OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

DeepRAG: A Conceptual Multi-Agent Retrieval-Augmented Generation Framework for Autonomous and Verifiable Research Synthesis

Prof. DhananjayRaut[1], Khushali Gatir[2], Aryan Balani[3], Harshala Gaykar[4], Dhanraj Banglurkar[5]

[1]*Professor, Watumull Institute of Engineering and Technology* Thane, India
[2,3,4,5] *Computer Engineering, Watumull Institute of Engineering and Technology* Thane, India

Abstract
In today's world of information overload, researchers face increased difficulty in identifying reliable, verified, and up-to-date knowledge from vast online data sources. While Large Language Models (LLMs) and traditional Retrieval-Augmented Generation (RAG) systems support factual retrieval, they remain limited by single-agent reasoning, lack of integrated validation, and dependence on static or outdated knowledge bases. This paper proposes DeepRAG, a conceptual, implementation-ready multi-agent Retrieval-Augmented Generation framework designed to support autonomous and verifiable research synthesis. DeepRAG introduces a coordinated set of specialised agents for planning, semantic retrieval, real-time web data acquisition, citation validation, knowledge synthesis, and structured report generation. The framework leverages agentic orchestration mechanisms (e.g., Autogen), vector-based retrieval using ChromaDB, and real-time web crawling via Crawl4AI to address core limitations of existing RAG pipelines. Rather than presenting empirical benchmarks, this work focuses on architectural design, agent-level responsibilities, and workflow specification, demonstrating how DeepRAG is intended to generate citation-backed research reports in PDF format using the FPDF library. Design-level reasoning indicates that the proposed framework addresses challenges related to factual grounding, citation consistency, and transparency in AI-assisted research. DeepRAG thus provides a robust foundation for future prototype development, experimental evaluation, and deployment of autonomous research assistants.

Keywords—*Retrieval-Augmented Generation (RAG), Agentic AI, Multi-Agent Systems, Autogen, Crawl4AI, ChromaDB, FPDF, Citation Validation, AI Research Automation.*

Date of Submission: 08-04-2026 Date of acceptance: 20-04-2026

I. INTRODUCTION

The rapid growth of digital information access has transformed how researchers, students, and professionals discover, analyze, and synthesize knowledge. Advances in Large Language Models (LLMs), such as GPT, Gemini, and Claude-based systems, have significantly improved the speed and accessibility of information retrieval and content generation. However, despite their linguistic fluency and contextual understanding, these models often generate unverifiable or hallucinated content, limiting their reliability for academic and research-oriented applications.

To mitigate these limitations, Retrieval-Augmented Generation (RAG) architectures were introduced by combining information retrieval mechanisms with LLM-based generation to improve factual grounding. While this paradigm represented a notable advancement, many existing RAG systems continue to fall short in providing real-time, verifiable, and citation-supported outputs. The growing demand for trustworthy, autonomous, and context-aware research assistance highlights the need for more advanced architectures capable of addressing these shortcomings. This motivates the development of DeepRAG, a multi-agent RAG-based research automation framework.

Traditional RAG and LLM-driven systems typically rely on static or pre-indexed knowledge sources that can become outdated over time. Such systems often lack mechanisms to retrieve the most recent information or verify the authenticity of external data sources. Furthermore, most current RAG implementations operate

through single-agent or linear pipelines, where retrieval, processing, and generation tasks are executed sequentially. This architectural rigidity limits scalability, adaptability, and collaborative reasoning across complex research workflows.

In addition, many AI-based research systems do not effectively maintain citation consistency or enforce systematic data validation. Without structured cross-verification and traceability, ensuring the credibility of generated research content remains challenging. The absence of real-time validation mechanisms reduces confidence in AI-assisted academic outputs and underscores the need for a more dynamic, modular, and autonomous approach capable of coordinating multiple specialized agents to enhance transparency and factual accuracy.

Reliable research fundamentally depends on the credibility and traceability of the information being used. Modern research environments increasingly require systems that can not only retrieve relevant data but also validate, contextualize, and synthesize it in a coherent and structured manner. An autonomous and verifiable knowledge retrieval framework should therefore integrate real-time web information, assess source credibility, and enrich internal knowledge representations to support ongoing and future research tasks.

By integrating agent-based artificial intelligence principles with retrieval-augmented generation, it becomes possible to design a framework capable of managing the entire research pipeline—from query planning and information acquisition to synthesis and report generation. Such an approach is intended to deliver accurate, up-to-date, and transparent research outputs while significantly reducing the manual effort involved in literature review and data validation.

Existing retrieval-based AI solutions continue to face challenges in autonomously synthesising reliable and verifiable research content. Many lack effective coordination between functional components, fail to combine real-time web knowledge with vector-based retrieval, or omit systematic content validation. These limitations often result in fragmented, inconsistent, or partially unverifiable outputs, reducing their suitability for rigorous academic or industrial research use cases.

Consequently, there is a clear need for an intelligent framework that can autonomously plan, retrieve, validate, and synthesize information from both internal knowledge stores and external web sources. Such a framework should ensure factual consistency, maintain proper citation traceability, and present users with comprehensive and verifiable research artifacts.

The primary goal of this research is to propose DeepRAG, a Multi-Agent Retrieval-Augmented Generation framework that integrates the capabilities of LLMs, vector databases, and autonomous agents to support verifiable research synthesis. The specific objectives of this work are to:

• Design an agent-based architecture that manages task planning, retrieval, validation, and synthesis through coordinated agent interaction using Autogen;
• Specify the integration of ChromaDB as a vector database for semantic search and memory-based contextual retrieval;
• Define the use of Crawl4AI for real-time web data extraction and information enrichment;
• Introduce a Citation Manager Agent responsible for validating, merging, and maintaining traceable references across sources;
• Outline a Report Generator capable of producing structured, citation-backed research reports in PDF format using FPDF;
• And establish a feedback and persistence mechanism where validated outputs are stored and reused for future research workflows.

Through these objectives, DeepRAG seeks to bridge the gap between static RAG pipelines and dynamic, agent-based frameworks, laying the groundwork for autonomous, explainable, and verifiable AI-assisted research systems.

II. LITERATURE REVIEW

Retrieval-Augmented Generation (RAG) is an architectural paradigm designed to improve the factual grounding of Large Language Models (LLMs) by integrating external information retrieval with text generation. Rather than relying solely on parameters learned during training, RAG systems retrieve relevant documents from external knowledge sources and condition the generation process on that retrieved context [Lewis et al., 2020]. This approach has demonstrated improved factual accuracy and contextual relevance in knowledge-intensive tasks.

Several frameworks have been developed to support RAG pipelines and agent-oriented AI architectures. LangChain [2] provides modular building blocks for connecting LLMs with retrievers, memory modules, and external APIs. While it facilitates rapid development of RAG-based applications, its primary execution model focuses on single-agent workflows, limiting autonomous coordination and dynamic task delegation.

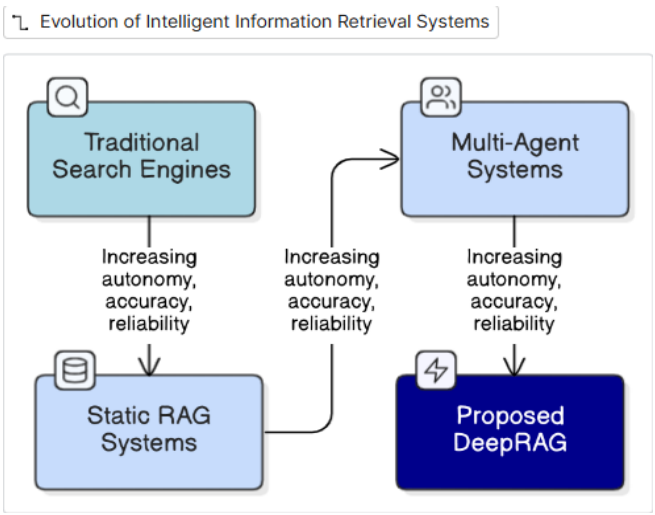

Fig.1.*Evolution of information retrieval architectures from traditional search to agentic RAG.*

Description: The figure illustrates how retrieval approaches have evolved, beginning with keyword-based search engines, moving to static Retrieval-Augmented Generation (RAG) that grounds generation on pre-indexed corpora, and progressing toward multi-agent architectures that enable decentralised reasoning. DeepRAG is presented as a design-level synthesis that integrates multi-agent orchestration, live web crawling, and an explicit Citation Manager to improve factual accuracy, citation consistency, and end-to-end report generation.

LlamaIndex [3] (formerly GPT-Index) emphasises efficient indexing and retrieval over large document collections to support LLM-based querying. Although it streamlines semantic search and vector-based retrieval, it does not natively support multi-agent orchestration or collaborative reasoning among system components.

Autogen [4], developed by Microsoft, introduces a framework for multi-agent conversational systems in which specialized agents collaborate through structured dialogue. Autogen demonstrates how agents can reason collectively and decompose complex tasks; however, it does not directly integrate retrieval-based grounding or citation-aware validation mechanisms, which are critical for research-oriented synthesis.

Together, these frameworks establish a strong foundation for intelligent information retrieval and agent-based reasoning. Nevertheless, they typically lack a unified mechanism that combines real-time web data acquisition, cross-source verification, and automated research report generation within a single coherent architecture.

Agentic Artificial Intelligence seeks to create autonomous, goal-driven systems capable of reasoning, communication, and cooperation. In multi-agent architectures, individual agents assume specialized roles—such as planning, information gathering, or synthesis—while coordinating to achieve shared objectives. Recent studies [4][10] indicate that coordinated multi-agent systems can outperform linear or single-pipeline architectures in complex reasoning and decision-making tasks. However, most practical applications of agentic AI remain focused on conversational agents or software automation rather than retrieval-augmented research synthesis.

Web-based data extraction plays a crucial role in obtaining up-to-date information. Traditional web crawlers rely primarily on HTML parsing and keyword matching, which can result in redundant or weakly relevant outputs. More recent tools, such as Crawl4AI [5], apply structured crawling and semantic filtering techniques to acquire domain-relevant, context-rich web content. Despite these advances, ensuring the credibility, consistency, and traceability of extracted web data remains a significant challenge.

Existing literature highlights the absence of automated validation layers capable of assessing source reliability, eliminating duplicate information, and integrating live web content with vector-based knowledge representations. As a result, many retrieval-based systems struggle to guarantee citation integrity and reproducibility in generated research outputs.

Based on the reviewed studies, several key research gaps can be identified:

1. Lack of Real-Time Knowledge Fusion: Most RAG systems rely on static or periodically updated datasets and do not integrate live web data streams.
2. Limited Multi-Agent Coordination: Many frameworks execute tasks sequentially using single-agent pipelines, reducing adaptability and scalability.
3. Absence of Verification Mechanisms: Existing systems rarely validate source authenticity or enforce citation consistency.
4. No Unified Output Generation: Few architectures produce structured, citation-backed research reports suitable for academic or industrial dissemination.

To address these gaps, the proposed DeepRAG framework conceptually integrates multi-agent orchestration, semantic vector retrieval, and real-time web data collection. By combining agent coordination mechanisms (Autogen), vector databases (ChromaDB), and structured web crawling (Crawl4AI), DeepRAG is designed as a unified framework for autonomous, verifiable, and context-aware research synthesis. This approach seeks to bridge the divide between static RAG pipelines and adaptive agentic intelligence.

III. PROPOSED METHODOLOGY AND SYSTEM ARCHITECTURE

The proposed DeepRAG framework is designed to automate and enhance research synthesis by integrating multi-agent orchestration, retrieval-augmented generation (RAG), and real-time web data extraction. Unlike traditional single-agent RAG pipelines, DeepRAG is a modular, implementation-ready architecture in which specialized agents perform distinct roles such as planning, retrieval, crawling, validation, and synthesis. The framework is designed for parallel execution, fault tolerance, and reproducible output generation.

DeepRAG's component ecosystem is specified to leverage established tools where appropriate: agent orchestration via Autogen, semantic vector storage with ChromaDB, structured web acquisition using Crawl4AI, PDF report formatting via FPDF, and persistent session and metadata storage with MongoDB. The user-facing interface is envisioned as a lightweight web client built with React or a simple app-level front-end using Streamlit. Each selection is chosen for reproducibility and open-source accessibility.

The architecture follows a layered design (Fig. 2) with five primary layers:

1. User Interaction Layer: web UI for query entry and report download (React or Streamlit).
2. Planning & Orchestration Layer: Planner Agent decomposes queries and schedules subtasks.
3. Retrieval & Data Acquisition Layer: Retriever Agent (vector DB) and Web Scraping Agent (real-time crawl).
4. Knowledge Validation & Synthesis Layer: Citation Manager validates, and the Answer Agent synthesises verified content.
5. Storage & Output Layer: dual-store persistence (ChromaDB for embeddings; MongoDB for sessions, logs) and PDF report generation (FPDF).

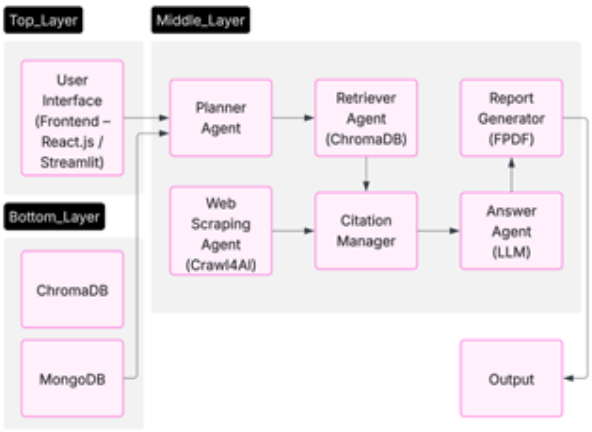

Fig.2.*System architecture of the DeepRAG framework.*

This modular decomposition supports parallel task execution, independent testing of agents, and extensibility to add domain-specific agents in future work. The end-to-end workflow is orchestrated as asynchronous subtasks so that retrieval, crawling, and validation can run concurrently when dependencies permit. Caching layers (short-term in-memory caches and ChromaDB persistence) are included to avoid redundant crawling and to reuse embeddings across related queries.

Agent Responsibilities:

A. Planner Agent: analyses the incoming research query, decomposes it into subtasks (e.g., topical retrieval, web crawl, cross-verification), and produces a task plan with dependency graph and retry policies. Design intent: idempotent subtasks, timeouts, and prioritised scheduling.

B. Retriever Agent: interfaces with ChromaDB to perform semantic search over stored embeddings. Interface (pseudo): POST /retriever/search payload { "query": "...", "top_k": N } → returns [{id, text, score, metadata}]. The agent ranks and returns context passages for downstream validation. Embeddings are computed using standard transformer encoders and stored in ChromaDB for reuse.

C. Web Scraping Agent (Crawler): invokes structured crawling using Crawl4AI to fetch fresh web passages when vector retrieval coverage is insufficient. Interface (pseudo): POST /crawler/crawl payload { "seed_urls": [...], "query": "...", "max_pages": M } → returns [{url, html, text, metadata}]. The agent applies semantic filtering and domain heuristics to reduce noise.
D. Citation Manager: merges RAG and web results, performs source credibility checks, de-duplicates content, and annotates passages with persistent reference identifiers. Interface (pseudo): POST /citation/validate payload { "documents": [...] } → returns { "validated_docs": [...], "flags": [...] }. The manager enforces citation-format templates and records provenance metadata in MongoDB.
E. Answer Agent (LLM): synthesises validated content into coherent sections constrained by citations and provenance. Design intent: supply the LLM with only validated passages and explicit citation tokens to minimise hallucination; output includes inline citation placeholders.
F. Report Generator: compiles synthesised content into a formatted PDF (via FPDF), including title page, sections, inline citation markers, and consolidated references. Interface (pseudo): POST /report/generate payload { "sections": [...], "references": [...] } → returns binary PDF.

Communication, Coordination & Fault Tolerance

DeepRAG is designed to tolerate partial failures and long-running subtasks. The Planner enforces timeouts, stores checkpoints in MongoDB, and triggers retries or alternate strategies (e.g., use more conservative retriever results when crawling fails). Agents communicate via asynchronous message passing (task queue or event bus) so slow I/O (web crawling) does not block fast-path operations (vector retrieval and synthesis). The Citation Manager keeps provenance logs for traceability and allows human-in-the-loop review for high-stakes topics.

End-to-End Workflow:

1. User submits a query via the UI
2. Planner creates subtasks and schedules Retriever + Crawler as needed.
3. Retriever queries ChromaDB for relevant embeddings; if coverage is low, Crawler fetches live web data.
4. Citation Manager validates and annotates candidate passages.
5. Answer Agent synthesizes citation-anchored content.
6. Report Generator compiles the final PDF artifact.
7. MongoDB records session metadata, task logs, and reference mappings; embeddings and validated texts persist in ChromaDB for reuse.

Data Management: Dual-Store Design

ChromaDB (Vector Store): stores embeddings and semantic indices for efficient similarity search and retrieval.
MongoDB (Session Store): records session context, task states, provenance logs, and report metadata.
The dual-store design decouples semantic retrieval from session state and lets the framework apply different retention and access policies per store.

IV. EXPECTED RESULTS AND EVALUATION PLAN

The DeepRAG framework is intended to be evaluated during implementation using a combination of quantitative metrics and qualitative assessments that capture retrieval quality, citation reliability, agent coordination efficiency, and system responsiveness. The evaluation plan below defines the metrics, measurement procedures, datasets, baselines, human-evaluation design, and statistical analysis methods to be used during prototype validation. All measures and thresholds listed are design goals or intended targets to guide the implementation and testing phases.

A. **Evaluation Metrics (definitions & computation)**

1. Retrieval accuracy and relevance measured by Precision, Recall, and F1-score computed over a set of ground-truth (query, relevant-document) judgments:

- Precision = TP / (TP + FP)
- Recall = TP / (TP + FN)
- F1 = 2 · (Precision · Recall) / (Precision + Recall)

Measurement procedure: for each query, mark retrieved documents as True Positive (TP) if relevant per gold judgments; aggregate micro- and macro-averages across the query set.

2. Citation Validity Ratio (CVR) proportion of generated citations that are verifiably accessible and correctly matched to the cited text:

- CVR = Validated_Citations / Total_Citations

Measurement procedure: for a sample of generated reports, automatically check URL accessibility and then have human annotators confirm whether the cited source supports the cited claim.

3. Agent Coordination Efficiency (ACE) quantifies multi-agent reliability:

- ACE = Successful_Subtasks / Planned_Subtasks (within time budget)

Measurement procedure: instrument the planner to log planned subtasks and mark each subtask success/failure within a predefined timeout; compute ACE per-query and average across runs.

4. Latency and Throughput per-stage and end-to-end timing measurements:

- Retrieval latency (T_retrieval): average ChromaDB query time.
- Crawling latency (T_crawl): average Crawl4AI fetch time.
- Synthesis latency (T_synth): LLM prompt/response time.
- Report generation latency (T_report): FPDF compile time.
- End-to-end latency = T_retrieval + T_crawl (if executed) + T_synth + T_report + orchestration overhead.

Design target: maintain end-to-end latency suitable for moderate research queries (target: $\leqslant$ 10 s) this is a system design objective to be validated.

5. Qualitative quality human raters score generated reports on scales (e.g., 1–5) for factual correctness, depth, clarity, and citation appropriateness. Inter-annotator agreement (Cohen's κ or Fleiss' κ) will be reported.

B. **Experimental Protocol & Datasets**

1. Benchmark queries and domains: Construct a query set of $\geqslant$100 queries spanning 4 - 6 domains (e.g., AI ethics, sustainable computing, IoT security, climate informatics) with varied difficulty (fact lookup, literature synthesis, comparative analysis). For each domain, create or sample 15–30 representative queries.

2. Gold judgments: For retrieval evaluation, prepare relevance labels per query (top 50 candidate documents) using domain experts or curated datasets (arXiv/PubMed subsets where applicable).

3. Baselines: Compare DeepRAG (design/prototype) against two representative baselines implemented with: (a) LangChain-based RAG pipeline and (b) LlamaIndex (GPT-Index) retrieval + generation. Ensure identical LLMs and embedding models across systems for fair comparison.

4. Retrieval configuration: use top-k retrieval (k = 10) as default; report sensitivity for $k \in \{5, 10, 20\}$.

5. Runs and seeds: Run each query 5 independent times to capture non-determinism; record random seeds, model versions, hardware, and runtime environment.

C. **Human Evaluation & User Study Design**

1. Annotators: Recruit 12–20 annotators (mix of graduate students and domain experts). Provide annotation guidelines and calibration examples.

2. Tasks: For a stratified sample ($\approx$30 reports per system), ask annotators to rate: factual correctness, citation relevance, coherence, and usefulness (Likert 1 – 5). Include an explicit question asking whether the report contains an unverified claim.

3. Aggregation: Compute mean scores per metric; measure inter-annotator agreement (Cohen's κ for pairs, Fleiss' κ for multiple raters). Use majority voting to resolve citation verification disagreement when needed.

4. Ethics: Obtain informed consent; anonymise any human-generated content; follow institutional human-subjects guidelines.

D. **Agent Coordination & Stress Testing**

1. ACE experiments: Create tasks with varying dependency complexity (shallow → deep task graphs) and measure ACE, task completion time, and retry rates.

2. Load testing: Simulate concurrent user sessions (e.g., 1, 5, 10, 25 concurrent queries) and measure throughput, average tail latency (P95, P99), and system resource utilization.

3. Failure modes: Intentionally inject crawler delays, unavailable sources, and LLM failures to evaluate planner retry policies and graceful degradation behavior. Log recovery strategies and human-in-the-loop triggers.

E. **Statistical Analysis & Significance Testing**

Use paired statistical tests to compare metric distributions between DeepRAG and each baseline (e.g., paired t-test if normality holds; otherwise Wilcoxon signed-rank test).

For classifier-like measures, report 95% bootstrap confidence intervals.

For human ratings, report effect sizes (Cohen's d) and test for significance with appropriate corrections for multiple comparisons (e.g., Bonferroni or Benjamini–Hochberg).

F. Result Visualisation & Artifacts for Reproducibility

Visualisations: bar charts for Precision/Recall/F1, line plots for latency breakdown, heatmaps for ACE vs. task complexity, and tables for CVR by domain.
Reproducibility artifacts: publish evaluation scripts, dataset sampling code, Dockerfile, and a sample of generated reports; include exact dependency versions and random seeds.
Supplementary materials: attach sample generated PDF reports and the pseudo-API spec in the appendix.

G. **Expected Outcomes (design-level summary)**

No.	Feature / Capability	LangChain	LlamaIndex	DeepRAG
I	Multi-agent orchestration	No	No	Yes (Designed)
II	Real-time web crawling	Limited / Custom Integration	No	Yes (Crawl4AI)
III	Citation validation support	No	No	Yes (Citation Manager)
IV	Automated PDF report generation	No	No	Yes (FPDF)
V	Dual-store persistence (Vector + Session)	Partial	Partial	Yes
VI	Agent coordination & fault tolerance	Limited	No	Yes
VII	Research-artifact oriented output	No	No	Yes

Table I. *Comparative Analysis of Retrieval-Augmented Generation Frameworks: LangChain, LlamaIndex, and DeepRAG*

DeepRAG is expected to provide design advantages over conventional RAG pipelines by:

- Increasing citation traceability through the Citation Manager (measured via CVR),
- Enabling coordinated, parallel retrieval + crawling to improve coverage, and
- Generating artifact-oriented outputs (PDF reports) that improve downstream usability.

The empirical evaluation described above will quantify these expectations and identify tradeoffs (e.g., latency vs. verification strictness).

V. FUTURE SCOPE

While the present DeepRAG design targets general-purpose research synthesis, the architecture is intentionally modular so that future iterations may incorporate specialized capabilities and deployment strategies. Below we outline prioritized extensions and practical considerations that will guide prototype development and subsequent evaluation.

A. Domain-specialised agents

The framework is designed to accommodate domain-specific agents that embed specialised knowledge, vocabularies, and source connectors. Example extensions include a Biomedical Research Agent with direct connectors to scholarly repositories (e.g., PubMed) or a Legal Agent that interfaces with legal repositories and case-law databases. Domain agents would provide tailored retrieval heuristics, credibility filters, and terminology-aware synthesis rules, improving contextual depth and precision for discipline-specific tasks.

B. **Multimodal reasoning and document understanding**

Current scope is primarily text-centric; however, integrating vision-language models and OCR pipelines will enable DeepRAG to interpret figures, charts, tables, and scanned documents. A multimodal pipeline would:

1. Extract structured data from images via OCR and VLM embeddings.
2. Align tabular/graphical content with textual evidence.
3. Include extracted objects as validated evidence items in the Citation Manager.

This extension will broaden applicability to scientific domains where key findings are presented as figures or datasets.

C. Self-optimisation and continual learning

To improve long-term performance, DeepRAG could adopt reinforcement learning (RL) or bandit-style strategies where agents receive feedback signals (e.g., citation validity, user ratings, task success) and adapt retrieval, validation, or synthesis policies. A carefully designed reward function emphasising factual correctness and citation quality, combined with safe exploration and periodic human review, would allow the system to refine agent policies over successive research cycles while controlling for drift.

D. **Cloud-native deployment and scalability**

For large-scale or multi-institution deployment, DeepRAG is designed to be cloud-deployable using containerization and orchestration. Recommended approaches include packaging agents as containers and orchestrating them with Kubernetes; persistent stores and vector services can be hosted on cloud providers (e.g., AWS, Microsoft Azure, or Google Cloud). Container images and CI/CD pipelines (Docker + Kubernetes) will support horizontal scaling, fault isolation, and rolling updates. A cloud-native setup also enables distributed agent execution for concurrent user sessions and research-as-a-service (RaaS) offerings.

E. **Application domains and impact**

Potential applications include:

1. Academic research assistance (automated literature reviews and citation checking).
2. Corporate knowledge management (verified insight extraction from live web sources).
3. Policy and legal analysis (evidence-based synthesis from public records).
4. Education technology (context-aware tutoring and research aids).

For each domain, domain agents and tailored credibility heuristics will be essential to achieve practical utility.

F. **Ethical, legal, and reproducibility considerations**

Future work must explicitly address ethics and reproducibility: respect robots.txt and copyright when crawling; flag pay-walled or restricted sources; and maintain provenance logs for all synthesized claims. Implementations should publish evaluation scripts, random seeds, dataset samples, and Docker images to permit reproducibility. For human subjects work (e.g., user studies), obtain informed consent and follow institutional review guidelines. Finally, introduce human-in-the-loop checkpoints for high-stakes domains (medical, legal) to prevent unverified automated dissemination.

G. **Roadmap (practical next steps)**

1. Prototype domain agent for one discipline (e.g., biomedical) and evaluate CVR and retrieval F1 on a domain query set.
2. Integrate a basic OCR + VLM pipeline and measure multimodal coverage improvement on sample figure-heavy papers.
3. Implement planner retry policies and containerised agent images, and perform load testing on a cloud staging environment.
4. Publish code, evaluation scripts, and sample generated reports as supplementary materials to support community adoption.

These future directions position DeepRAG to evolve from a conceptual framework into a robust, extensible research ecosystem that balances autonomous reasoning with verifiable evidence and human oversight.

VI. ETHICAL CONSIDERATIONS AND DATA USE

The DeepRAG framework is designed with explicit consideration for ethical data use, transparency, and responsible AI deployment. As the system integrates both internal knowledge stores and real-time web sources, strict safeguards are required to ensure lawful, ethical, and reproducible operation.

Web data acquisition components are intended to respect website access policies, including compliance with robots.txt directives and applicable copyright restrictions. The framework does not aim to bypass paywalls or restricted-access content; instead, all retrieved sources are explicitly tracked, and their availability and accessibility are verified before inclusion in synthesized outputs.

To mitigate the risk of misinformation, DeepRAG incorporates a dedicated Citation Manager Agent that enforces source traceability and rejects unverifiable or duplicate references. Generated outputs are explicitly grounded in validated evidence, and uncertainty or source limitations can be flagged when confidence thresholds are not met.

For high-stakes domains such as medical, legal, or policy analysis, DeepRAG is intended to operate as a decision-support tool rather than a decision-making authority. Human oversight and expert review remain essential, and future deployments should include configurable human-in-the-loop checkpoints before disseminating critical information.

User interaction data, session logs, and stored research artefacts are intended to be handled in accordance with data protection best practices. Personally identifiable information, if collected during user studies or system evaluation, should be anonymized, securely stored, and used solely for research and system improvement purposes with informed consent.

Finally, reproducibility and transparency are treated as ethical imperatives. Future implementations of DeepRAG are expected to publish evaluation protocols, dataset sampling procedures, system configurations, and random seeds where applicable, enabling independent verification and responsible reuse of the framework.

VII. CONCLUSION

This paper proposed DeepRAG, a conceptual, implementation-ready multi-agent Retrieval-Augmented Generation framework designed to automate, verify, and enhance literature-based research synthesis. DeepRAG specifies a modular agent decomposition Planner, Retriever, Web Scraper, Citation Manager, Answer Agent, and Report Generator orchestrated to combine semantic vector retrieval (ChromaDB) with real-time web acquisition (Crawl4AI) and produce citation-anchored research artefacts (PDFs via FPDF). By prioritizing citation traceability, provenance logging, and dual-store persistence (ChromaDB + MongoDB), the framework targets improved transparency and reproducibility for AI-assisted research.

The principal contributions of this work are:

- A formal agentic architecture for coordinated retrieval and synthesis.
- A citation-first design that embeds verification into the synthesis loop.
- An implementation-ready evaluation plan with well-defined metrics (Precision/Recall/F1, Citation Validity Ratio, Agent Coordination Efficiency, and latency breakdown) to quantify trade-offs between accuracy and responsiveness. The proposed design explicitly treats generated reports as first-class research artefacts and provides pseudo-APIs and reproducibility artefacts to accelerate prototype development.

To transition from design to practice, we outline a three-phase development roadmap with deliverables and evaluation checkpoints:

Phase 1: System Setup & Core Agents: implement Planner, Retriever, and Crawler agents; integrate ChromaDB for embeddings and a basic UI for query submission. Deliverables: containerized agent images, ChromaDB schema, simple example queries. Checkpoint: retrieval F1 $\geqslant$ baseline on domain test set.

Phase 2: Validation & Synthesis: implement the Citation Manager and Answer Agent; add report generation (FPDF) and provenance logging to MongoDB. Deliverables: validated citation pipeline, example PDF reports, annotation guidelines for human evaluation. Checkpoint: Citation Validity Ratio (CVR) target and inter-annotator agreement thresholds met.

Phase 3: Testing, Scalability & Release: end-to-end evaluation versus baselines (LangChain, LlamaIndex); load testing and cloud-native deployment artefacts (Docker/Kubernetes). Deliverables: evaluation scripts, dataset samples, reproducibility Dockerfile, and public supplementary materials. Checkpoint: statistically significant improvements on designed metrics or documented trade-offs with clear mitigation strategies.

DeepRAG is presented as a reproducible architectural blueprint rather than a finalised system. Future extensions (domain-specialised agents, multimodal reasoning, and self-optimisation) are intentionally supported by the modular design. The framework also embeds ethical and data-use safeguards, provenance tracking, robots.txt adherence, paywall handling, and human-in-the-loop checkpoints for high-stakes domains to balance automation with responsible deployment.

In summary, DeepRAG outlines a practical pathway toward autonomous, verifiable research assistants that emphasize factual correctness, citation traceability, and continual improvement. The manuscript provides the community with a concrete starting point for prototyping, comparative evaluation, and responsible deployment in academic and industrial environments.

REFERENCES

[1]. P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 9459–9474, 2020.

[2]. R. L. Logan, N. S. Thomas, and C. J. Reed, "LangChain: Building Context-Aware AI Applications Using Large Language Models," *LangChain Documentation*, 2023. [Online]. Available: https://www.langchain.com

[3]. J. Zhang, L. Wang, and M. Xu, "LlamaIndex: Indexing and Retrieval for Large Language Models," *LlamaIndex Technical Report*, 2023. [Online]. Available: https://docs.llamaindex.ai

[4]. Microsoft Research, "Autogen: Multi-Agent Conversational Framework for LLM Applications," *Microsoft Research Technical Report*, 2024. [Online]. Available: https://microsoft.github.io/autogen

[5]. Crawl4AI Development Team, "Crawl4AI: Web Crawling Framework for AI Agents," GitHub Repository, 2024. [Online]. Available: https://github.com/unclecode/crawl4ai

[6]. ChromaDB, "Chroma: Open-Source Embedding Database for AI Applications," *Chroma Documentation*, 2024. [Online]. Available: https://www.trychroma.com

[7]. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, pp. 5998–6008, 2017.

[8]. OpenAI, "GPT-4 Technical Report," *arXiv preprint arXiv:2303.08774*, 2023.

[9]. M. Zhao, G. Zhao, and M. Qu, "College Smart Classroom Attendance Management System Based on Internet of Things," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 4953721, Jul. 2022.

[10]. Y. Wang, X. Zhao, and H. Tang, "Multi-Agent Collaboration for Enhanced Reasoning in Large Language Model Systems," arXiv preprint arXiv:2309.01234, 2023.