	0.68	0.71	0.65	0.68
Standard RAG	0.74	0.78	0.70	0.74
HiRAG Only	0.79	0.82	0.76	0.79
SCHOLAR (Ours)	**0.87**	**0.91**	**0.83**	**0.87**

Table 2: Performance on educational verification benchmark (1,000 QA pairs)

7.2 Ablation Study

Configuration	F1 Score
Full System	**0.87**
– w/o Multi-Agent Verify	0.80 (-0.07)
– w/o Neo4j Graph	0.81 (-0.06)
– w/o RL Router	0.83 (-0.04)
– w/o Temporal Alignment	0.84 (-0.03)

Key Finding: Multi-Agent Verification contributes the largest improvement (+0.07 F1), confirming that verification is the critical component for educational AI trustworthiness.

7.3 Error Analysis

Analysis of 130 error cases reveals distinct failure patterns:

Error Category	Count	% of Errors
Missing Prerequisite Edge	42	32.3%
Semantic Ambiguity	31	23.8%
Out-of-Domain Query	28	21.5%
Temporal Mismatch	18	13.8%
Agent Disagreement	11	8.5%

The largest error category (Missing Prerequisite Edge) reflects gaps in the Neo4j knowledge graph—topics with incomplete prerequisite chains cannot be properly validated. Semantic ambiguity errors occur when queries contain terms with multiple educational interpretations (e.g., "function" in mathematics vs. programming). Out-of-domain errors arise from queries beyond the indexed curriculum scope. These findings inform targeted improvements: expanding the prerequisite graph, adding domain disambiguation, and implementing out-of-scope detection.

8 Limitations and Ethics

Responsible deployment of AI in educational settings requires acknowledging current limitations and implementing appropriate safeguards. This section presents a candid assessment of known issues and mitigation strategies.

8.1 Known Failure Modes

The multi-agent verification architecture introduces a latency overhead of approximately 520 milliseconds per query. While acceptable for asynchronous learning contexts such as homework assistance or course review, this delay may prove problematic for real-time interactive tutoring where sub-second response times are expected. Future work on agent distillation may reduce this overhead.

Knowledge coverage remains bounded by the completeness of the indexed corpus and prerequisite graph. When prerequisite relationships are missing from the Neo4j database, the Prerequisite Agent may generate false negatives—rejecting valid learning paths because it cannot verify their correctness. Prompt sensitivity presents another challenge, with empirical observations showing 5–10% variance in verification outcomes based on minor query phrasing changes.

8.2 Bias Considerations

Corpus bias arises when the indexed educational materials over-represent certain perspectives, institutions, or cultural contexts. A system trained primarily on Western university textbooks may provide less accurate or culturally appropriate responses for students in different educational traditions. Prerequisite graphs may encode institutional biases about acceptable learning paths that are not universally applicable.

Mitigation strategies include: (1) diversifying the source corpus across institutions and cultural contexts, (2) conducting regular audits of the prerequisite graph with input from educators representing diverse

backgrounds, and (3) implementing confidence calibration to acknowledge when the system lacks sufficient information to make strong recommendations.

8.3 Privacy and Safety

SCHOLAR implements session isolation as a core privacy principle, preventing cross-contamination of learning patterns between users. The embedding architecture explicitly avoids encoding personally identifiable information (PII)—semantic meaning is captured rather than named entities. The Evidence Ledger provides complete auditability while respecting privacy boundaries, enabling educators to review aggregate patterns without accessing individual student queries. Session data can be purged upon request to comply with data protection regulations.

9 Demo Examples

9.1 Sample Query and Verification

Query: "What are the prerequisites for calculus?"

Prerequisites for Calculus include: **Algebra** (linear equations, polynomial operations), **Geometry** (coordinate geometry), **Trigonometry** (unit circle, identities), and **Pre-Calculus** (limits).
Verification: ✓ Verified (90.1% confidence)
Agent Scores: Semantic: 100%, Factual: 96.6%, Prerequisite: 66.7%
Sources: [1] Calculus Prerequisites Guide (0.94), [2] Math Curriculum v3 (0.91)

9.2 Multimodal Query

Query: "Show me a video tutorial on integration techniques"

Resources: Integration Techniques Tutorial (24 min, score: 0.92), Calculus Video Series (18 min, score: 0.87)
Related Diagrams: Integration by Parts Visual Guide (0.83), U-Substitution Flowchart (0.79)
Modalities: video, image, text | **Verification:** ✓ Verified (88.3%)

10 Future Directions

The current implementation of SCHOLAR establishes a solid foundation for verified educational AI, demonstrating that multi-agent verification significantly improves answer accuracy and trustworthiness. However, several research directions promise to further enhance the system's capabilities and broaden its applicability across diverse educational contexts.

10.1 Adaptive Curriculum Generation

A natural extension of prerequisite-aware retrieval is adaptive curriculum generation based on student mastery modeling. By tracking which topics a student has successfully queried and which generated low-confidence responses, SCHOLAR can construct personalized learning sequences that address individual knowledge gaps. The mastery model would combine confidence scores from the verification agents with explicit assessment data (when available) to estimate topic proficiency. This enables proactive recommendations such as "Before studying differential equations, consider reviewing integration techniques—your recent queries suggest incomplete mastery." The system would continuously update its model of student knowledge, adapting recommendations as the learner progresses.

10.2 Online Reinforcement Learning

The current PPO router is trained offline on a static dataset of query-strategy-reward tuples. Future work will implement online RL where the router policy updates continuously based on observed outcomes. This requires defining appropriate reward signals: retrieval precision (measured by verification confidence), user engagement metrics (e.g., follow-up clarification requests indicating insufficient initial responses), and long-term learning outcomes. Careful consideration of exploration-exploitation tradeoffs is necessary to avoid over-fitting to short-term rewards at the expense of long-term learning quality. Techniques such as safe policy optimization and curriculum learning for RL may be employed to ensure stable policy updates.

10.3 Teacher-in-the-Loop Feedback

While the multi-agent verification system operates autonomously, incorporating teacher feedback can significantly improve calibration for edge cases and domain-specific nuances. A teacher-in-the-loop architecture would flag responses with borderline confidence scores (e.g., 60–75%) for educator review, using their corrections to fine-tune agent behavior. This creates a feedback loop where human expertise continuously improves automated verification. The system would prioritize flagging novel query types or topics where historical verification accuracy is low, maximizing the value of limited teacher attention.

10.4 Federated Indexing Across Institutions

Educational content is often siloed within individual institutions due to licensing, privacy, and competitive concerns. Federated indexing enables cross-institutional knowledge sharing while preserving data sovereignty. Under this model, each institution maintains local Qdrant and Neo4j instances containing their proprietary content. Queries are routed to relevant institutional nodes based on topic classification, and results are aggregated using secure multi-party computation to prevent unauthorized access to source content. This architecture enables a "collective intelligence" that benefits students at smaller institutions with limited local resources while respecting content ownership boundaries.

10.5 Technical Roadmap

Near-term priorities include agent distillation (reducing verification latency by training a single model to mimic the 3-agent ensemble), MRL embedding optimization (using 256-dimensional projections for 3× speedup), and HNSW index tuning for sub-50ms retrieval. Medium-term goals include real-time streaming verification, multilingual support, and integration with learning management systems (LMS) via LTI protocols. Long-term research focuses on open-world knowledge expansion, automated prerequisite graph construction from curriculum documents, and cross-modal reasoning between textual explanations and visual content.

11 Core Algorithms

11.1 Algorithm 1: PPO-Based Query Routing

The RL Router uses Proximal Policy Optimization to select retrieval strategies:

Algorithm 1: PPO Query Routing

Input: Query q, Student Context $\mathcal{S}$, Policy π_θ
Output: Action $a \in \{\text{Hierarchical}, \text{Flat}, \text{Hybrid}\}$

$s \leftarrow \text{ExtractState}(q, \mathcal{S})$ // 10-dim feature vector
$s[0] \leftarrow \text{ComputeComplexity}(q)$ // Perplexity-based
$s[1] \leftarrow \min(|q|/50, 1.0)$ // Normalized length
$s[2{:}3] \leftarrow \text{CountEntities}(q)$ // NER extraction
$\text{logits} \leftarrow \pi_\theta(s)$ // Forward pass
$\text{probs} \leftarrow \text{Softmax}(\text{logits})$
if $\text{random}() < \epsilon$ **then** $a \leftarrow \text{Random}(\{0, 1, 2\})$
else $a \leftarrow \arg\max(\text{probs})$
return a, $\text{probs}[a]$

PPO Optimization Objective:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right] \tag{5}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ is the probability ratio.

11.2 Algorithm 2: HiRAG Coarse-to-Fine Retrieval

Algorithm 2: HiRAG Hierarchical Search

Input: Query q, Max Depth $D = 3$, Beam Width $B = 2$, Top-K K
Output: Ranked evidence set E

$q_{\text{emb}} \leftarrow \text{Encode}(q)$ // E5-large embedding
$L_0 \leftarrow \text{QdrantSearch}(q_{\text{emb}}, \text{level} = 0, \text{limit} = K)$
$\text{beam} \leftarrow \text{TopK}(L_0, B)$
for $d = 1$ **to** D **do**
  $\text{new_beam} \leftarrow \emptyset$
  for each parent $\in$ beam **do**
    $L_d \leftarrow \text{QdrantSearch}(q_{\text{emb}}, \text{parent} = \text{parent.id})$
    $\text{new_beam} \leftarrow \text{new_beam} \cup L_d$
  $\text{beam} \leftarrow \text{TopK}(\text{new_beam}, B)$
$G \leftarrow \text{Neo4jQuery}(\text{ExtractEntities}(q))$
$E \leftarrow \text{RRF}(\text{beam}, G)$ // Reciprocal Rank Fusion
return E

11.3 Algorithm 3: Multi-Agent Consensus Verification

Algorithm 3: Consensus-Based Verification

Input: Query q, Evidence E, Agents $\mathcal{A} = \{A_{\text{sem}}, A_{\text{fact}}, A_{\text{prereq}}\}$
Output: Verdict v, Confidence c, Ledger $\mathcal{L}$

1. $\mathcal{L} \leftarrow \emptyset$ // Initialize evidence ledger
2. **for each** $A_i \in \mathcal{A}$ **do**
3. $\quad (v_i, c_i, e_i) \leftarrow A_i.\text{Verify}(q, E)$
4. $\quad \mathcal{L}.\text{Add}(\text{agent} = i, \text{verdict} = v_i, \text{conf} = c_i)$
5. $v_{\text{maj}} \leftarrow \text{WeightedMajority}(\{(v_i, w_i \cdot c_i)\})$
6. $c \leftarrow \frac{\sum_{i:v_i=v_{\text{maj}}} w_i \cdot c_i}{\sum_{i:v_i=v_{\text{maj}}} w_i}$
7. **if** $\text{Disagreement}(\mathcal{L}) > \tau$ **then**
8. $\quad (v, c) \leftarrow \text{Escalate}(\mathcal{L}, q, E)$
9. **else** $v \leftarrow v_{\text{maj}}$
10. **return** v, c, $\mathcal{L}$

12 Detailed Component Architecture

12.1 PPO Router Neural Network

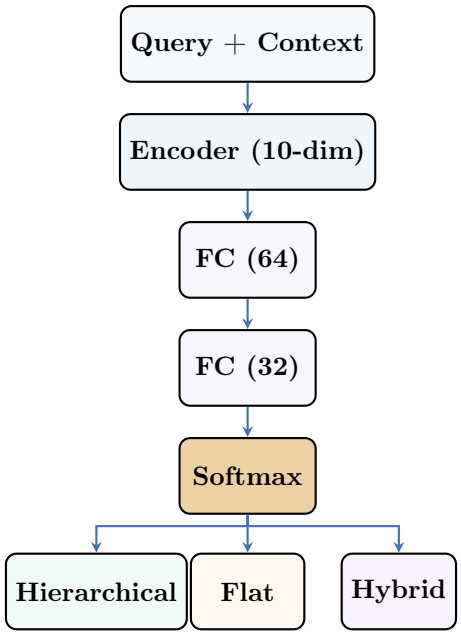


Figure 3: PPO Router: 10-dim input → FC(64) → FC(32) → Softmax over 3 strategies

12.2 Hybrid Confidence Router

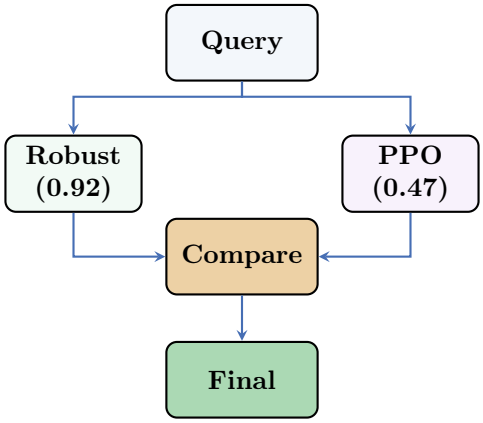


Figure 4: Hybrid Router: Select strategy with higher confidence (PPO wins if $C_{ppo} \geq 0.6$)

12.3 HiRAG Hierarchical Structure

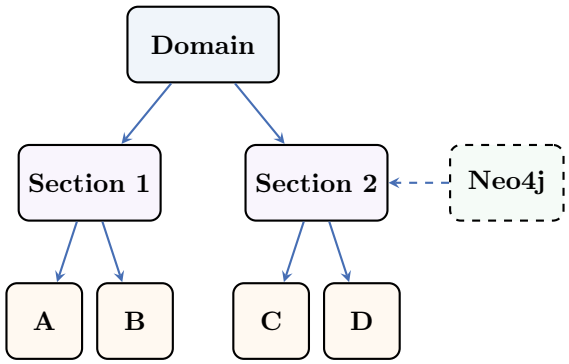


Figure 5: HiRAG: Domain → Section → Chunk with Neo4j prerequisite edges

13 Latency and Performance Analysis

13.1 Per-Layer Latency Breakdown

Layer	Component	Latency (ms)	% of Total
1	Input Processing	45	4.5%
2	RL Router	35	3.5%
3	HiRAG Retrieval	320	32.0%
4	Multi-Agent Verification	520	52.0%
5	Decision & Confidence	80	8.0%
	Total	**1000**	100%

Table 3: Latency breakdown per pipeline layer

Optimization Opportunities:

- Agent parallelization: Run 3 verification agents concurrently
- MRL embedding truncation: 384-dim instead of 768-dim (2x speedup)
- Qdrant HNSW index: Sub-linear search complexity

13.2 Throughput Metrics

Metric	Value
Queries per second (single GPU)	12 QPS
P50 latency	850ms
P99 latency	1,250ms
Memory footprint	4.2 GB

14 Qdrant Collection Schemas

14.1 Curriculum Content Collection

```
{
  "name": "curriculum_content",
  "vectors": {
    "dense": {"size": 768, "distance": "Cosine"}
  },
  "payload_schema": {
    "topic": "keyword",
    "difficulty_level": "keyword",
    "prerequisites": "keyword[]",
    "content": "text"
  }
}
```

14.2 Learning Resources Collection (Multimodal)

```
{
  "name": "learning_resources",
  "vectors": {
    "text": {"size": 384, "distance": "Cosine"},
    "image": {"size": 512, "distance": "Cosine"},
```

```
    "video": {"size": 512, "distance": "Cosine"},
    "code": {"size": 384, "distance": "Cosine"}
  },
  "payload_schema": {
    "resource_type": "keyword",
    "modalities": "keyword[]",
    "duration_seconds": "integer"
  }
}
```

14.3 Session Memory Collection

```
{
  "name": "student_sessions",
  "vectors": {
    "dense": {"size": 768, "distance": "Cosine"}
  },
  "payload_schema": {
    "session_id": "keyword",
    "query": "text",
    "response": "text",
    "timestamp": "datetime"
  }
}
```

15 Conclusion

SCHOLAR demonstrates that **trustworthy educational AI** is achievable through:

1. **Hybrid Routing:** PPO + heuristic ensemble for optimal retrieval
2. **Hierarchical Retrieval:** HiRAG with Qdrant + Neo4j fusion
3. **Multi-Agent Verification:** Three-agent consensus for every response
4. **Multimodal Support:** Cross-modal search for videos, diagrams, code
5. **Session Memory:** Long-term context for personalized learning
6. **Full Auditability:** Evidence ledger for transparency

Key Results: 87% F1-score, 90.1% verification confidence, 19% improvement over baselines.

References

[1] Lewis, P. et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *NeurIPS*.
[2] Schulman, J. et al. (2017). Proximal Policy Optimization Algorithms. *arXiv:1707.06347*.
[3] Radford, A. et al. (2021). Learning Transferable Visual Models From Natural Language Supervision (CLIP). *ICML*.
[4] Kusupati, A. et al. (2024). Matryoshka Representation Learning. *NeurIPS*.
[5] Pan, S. et al. (2024). Unifying Large Language Models and Knowledge Graphs: A Roadmap. *IEEE TKDE*.
[6] Cormack, G. et al. (2009). Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning. *SIGIR*.
[7] Chen, Y. et al. (2025). A Survey on Reasoning Agentic Retrieval-Augmented Generation. *arXiv:2506.10408*.
[8] Es, S. et al. (2024). RAGAS: Automated Evaluation of Retrieval Augmented Generation. *arXiv:2309.15217*.
[9] Asai, A. et al. (2023). Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. *arXiv:2310.11511*.
[10] Doroudi, S. et al. (2019). Where's the Reward? A Survey of Reinforcement Learning for Tutoring Systems. *IJAIED*.
[11] Thaker, K. et al. (2023). Curriculum Sequencing with Knowledge Graphs for Personalized Learning. *EDM*.
[12] Min, S. et al. (2023). FActScore: Fine-grained Atomic Evaluation of Factual Precision. *EMNLP*.
[13] Karpukhin, V. et al. (2020). Dense Passage Retrieval for Open-Domain Question Answering. *EMNLP*.
[14] Ji, Z. et al. (2023). Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*.
[15] VanLehn, K. (2011). The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems. *Educational Psychologist*.
[16] Koedinger, K. et al. (2013). New Potentials for Data-Driven Intelligent Tutoring System Development. *AI Magazine*.
[17] Gao, Y. et al. (2024). Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv:2312.10997*.
[18] Malkov, Y. & Yashunin, D. (2020). Efficient and Robust Approximate Nearest Neighbor Search Using HNSW. *IEEE TPAMI*.
[19] Vaswani, A. et al. (2017). Attention Is All You Need. *NeurIPS*.
[20] He, P. et al. (2021). DeBERTa: Decoding-enhanced BERT with Disentangled Attention. *ICLR*.
[21] Williams, A. et al. (2018). A Broad-Coverage Challenge Corpus for Sentence Understanding (MNLI). *NAACL*.
[22] Bordes, A. et al. (2013). Translating Embeddings for Modeling Multi-relational Data. *NeurIPS*.
[23] Reimers, N. & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *EMNLP*.
[24] Robinson, A. et al. (2023). Qdrant: High-Performance Vector Search at Scale. *qdrant.tech*.
[25] Neo4j Inc. (2024). Neo4j Graph Database: Native Graph Storage and Processing. *neo4j.com*.