IntrAgent: An LLM Agent for Content-Grounded Information Retrieval through Literature Review

Fengbo Ma[1,*], Zixin Rao[1,*], Xiaoting Li[1], Zhetao Chen[1], Hongyue Sun[1], Yiping Zhao[1], Xianyan Chen[1,†], Zhen Xiang[1,†]

[1]University of Georgia, Athens, GA, USA

[*]Equal contribution; [†]Corresponding authors: xychen@uga.edu, zxiangaa@uga.edu

Abstract

Scientific research relies on accurate information retrieval from literature to support analytical decisions. In this work, we introduce a new task, *INformation reTRieval through literAture reVIEW* (IntraView), which aims to automate fine-grained information retrieval *faithfully* grounded in the provided content in response to research-driven queries, and propose IntrAgent, an LLM-based agent that addresses this challenging task. In particular, IntrAgent is designed to mimic human behaviors when reading literature for information retrieval – identifying relevant sections and then iteratively extracting key details to refine the retrieved information. It follows a two-stage pipeline: a *Section Ranking* stage that prioritizes relevant literature sections through structural-knowledge-enabled reasoning, and an *Iterative Reading* stage that continuously extracts details and synthesizes them into concise, contextually grounded answers. To support rigorous evaluation, we introduce IntraBench, a new benchmark consisting of 315 test instances built from expert-authored questions paired with literature spanning *five* STEM domains. Across seven backbone LLMs, IntrAgent achieves on average 13.2% higher cross-domain accuracy than state-of-the-art RAG and research-agent baselines.

1 Introduction

In modern scientific research, accurately extracting metadata, experimental setups, and contextual knowledge from prior literature is essential. However, such information retrieval is challenging due to the complexity of scientific literature (Huang et al., 2025), which often requires deep domain expertise and substantial time to read and interpret (Wu et al., 2025; Lu et al., 2024a; Li et al., 2024; Wang et al., 2024c). Therefore, an automated system for researchers to accurately and efficiently extract relevant information from literature can potentially transform research practices across many scientific disciplines.

Motivated by this need, we propose a new task called INformation reTRieval through literAture reVIEW (IntraView) – given a literature and an information retrieval query, extract and synthesize information **faithfully grounded in the provided content**. While IntraView follows the general format of Content Question Answering (CQA) (Jin et al., 2019; Mihaylov et al., 2018; Welbl et al., 2017; Chen et al., 2021; Reddy et al., 2019; Chen et al., 2017), it introduces greater challenges due to the structural complexity and domain-specific language of scientific literature, in contrast to the more generic knowledge chunks typically used for standard CQA (Dasigi et al., 2021; Lu et al., 2022; Joshi et al., 2017; He et al., 2024; Rajpurkar et al., 2018; Rein et al., 2023; Jin et al., 2021; Krithara et al., 2023). Moreover, in practical applications of IntraView, it is crucial to avoid hallucination by explicitly acknowledging when the requested information is not present in the literature.

Due to these critical differences, existing methods for conventional CQA tasks are insufficient for addressing the new challenges of IntraView (Wu et al., 2024b; Lu et al., 2024c; Wang et al., 2023a). *Direct LLM querying* for specific information retrieval – even with models fine-tuned for scientific domains (Zhang et al., 2024c) – struggles to manage the overwhelming information contained in a full literature. *Retrieval-augmented generation* (RAG) approaches attempt to address this limitation by selecting a small set of text chunks deemed most relevant to the query (Lewis et al., 2020; Wang et al., 2024d; Ye et al., 2024), typically based on semantic similarity, and then using these chunks to prompt the LLM. However, such methods rely solely on surface-level semantic similarity,

IntrAgent Project page: https://intragent.github.io/. IntrAgent Code: https://github.com/FengboMa/IntrAgent. IntraBench Dataset: https://huggingface.co/datasets/IntrAgent/IntraBench.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 674–715
July 2-7, 2026

often failing to align domain-specific terminology in the query with the actual relevant content, and ignore the rich structural organization inherent to scientific documents.

Recently, numerous LLM-based agents have been developed to tackle complex tasks (M. Bran et al., 2024; Yin et al., 2024; Wu et al., 2024a; Tang et al., 2023; Nguyen et al., 2025), including those involving scientific literature, such as literature search, summarization, research idea exploration, and academic writing assistance. Compared to standalone LLMs, these agents leverage the reasoning capabilities of LLMs for comprehensive task planning and interact with external tools and environments to enhance task execution (Whitfield and Hofmann, 2023). However, the scientific QA task addressed by these agents *fundamentally differs* from our `IntraView`. They aim to answer general scientific questions through external resource exploration, whereas `IntraView` focuses on retrieving and reasoning strictly constrained to a provided paper. Thus, these methods perform poorly on `IntraView`, as will be demonstrated in our experiments.

To close this gap, we propose an LLM agent, `IntrAgent`, as the first specialized solution to `IntraView`. The key idea behind `IntrAgent` is to emulate human behavior when reading technical literature for information retrieval – identifying the most relevant sections and progressively accumulating details until the query is fully addressed (Suppawittaya, 2021; Miller, 1956). Accordingly, `IntrAgent` comprises two main stages. First, in the *section ranking* stage, paper sections are reordered based on structure-aware reasoning by the LLM to reflect their relevance to the query. Second, in the *iterative reading* stage, the reordered sections are sequentially accessed to extract relevant details, continuing until the accumulated information is deemed, via LLM reasoning, sufficient to answer the query. Notably, `IntrAgent` incorporates several novel designs to address the unique characteristics and challenges of `IntraView`. For example, we introduce a hierarchy preservation step that captures the structural organization of the research literature, enabling more comprehensive reasoning during section ranking. Additionally, during iterative reading, we design a sufficiency check mechanism to assess whether the extracted details are adequate for answering the query, ensuring the faithfulness by reducing the risk of hallucination.

In addition, we introduce `IntraBench`, the first benchmark designed to evaluate `IntraView` approaches, including our proposed `IntrAgent`. The benchmark consists of 315 test instances drawn from five scientifically and societally significant domains—physics, earth science, public health, engineering, and material science. We further assess `IntrAgent`against state-of-the-art RAG systems and literature-oriented agents, demonstrating its superior performance across all domains. Our key contributions are summarized below:

- We introduce `IntraView`, a novel task for accurate, automated, and *content-grounded* information retrieval from a provided scientific literature, and propose `IntrAgent`, an LLM agent specifically designed to tackle this task.
- We develop a novel two-stage pipeline for `IntrAgent`, consisting of section ranking and iterative reading, designed to mimic human reading behavior. We introduce a hierarchy preservation mechanism to help the agent leverage structural knowledge for more effective section ranking, and implement a sufficiency check to mitigate hallucination during iterative reading.
- We propose `IntraBench`, the first benchmark for evaluating `IntraView`, consisting of 315 test instances across five impactful domains.
- We evaluate `IntrAgent` on `IntraBench` and show that it outperforms both state-of-the-art RAG and literature-agent baselines across representative backbone LLMs in average cross-domain accuracy.

2 Related Work

Retrieval-Augmented Generation (RAG) RAG enhances LLMs by retrieving relevant documents for response generation (Gao et al., 2024), enabling external knowledge integration in knowledge-intensive tasks (Wu et al., 2025). The vanilla RAG framework adopts a three-stage pipeline consisting of indexing, retrieval, and generation (Lewis et al., 2020), but it faces challenges such as retrieval noise, hallucinated outputs, and limited reasoning over retrieved content (Zhu et al., 2024). Recent RAG variants improve embedding quality, retrieval accuracy, and context control through techniques such as re-ranking (Ye et al., 2024), dynamic embeddings (Jiang et al., 2024), contextual retrieval (Anthropic, 2024), external memory (Li et al., 2024; Wang et al., 2024c), and modular pipelines (Lu et al., 2024a). However, most of these approaches

still follow a flat retrieve-then-generate architecture, which limits their effectiveness for fine-grained information retrieval from structured scientific content, as required by IntraView.

LLM Agent for Literature Tasks Many domain-specific research agents have been developed across scientific disciplines, including chemistry (M. Bran et al., 2024; Tang et al., 2025), engineering (Zhang et al., 2024b; Singh et al., 2025), mathematics (Wu et al., 2024b), and bioinformatics (Xin et al., 2024); but they are tailored to tasks within a single field. There also exist agents for cross-domain literature-related tasks other than IntraView, such as literature search (Agarwal et al., 2024), research ideation (Lu et al., 2024b), and end-to-end scientific writing (Schmidgall et al., 2025; Shao et al., 2024). However, most of these agents are not directly applicable to IntraView, which presents distinct goals and requirements. While some agents for general scientific QA via literature search, such as PaperQA (Lála et al., 2023), PaperQA2 (Skarlinski et al., 2024), QASA (Lee et al., 2023), and SciMaster (Chai et al., 2025), can be adapted to IntraView by disabling their retrieval component and supplying the same paper as input, these systems are not designed for IntraView and therefore cannot match the performance of our IntrAgent, as will be demonstrated in our extensive evaluation.

3 Proposed IntraView Task

Information retrieval from scientific literature is critical to a wide range of subjects. The retrieved information could be used to guide the experimental design, hypothesis refinement, simulation configurations, statistical methodology, results validation, and other downstream decision-making in research workflows (Rothstein et al., 2024; Kumar et al., 2024; Yadav et al., 2022; Senapati et al., 2024).

In this work, we propose a novel task, INformation reTRieval through literAture reVIEW (IntraView), which aims to accurately extract key information from a provided research paper in response to a specific query. For example, given a Surface-enhanced Raman spectroscopy (SERS) paper with query *"What is the excitation laser wavelength used for SERS measurements?"* one should respond with a specific laser wavelength *solely* based on the experimental context described in the paper (Zhao et al., 2024). Formally, we construct IntraView as a CQA problem (Jin et al., 2019). The objective is to build an automated system that, when given a literature C and a research-driven question Q, generates an accurate and *content-grounded* answer A by identifying the most relevant information in C without hallucination.

Compared to **existing CQA tasks**, IntraView differs in two aspects: (a) it provides the full literature rather than a pre-selected or processed chunk, with the relevant information potentially appearing anywhere in the literature or *not at all*; and (b) it tackles domain-specific queries that may require cross-referencing multiple sections beyond where the final answer resides. Compared with other literature-related tasks, such as **scientific QA via literature search**, IntraView emphasizes **faithful** information retrieval restricted to the provided content, independent of the external validity of its scientific claims. In contrast to other existing tasks such as PeerQA(Baumgärtner et al., 2025), where answers may involve author intent, argumentative reasoning, or external domain knowledge beyond the paper, IntraView strictly requires that all answers be directly grounded in the provided content, necessitating structured, multi-stage reading rather than generative reconstruction. Therefore, existing approaches for those tasks cannot effectively tackle IntraView, as will be shown by our extensive evaluation.

4 Proposed IntrAgent Framework

4.1 Overview of IntrAgent

As the first agent framework specially designed for IntraView, IntrAgent employs "*mindset bionics*" approach to emulate the natural reading workflow of humans during information retrieval – it begins with a guiding question, infers the section most likely to contain the answer, extracts the key information, evaluates whether the question has been sufficiently addressed, and iterates through additional sections as needed (Miller, 1956; Suppawittaya, 2021). Accordingly, IntrAgent comprises two major stages – *section ranking* (Section 4.2) and *iterative reading* (Section 4.3) – as illustrated in Figure 1. In the section ranking stage, the agent identifies the most relevant sections of a literature using an LLM by structure-aware reasoning. In the iterative reading stage, it repeatedly gathers information and evaluates sufficiency until the input query can be adequately answered.

Our design of IntrAgent offers several advantages over approaches for other CQA tasks, such as RAG (Lewis et al., 2020), making it well-suited for

IntraView: 1) Effective context prioritizing: Unlike RAG's semantic-similarity-based chunk ranking, our reasoning-based section ranking more precisely locates the details relevant to the query within the literature. 2) Explicit hallucination mitigation: The sufficiency check in the iterative reading stage determines whether additional reading is necessary, thereby explicitly reducing hallucination by ensuring that only substantiated answers are returned. A toy example of IntrAgent handling a SERS-related query is shown in Figure 2.

4.2 Section Ranking

Scientific research literature typically follow a well-defined *section hierarchy*: parent-level headings convey broader topics, while sub-level headings provide more specific details. Unlike flat, semantic-similarity-based RAG that overlooks document structure and often fails to align a scientific question with the relevant sections (Lewis et al., 2020), IntrAgent leverages this structural knowledge for reasoning-based ranking, prioritizing sections relevant to the question through three key steps:

Section Heading Parsing To standardize structural information across literature with diverse templates, we convert each input literature C into a Markdown-formatted version C' with systematic markers for section headings. For papers provided as PDFs, which is the most common case, we use minerU (Wang et al., 2024a), a visual model for layout and section detection, for the conversion.

Hierarchy Preservation This step aims to construct a *section tree* that represents the hierarchical structure of the literature. This tree will support: 1) LLM-based reasoning for section ranking, and 2) structured text parsing for iterative reading. We begin by extracting all section and subsection headings from the Markdown-parsed text C' to form an initial heading set $\mathcal{H}_0$[1]. Next, we prompt an LLM to infer the hierarchical relationships among these headings, treating each parent section as a node with child nodes corresponding to its subsections. The prompt is detailed in Appendix E.1.1. The LLM returns a set of paths from the root to each node in the hierarchy. To eliminate redundant nodes, we remove paths where a parent section is immediately followed by a subsection without intervening content. The resulting filtered set of headings is denoted as $\mathcal{H} = \{h_1, h_2, \ldots, h_n\}$.

[1]Markdown files from minerU uniformly prepend a single pound sign to all section and subsection titles.

Reasoning-Based Ranking Given the filtered heading set $\mathcal{H}$ and the research question Q, the model is prompted to reason about which section is most likely to contain the information needed to answer Q – this design mirrors natural human reading behavior (see Appendix E.1.2 for the detailed prompt). The output of this step is a ranked list of indices $R = [r_1, \ldots, r_n]$. Using R, we reorder the sections to construct a list of (heading, text) pairs $C_R = [(h_{r_1}, t_{r_1}), \ldots, (h_{r_n}, t_{r_n})]$, where t_{r_i} denotes the text in the literature corresponding to section h_{r_i}. These reordered sections will be sequentially accessed during the iterative reading stage described next.

4.3 Iterative Reading

At each step, the agent selects its next action from a predefined set: *reordered section access*, *section detail extraction*, and *information sufficiency check*, based on reasoning over the outcomes of the previous step. This action space is carefully designed to enable targeted retrieval of key information relevant to the input query while suppressing hallucination.

Reordered Section Access When reading is initiated or a section (h_{r_i}, t_{r_i}) has been fully processed, the agent retrieves the next section $(h_{r_{i+1}}, t_{r_{i+1}})$ from the list C_R. This ensures that sections are read in descending order of estimated relevance to the research question Q, maintaining a focused and efficient reading trajectory.

Section Detail Extraction The agent examines the current section (h_{r_i}, t_{r_i}) to extract information relevant to Q. This action is designed to identify and record key scientific details D_i, including terminology, numerical data, experimental results, measurements, statistical indicators, conclusions, and any comparative or causal statements that explicitly address the research question. Each detail in D_i is anchored to its original sentence and stored in a short-term memory for final answer synthesis. See Appendix E.2.3 for detailed prompts.

Information Sufficiency Check This step governs the action loop of iterative reading by determining whether more sections need to be accessed. The agent uses an LLM to reason over the details D_i gathered in the current iteration and assess whether they are sufficient to answer the research question Q. The reading loop continues with the next section if the LLM outputs NO; otherwise, it terminates if YES is returned. Notably, our prompt here includes explicit instructions to avoid speculation and hallucination, as detailed

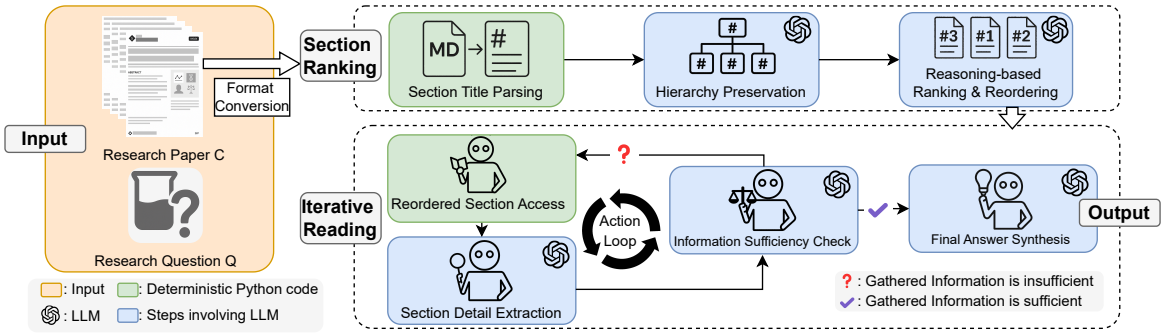

Figure 1: Overview of the IntrAgent pipeline containing two stages: *Section Ranking* (top) reorders the paper's sections by relevance to the Research Question Q, while *Iterative Reading* (bottom) steps through ranked sections, extracting information until gathered information is sufficient.

in Appendix E.2.4. The relevant evidence for a given query may be distributed across multiple non-adjacent sections of a paper. The sufficiency check explicitly prevents premature termination by requiring the agent to continue reading until sufficient supporting information has been accumulated across sections, enabling reliable *cross-section* synthesis. Moreover, we introduce a confidence-based mechanism that supports three reading styles: *conservative*, *balanced* (default), and *aggressive*, allowing the user to control the operational overhead – conservative reading accesses fewer sections, while aggressive reading explores more. Relevant prompts are detailed in Appendix E.4.

Once the sufficiency check terminates the action loop after reading m sections, the agent invokes an LLM to synthesize the final answer $A = \text{LLM}(D_1, \ldots, D_m, Q)$ using the accumulated details $\{D_1, \ldots, D_m\}$ during the iterative reading. Detailed prompts are provided in Appendix E.2.5.

5 Proposed IntraBench for Evaluation

5.1 Overview of IntraBench

As a novel task, IntraView lacks dedicated evaluation benchmarks. Thus, we propose IntraBench, a benchmark specially designed for IntraView, which demands expert-level understanding of domain-specific contexts. IntraBench consists of 315 test instances, derived from expert-curated insights and grounded theory (Hallberg, 2010) – paired with research literature from five highly impactful domains spanning physics, earth science, public health, engineering, and material science. Although focusing on different tasks, IntraBench contains more test instances from broader scientific fields compared with existing benchmarks for QA via literature search (see Table 1). More details on benchmark construction, including the domain choices, are deferred to Appendix A.

Benchmark	# Instances	Domains
LitQA	50	Bio
LitQA2	248	Bio
IntraBench	**315**	Phys, ES, PH, Engr, MS

Table 1: Comparing IntraBench with existing benchmarks for scientific QA via literature search (Phys: Physics, ES: Earth Science, PH: Public Health, Engr: Engineering, MS: Material Science, Bio: Biology).

5.2 LLM-Grounded Multiple-Choice Evaluation for IntraBench

In real-world applications, IntraView is expected to produce concise, *free-form* answers rather than selecting from predefined choices. However, this presents two major challenges for evaluation. 1) Scientific terminologies often involve abbreviations and synonyms – for example, terms like AgNP, silver nanoparticle, and silver nanorod may all refer to the same surface-enhanced substrate treatment in a SERS experiment. 2) When retrieved information is numerical or factual, precise accuracy is critical. Traditional string evaluation methods, such as BLEU (Papineni et al., 2002) or ROUGE (Lin, 2004), are unsuitable since they rely on surface-level text similarity and often fail to capture semantic correctness in specialized scientific contexts.

To address these challenges, we adopt a multiple-choice format when creating questions for IntraBench, though the evaluated methods must still generate short answers without being shown the multiple-choice options. During evaluation, an LLM maps each generated short answer to the most relevant multiple-choice candidate. This

relevance-based mapping effectively addresses the first challenge of varied terminologies such as abbreviations, synonyms, or domain-specific expressions referring to the same concept. If the LLM cannot confidently align the generated answer with any provided choice (due to insufficient relevance), we use a fallback label – "None of the above" – to avoid misclassification of ambiguous responses.

The final evaluation metric is accuracy based on the multiple-choice mapping. Empirically, this method demonstrates strong alignment (Average correctness agreement of 63/65 on the Physics dataset using GPT-4.1) with manual mappings by domain experts; see Section 6.4 for details.

5.3 Construction of IntraBench Dataset

Selection of Papers For each domain, papers are manually selected by a corresponding expert on our team. To reduce selection bias, each expert first curates a larger pool of familiar papers, from which five are randomly chosen. The selected papers are constrained to impactful, peer-reviewed journals to ensure their authority. Experts' familiarity with these papers ensures accurate annotation of the ground-truth answers for the associated questions.

Creation of Questions We aim to capture both technical depth and conceptual complexity through expert-level inquiry. To this end, the questions are generated by our domain experts based on their natural reading practices, i.e., what information they would seek from a paper within their field. These questions are categorized into four task-oriented categories based on general research principles (Dillon, 1984): study subject & experimental setup, data characteristics & collection, technical approach & details, conclusions & results. Details about question setup in Appendix A.3

Creation of Answer Choices For each question and its paired paper, six answer options are created by domain experts, including one correct answer and five distractors. The six options consistently include "All of the above" and "None of the above", either of which may serve as the correct answer when applicable. Distractors are manually constructed based on: 1) concepts, numerical values, or textual information from the paper that closely resemble the correct answer; or 2) commonly used information or conventions in the respective field.

6 Experiments

Our experiments aim to address the following research questions: (1) **RQ1** (Section 6.2): Can IntrAgent effectively solve the IntraView task compared with the baselines? (2) **RQ2** (Section 6.3): Do our designed components for IntrAgent, such as hierarchy preservation, the confidence level mechanism, and the information sufficiency check, function as intended? (3) **RQ3** (Section 6.4): Is IntrAgent robust to variations in evaluation and input conditions, such as different mapping models and non-standard headings?

6.1 Experiment Settings

Dataset and Evaluation Metrics We use our proposed IntraBench for evaluation. Following Section 5.2, we use GPT-4.1 to map each short answer generated by the method to one of the predefined answer choices. Evaluation results using alternative mapping models are presented in Section 6.3. Our primary evaluation metric is accuracy after the answer mapping.

Baseline We evaluate a broad range of retrieval-augmented generation (RAG)–based approaches widely adopted for CQA and scientific reasoning, including (1) **vanilla RAG** using embedding models *all-MiniLM-L6-v2* (Wang et al., 2020), *E5-mistral-7b-instruct* (Wang et al., 2023b), and *GritLM-7B* (Muennighoff et al., 2024), respectively, with cosine-similarity retrieval over 500-token chunks (50-token overlap); (2) **contextual RAG** variants that enhance retrieval via dynamic chunk selection and adaptive context expansion (Anthropic, 2024); and (3) **advanced RAG extensions** such as DRAGIN (Su et al., 2024b), R²AG (Ye et al., 2024), and LongRAG (Jiang et al., 2024), which introduce multi-hop reasoning, re-ranking, and long-context retrieval. We also consider representative literature-focused agents, including LUMOS (Yin et al., 2024), PaperQA2 (Skarlinski et al., 2024), Agentic-Hybrid-RAG (Nagori et al., 2025), and SciMaster (Chai et al., 2025), the last of which reports a score of 32.1 on Humanity's Last Exam, marking the state of the art of scientific agents. All baselines follow the same IntraBench evaluation protocol described above. Default hyperparameters are used unless otherwise specified; detailed configurations are provided in Appendix F.

	Method	GPT-4o	GPT-4.1	DS-R1	o3	o4-mini	Gemini-2.5 Pro	Llama-3.1-70B
RAG	Vanilla RAG all-MiniLM-L6-v2	60.3	61.2	64.3	60.4	61.5	61.8	59.2
	Vanilla RAG E5-mistral-7b-instruct	59.4	64.2	63.8	60.3	61.4	59.9	60.5
	Vanilla RAG GritLM-7B	60.4	63.2	63.2	59.7	58.4	58.4	61.4
	Context. RAG E5-mistral-7b-instruct	60.7	63.8	62.8	59.1	58.3	58.9	58.9
	Context. RAG GritLM-7B	60.8	62.8	61.6	58.4	60.7	61.6	59.2
	DRAGIN	42.5	44.6	46.9	44.0	46.9	45.9	45.4
	R^2AG	59.4	59.5	61.5	56.6	55.3	55.6	56.1
	LongRAG	62.1	64.7	65.5	57.0	58.3	57.1	57.4
Agent	LUMOS	50.2	52.1	55.4	55.2	56.4	54.9	54.4
	PaperQA2	47.7	48.9	54.0	51.8	49.2	51.2	53.8
	Agentic-Hybrid-RAG	59.8	60.2	62.3	57.5	57.8	57.2	56.6
	SciMaster	59.0	57.6	63.3	57.2	58.1	57.2	57.0
	IntrAgent (Ours)	**70.0**	**75.8**	**74.4**	**73.4**	**73.8**	**75.9**	**68.8**

Table 2: Cross-domain accuracy (in %, defined by the macro average over the five domains) on IntraBench. Our IntrAgent uniformly outperforms the RAG-based retrieval and agent-based baselines across the five domains for seven model choices. See Appendix B.1 for the complete breakdown results.

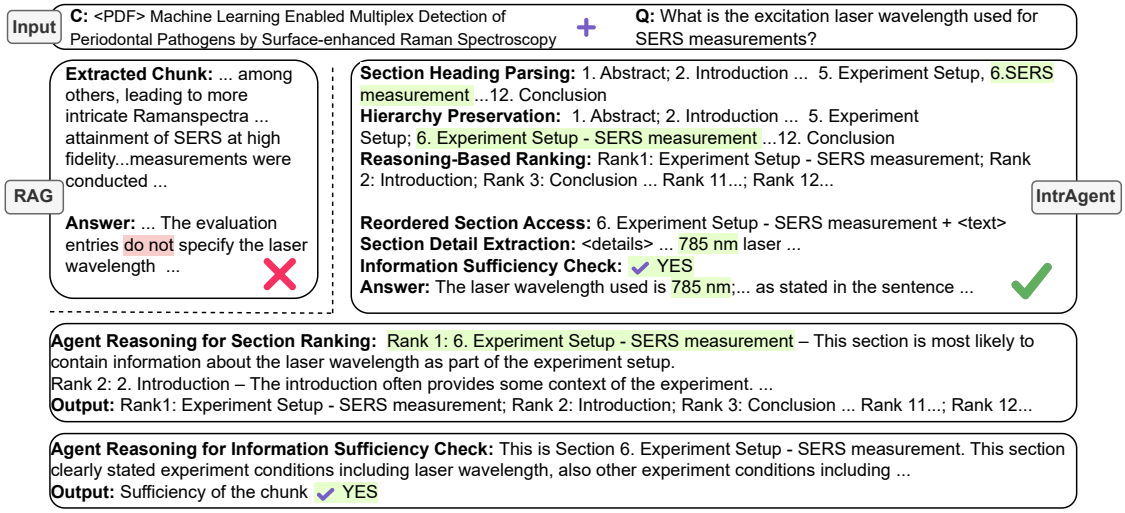

Figure 2: An example of IntrAgent executing a question-paper pair from IntraBench. For an input question Q regarding paper C, vanilla RAG fails to extract the correct chunk, resulting in an incorrect answer. In contrast, IntrAgent ranks the sections through reasoning and retrieves the correct details that pass the sufficiency check in the first iteration, leading to a correct answer. Details for section ranking and sufficiency check are also presented.

6.2 Main Result

Table 2 show that IntrAgent sets a new state of the art across all five domains of IntraBench for seven backbone LLMs. Averaged over physics, public health, earth science, engineering, and material science, IntrAgent achieves 70.0% with GPT-4o, 75.8% with GPT-4.1, 74.4% with DeepSeek-R1, 73.4% with o3, 73.8% with o4-mini, 75.9% with Gemini-2.5 Pro, and 68.8% with Llama-3.1-70B. Compared with the strongest baseline under *each* model, IntrAgent surpasses them by 7.9%, 11.6%, 8.9%, 18.2%, 17.4%, 21.0%, and 7.4%, respectively. These results suggest that performance gains stem not from increased context length, but from targeted evidence selection: unlike chunk-based methods that introduce irrelevant context, IntrAgent isolates task-relevant sections, enabling more precise reasoning. Such a performance gain stems from our novel agent design: 1) hierarchy-aware section ranking based on reasoning, and 2) sufficiency check that halts reading once evidence is complete.

In contrast, RAG sequentially feeds multiple unstructured text chunks into the LLM, which often introduces irrelevant or noisy information. The agent baselines, originally designed for scientific QA through online search, degenerate into static information retrieval pipelines similar to RAG when they are directly provided with the literature for IntraView. A representative example demonstrating the advantage of IntrAgent over the RAG baseline is shown in Figure 2, along with the reasoning trajectory for IntrAgent's section ranking and sufficiency check. In this example, the agent

Model	GPT-4o	GPT-4.1	DeepSeek-R1
w/o HP	60.7	70.3	64.2
w/ HP	**65.6**	**72.5**	**67.6**

Table 3: Accuracy (%) of IntrAgent w/ and w/o the hierarchy preservation (HP) step for three model choices.

Metric	Conservative	Balanced	Aggressive
Accuracy (%)	58.9	68.3	62.7
Avg. Iterations	9.9	5.1	3.9
Med. Iterations	11	2	1
Std Dev. Iterations	7.9	5.5	5.3
Med. Token Count	7853	6376	2233

Table 4: Impact of confidence level on the MCQ accuracy, the number of reading iterations, and token count.

Mapping	**IntrAgent**	**RAG**	**Avg.**
GPT-4o	59/65	60/65	59.5/65
GPT-4.1	65/65	61/65	**63/65**
DeepSeek-R1	61/65	61/65	61/65

Table 5: Correctness agreement between human annotations and LLM-based mapping on the physics dataset with 65 instances.

prioritizes the Experiment Setup – SERS measurement section based on the reasoning that laser wavelength information is typically part of the experimental setup. During the information sufficiency check, this section is verified to explicitly state the excitation wavelength (785 nm), confirming that the relevant detail is correctly retrieved from the appropriate section. These results address **RQ1**, confirming that IntrAgent more effectively solves the IntraView task compared to the baselines.

6.3 Ablation Studies

Study on Hierarchy Preservation We evaluate IntrAgent without the hierarchy preservation step by directly ranking the raw section headings $\mathcal{H}_0$ extracted from the Markdown-parsed text. As shown in Table 3, removing the hierarchy preservation step leads to a clear drop in cross-domain accuracy for all three model choices, highlighting the necessity of this key step for effectively incorporating structural context into section ranking.

Study on Confidence Level IntrAgent supports three confidence levels via prompt variation during iterative reading: 1) Conservative, which halts only when all answer components are explicitly observed; 2) Balanced (the default), which stops when evidence appears sufficient; and 3) Aggressive, which allows early termination based on partial evidence. The full prompts are provided in Appendix E.4. This confidence level parameter enables users of IntrAgent to control the operational overhead (more details on overhead are shown in Appendix B.3). As shown in Table 4, more aggressive reading reduces the number of iterative reading steps but at the cost of lower information retrieval accuracy. Interestingly, the conservative mode, despite reading more sections, performs the worst, consistent with RAG observations that performance degrades with very long context windows (Jiang et al., 2024; Yepes et al., 2024; Wang et al., 2024b).

Study on Information Sufficiency Check We evaluate IntrAgent without the information sufficiency check by accessing only the top-1 section during iterative reading. On the physics dataset, using GPT-4o as the backbone LLM and GPT-4.1 as the mapping model, this modification results in a substantial accuracy drop from 75.4% to 32.2%, showing the necessity of this critical component. Moreover, analysis of failure cases reveals two key issues: 1) Incomplete retrieval: The necessary information could be distributed across multiple sections or require cross-referencing. Although the most relevant section is correctly identified in most cases, it alone may not suffice to answer the query – leading the agent to respond with "None of the above." 2) Hallucination: The agent produces confident but unsupported answers. An example failure case for hallucination is provided in Appendix C.1.

These results confirm that all the designed components function as intended, addressing **RQ2**.

6.4 Robustness of IntrAgent to Mapping Model and Input Variability

Impact of Mapping Model Since the evaluation protocol of IntraBench involves mapping short-form answers to multiple-choice (MCQ) selections, it is important to assess the reliability of the mapping model. Using identical prompts on the physics subset, we compare the mappings produced by GPT-4o, GPT-4.1, and DeepSeek-R1 against ground-truth annotations provided by domain experts. Here, we consider short-form answers generated by IntrAgent (with GPT-4o as the backbone) and by vanilla RAG. We also define a *correctness agreement* metric as the ratio of MCQ choices that match the human label in correctness. As shown in Table 5, GPT-4o and GPT-4.1 yield nearly identical accuracy, with GPT-4.1 achieving slightly better

	IntrAgent			
Mapping	**GPT-4o**	**GPT-4.1**	**DeepSeek-R1**	**RAG**
GPT-4o	73.8	76.9	73.8	63.1
GPT-4.1	75.4	78.5	72.3	60.0
DeepSeek-R1	67.7	70.8	76.9	61.5

Table 6: Accuracy (%) of IntrAgent for different mapping models compared with the Vanilla RAG baseline on the physics dataset.

alignment (63 out of 65). Based on these results, we adopt GPT-4.1 as the default mapping model for all main experiments. We hypothesize that LLMs with stronger scientific reasoning and domain expertise could further improve mapping quality—an investigation we leave to future work.

We also conduct a parallel study on the same physics dataset, where short-form answers are generated by IntrAgent using three different backbone models, as well as by the vanilla RAG baseline. Each answer is then mapped to an MCQ option using the same prompt across three different mapping models. As shown in Table 6, although accuracy varies across backbone-mapping model combinations, IntrAgent consistently outperforms the vanilla RAG baseline, which is the strongest among all baselines in our experiments.

Robustness of IntrAgent Under Subpar Section Headings Here, we investigate an interesting edge case where a research paper fails to annotate its sections with standard headings. We select a representative paper from the physics dataset (Rathnayake et al., 2024) and construct three alternative heading styles for the paper: a beginner-style rewrite, a highly noisy version, and a Shakespearean rendition. The full list of rewrites is shown in Appendix D.1. With GPT-4o as the backbone, IntrAgent achieves 89.2% accuracy on the original headings. For all three altered versions, it maintains strong performance, achieving 84.6% accuracy with only a modest drop. Appendix D.2 provides detailed reasoning traces illustrating how IntrAgent effectively handles noisy headings during section ranking.

These results address **RQ3**, demonstrating the robustness of IntrAgent to variations in both evaluation settings and input conditions.

7 Conclusion

We introduce IntraView, a novel task that targets the critical yet underexplored practice of retrieving information from scientific literature. We introduce IntrAgent, the first LLM-based agent specifically designed for IntraView. IntrAgent mimics human reading behavior by first identifying the most relevant sections and then iteratively extracting key information. To enable systematic evaluation of IntraView approaches, including IntrAgent, we present IntraBench– a benchmark spanning five high-impact scientific domains. Experimental results show the superior performance of IntrAgent over baselines on this benchmark.

Limitations

While IntrAgent and IntraBench focus on advancing text-based scientific information retrieval and understanding, this work does not yet incorporate non-textual modalities such as plots, figures, and tables. These visual elements often encapsulate concentrated insights, including experimental trends, quantitative comparisons, and structural relationships, which can be essential for comprehensive scientific reasoning.

In addition, the literature considered in this paper encompasses the major article types, but not all. For example, review papers are not included in our evaluation. In the future work, we will expand our benchmark to include more types of papers and queries.

References

Shubham Agarwal, Gaurav Sahu, Abhay Puri, Issam H Laradji, Krishnamurthy DJ Dvijotham, Jason Stanley, Laurent Charlin, and Christopher Pal. 2024. Litllm: A toolkit for scientific literature review. *arXiv preprint arXiv:2402.01788*.

Nursanti Anggriani, Meksianis Z Ndii, Rika Amelia, Wahyu Suryaningrat, and Mochammad Andhika Aji Pratama. 2022. A mathematical covid-19 model considering asymptomatic and symptomatic classes with waning immunity. *Alexandria Engineering Journal*, 61(1):113–124.

Shahzeb Ansari, Haiping Du, Fazel Naghdy, and David Stirling. 2022. Automatic driver cognitive fatigue detection based on upper body posture variations. *Expert Systems with Applications*, 203:117568.

Anthropic. 2024. Introducing contextual retrieval. https://www.anthropic.com/engineering/contextual-retrieval. Accessed: 2025-09-10.

Tim Baumgärtner, Ted Briscoe, and Iryna Gurevych. 2025. PeerQA: A scientific question answering dataset from peer reviews. In *Proceedings of the 2025 Conference of the Nations of the Americas*

Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 508–544, Albuquerque, New Mexico. Association for Computational Linguistics.

Sudhanshu Kumar Biswas, Jayanta Kumar Ghosh, Susmita Sarkar, and Uttam Ghosh. 2020. Covid-19 pandemic in india: a mathematical model study. *Nonlinear dynamics*, 102(1):537–553.

Jingyi Chai, Shuo Tang, Rui Ye, Yuwen Du, Xinyu Zhu, Mengcheng Zhou, Yanfeng Wang, Yuzhi Zhang, Linfeng Zhang, and Siheng Chen. 2025. Scimaster: Towards general-purpose scientific ai agents, part i. x-master as foundation: Can we lead on humanity's last exam? *arXiv preprint arXiv:2507.05241*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Wang. 2021. HybridQA: a dataset of multi-hop question answering over tabular and textual data. *arXiv preprint*.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. *arXiv preprint*.

James T Dillon. 1984. The classification of research questions. *Review of Educational Research*, 54(3):327–361.

Song Ding, Lunhu Hu, Xing Pan, Dujun Zuo, and Liuwang Sun. 2025. Assessing human situation awareness reliability considering fatigue and mood using eeg data: A bayesian neural network-bayesian network approach. *Reliability Engineering & System Safety*, 260:110962.

Mariia Erzina, Andril Trelin, Olga Guselnikova, Anastasiia Skvortsova, Karolina Strnadova, Vaclav Svorcik, and Oleksiy Lyutakov. 2022. Quantitative detection of α1-acid glycoprotein (agp) level in blood plasma using sers and cnn transfer learning approach. *Sensors and Actuators B: Chemical*, 367:132057.

Elena Escobar-Linero, Manuel Domínguez-Morales, and José Luis Sevillano. 2022. Worker's physical fatigue classification using neural networks. *Expert Systems with Applications*, 198:116784.

Aniruddha Gaikwad, Reza Yavari, Mohammad Montazeri, Kevin Cole, Linkan Bian, and Prahalada Rao. 2020. Toward the digital twin of additive manufacturing: Integrating thermal simulations, sensing, and analytics to detect process faults. *Iise Transactions*, 52(11):1204–1217.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: a survey. *arXiv preprint*.

Peng Gong, Bin Chen, Xuecao Li, Han Liu, Jie Wang, Yuqi Bai, Jingming Chen, Xi Chen, Lei Fang, Shuailong Feng, and 1 others. 2020a. Mapping essential urban land use categories in china (euluc-china): Preliminary results for 2018. *Science Bulletin*, 65(3):182–187.

Peng Gong, Xuecao Li, Jie Wang, Yuqi Bai, Bin Chen, Tengyun Hu, Xiaoping Liu, Bing Xu, Jun Yang, Wei Zhang, and Yuyu Zhou. 2020b. Annual maps of global artificial impervious area (GAIA) between 1985 and 2018. *Remote Sensing of Environment*, 236:111510.

Peng Gong, Jie Wang, Le Yu, Yongchao Zhao, Yuanyuan Zhao, Lu Liang, Zhenguo Niu, Xiaomeng Huang, Haohuan Fu, Shuang Liu, and 1 others. 2013. Finer resolution observation and monitoring of global land cover: First mapping results with landsat tm and etm+ data. *International journal of remote sensing*, 34(7):2607–2654.

Lillemor RM Hallberg. 2010. Some thoughts about the literature review in grounded theory studies. *International journal of qualitative studies on health and well-being*, 5(3):10–3402.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, and Yuxiang Zhang et al. 2024. OlympiadBench: a challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. *arXiv preprint*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, and Bing et al. Qin. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Enahoro Iboi, Oluwaseun O Sharomi, Calistus Ngonghala, and Abba B Gumel. 2020. Mathematical modeling and analysis of covid-19 pandemic in nigeria. *MedRxiv*, pages 2020–05.

Ziyan Jiang, Xueguang Ma, and Wenhu Chen. 2024. Longrag: Enhancing retrieval-augmented generation with long-context llms. *Preprint*, arXiv:2406.15319.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: a dataset

for biomedical research question answering. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 2567–2577, Hong Kong, China. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. TriviaQA: a large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint*.

Muhammad Altaf Khan and Abdon Atangana. 2022. Mathematical modeling and analysis of COVID-19: A study of new variant Omicron. *Physica A: Statistical Mechanics and its Applications*, 599:127452.

Anastasia Krithara, Anastasios Nentidis, Konstantinos Bougiatiotis, and Georgios Paliouras. 2023. BioASQ-QA: A manually curated corpus for Biomedical Question Answering. *Scientific Data*, 10(1):170.

Amit Kumar, Md Redwan Islam, Susu M. Zughaier, Xianyan Chen, and Yiping Zhao. 2024. Precision classification and quantitative analysis of bacteria biomarkers via surface-enhanced Raman spectroscopy and machine learning. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, 320:124627.

Saeb Ragani Lamooki, Sahand Hajifar, Jiyeon Kang, Hongyue Sun, Fadel M Megahed, and Lora A Cavuoto. 2022. A data analytic end-to-end framework for the automated quantification of ergonomic risk factors across multiple tasks using a single wearable sensor. *Applied ergonomics*, 102:103732.

Yoonjoo Lee, Kyungjae Lee, Sunghyun Park, Dasol Hwang, Jaehyeon Kim, Hong-in Lee, and Moontae Lee. 2023. Qasa: advanced question answering on scientific articles. In *International Conference on Machine Learning*, pages 19036–19052. PMLR.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. volume 33, pages 9459–9474.

Jiarui Li, Ye Yuan, and Zehua Zhang. 2024. Enhancing LLM factual accuracy with RAG to counter hallucinations: a case study on domain-specific queries in private knowledge-bases. *arXiv preprint*.

Rui Li, Mingzhou Jin, and Vincent C Paquit. 2021a. Geometrical defect detection for additive manufacturing with machine learning models. *Materials & Design*, 206:109726.

Xiaoting Li, Tengyun Hu, Peng Gong, Shihong Du, Bin Chen, Xuecao Li, and Qi Dai. 2021b. Mapping essential urban land use categories in beijing with a fast area of interest (AOI)-based method. *Remote Sensing*, 13(3):477.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Han Liu, Peng Gong, Jie Wang, Nicholas Clinton, Yuqi Bai, and Shunlin Liang. 2020. Annual dynamics of global land cover and its long-term changes from 1982 to 2015. *Earth System Science Data*, 12(2):1217–1243.

Bo-Ru Lu, Nikita Haduong, Chien-Yu Lin, Hao Cheng, Noah A. Smith, and Mari Ostendorf. 2024a. Efficient encoder-decoder transformer decoding for decomposable tasks. *arXiv preprint*.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024b. The AI scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint*.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024c. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint*.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Preprint*, arXiv:2209.09513.

Jakub Lála, Odhran O'Donoghue, Aleksandar Shtedritski, Sam Cox, Samuel G. Rodriques, and Andrew D. White. 2023. PaperQA: Retrieval-augmented generative agent for scientific research. *arXiv preprint*.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. 2024. Augmenting large language models with chemistry tools. *Nature machine intelligence*, 6(5):525–535.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? A new dataset for open book question answering. *arXiv preprint*.

George A. Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2):81–97.

Niklas Muennighoff, SU Hongjin, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2024. Generative representational instruction tuning. In *The Thirteenth International Conference on Learning Representations*.

Aditya Nagori, Ricardo Accorsi Casonatto, Ayush Gautam, Abhinav Manikantha Sai Cheruvu, and Rishikesan Kamaleswaran. 2025. Open-source agentic hybrid rag framework for scientific literature review. *arXiv preprint arXiv:2508.05660*.

Faïçal Ndaïrou, Iván Area, Juan J. Nieto, and Delfim F. M. Torres. 2020. Mathematical modeling of COVID-19 transmission dynamics with a case study of Wuhan. *Chaos, Solitons & Fractals*, 135:109846.

Duc S. H. Nguyen, Bach G. Truong, Phuong T. Nguyen, Juri Di Rocco, and Davide Di Ruscio. 2025. Teamwork makes the dream work: LLMs-based agents for GitHub readme.md summarization.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the association for computational linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. *arXiv preprint*.

Rathnayake Rathnayake, Zhenghao Zhao, Nathan McLaughlin, Wei Li, Yan Yan, Liaohai Chen, Qian Xie, Christine Wu, Mathew Mathew, and Rong Wang. 2024. Machine learning enabled multiplex detection of periodontal pathogens by surface-enhanced Raman spectroscopy. *International Journal of Biological Macromolecules*, 257:128773.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: a conversational question answering challenge. *arXiv preprint*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. GPQA: a graduate-level google-proof Q&a benchmark. *arXiv preprint*.

Joshua C. Rothstein, Jiaheng Cui, Yanjun Yang, Xianyan Chen, and Yiping Zhao. 2024. Ultra-sensitive detection of PFASs using surface enhanced Raman scattering and machine learning: a promising approach for environmental analysis. *Sensors & Diagnostics*, 3(8):1272–1284.

Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Michael Moor, Zicheng Liu, and Emad Barsoum. 2025. Agent laboratory: Using llm agents as research assistants. *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 5977–6043.

Luis Javier Segura, Tianjiao Wang, Chi Zhou, and Hongyue Sun. 2021. Online droplet anomaly detection from streaming videos in inkjet printing. *Additive Manufacturing*, 38:101835.

Seyyed Hadi Seifi, Wenmeng Tian, Haley Doude, Mark A Tschopp, and Linkan Bian. 2019. Layer-wise modeling and anomaly detection for laser-based additive manufacturing. *Journal of Manufacturing Science and Engineering*, 141(8):081013.

Sneha Senapati, Manleen Kaur, Neetu Singh, Smita S. Kulkarni, and Jitendra Pratap Singh. 2024. Affordable paper-based surface-enhanced raman scattering substrates containing silver nanorods using glancing-angle deposition for nosocomial infection detection. *ACS Applied Nano Materials*, 7(7):6736–6748.

Yijia Shao, Yucheng Jiang, Theodore A. Kanell, Peter Xu, Omar Khattab, and Monica S. Lam. 2024. Assisting in writing wikipedia-like articles from scratch with large language models. *Preprint*, arXiv:2402.14207.

Chandan Kumar Singh, Devesh Kumar, Vipul Sanap, and Rajesh Sinha. 2025. Llm-rspf: Large language model-based robotic system planning framework for domain specific use-cases. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 7277–7286.

Michael D Skarlinski, Sam Cox, Jon M Laurent, James D Braza, Michaela Hinks, Michael J Hammerling, Manvitha Ponnapati, Samuel G Rodriques, and Andrew D White. 2024. Language agents achieve superhuman synthesis of scientific knowledge. *arXiv preprint arXiv:2409.13740*.

Bingyi Su, Liwei Qing, Lu Lu, SeHee Jung, Xiaolei Fang, and Xu Xu. 2024a. Enhancing data privacy in human factors studies with federated learning. *Human Factors*, page 00187208251348025.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024b. DRAGIN: Dynamic Retrieval Augmented Generation based on the Information Needs of Large Language Models. *arXiv preprint*.

Piwat Suppawittaya. 2021. The effectiveness of chunking methods for enhancing short-term memory of textual information. *Psychology and Education Journal*, 57:6313–6327.

Peggy Tang, Junbin Gao, Lei Zhang, and Zhiyong Wang. 2023. Efficient and Interpretable Compressive Text Summarisation with Unsupervised Dual-Agent Reinforcement Learning. *arXiv preprint*.

Xiangru Tang, Tianyu Hu, Muyang Ye, Yanjun Shao, Xunjian Yin, Siru Ouyang, Wangchunshu Zhou, Pan Lu, Zhuosheng Zhang, and Yilun Zhao et al. 2025. ChemAgent: Self-updating library in large language models improves chemical reasoning. *arXiv preprint*.

William John Thrift and Regina Ragan. 2019. Quantification of analyte concentration in the single molecule regime using convolutional neural networks. *Analytical chemistry*, 91(21):13337–13342.

Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, and Fukai Shang et al. 2024a. MinerU: An open-source solution for precise document content extraction.

Ke Wang, Houxing Ren, Aojun Zhou, Zimu Lu, Sichun Luo, Weikang Shi, Renrui Zhang, Linqi Song, Mingjie Zhan, and Hongsheng Li. 2023a. MathCoder: Seamless code integration in LLMs for enhanced mathematical reasoning. *arXiv preprint*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023b. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Preprint*, arXiv:2002.10957.

Xiaohua Wang, Zhenghua Wang, Xuan Gao, Feiran Zhang, Yixin Wu, Zhibo Xu, Tianyuan Shi, Zhengyuan Wang, Shizheng Li, Qi Qian, Ruicheng Yin, Changze Lv, Xiaoqing Zheng, and Xuanjing Huang. 2024b. Searching for best practices in retrieval-augmented generation. *Preprint*, arXiv:2407.01219.

Zheng Wang, Zhongyang Li, Zeren Jiang, Dandan Tu, and Wei Shi. 2024c. Crafting personalized agents through retrieval-augmented generation on editable memory graphs. *arXiv preprint*.

Zheng Wang, Shu Teo, Jieer Ouyang, Yongjun Xu, and Wei Shi. 2024d. M-RAG: Reinforcing large language model performance through retrieval-augmented generation with multiple partitions. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 1966–1978, Bangkok, Thailand. Association for Computational Linguistics.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. *arXiv preprint*.

Sharon Whitfield and Melissa A Hofmann. 2023. Elicit: Ai literature review research assistant. *Public Services Quarterly*, 19(3):201–207.

Dayong Wu, Jiaqi Li, Baoxin Wang, Honghong Zhao, Siyuan Xue, Yanjie Yang, Zhijun Chang, Rui Zhang, Li Qian, and Bo Wang et al. 2024a. SparkRA: a retrieval-augmented knowledge service system based on spark large language model. In *Proceedings of the 2024 conference on empirical methods in natural language processing: System demonstrations*, pages 382–389, Miami, Florida, USA. Association for Computational Linguistics.

Shangyu Wu, Ying Xiong, Yufei Cui, Haolun Wu, Can Chen, Ye Yuan, Lianming Huang, Xue Liu, Tei-Wei Kuo, and Nan Guan et al. 2025. Retrieval-augmented generation for natural language processing: a survey. *arXiv preprint*.

Yiran Wu, Feiran Jia, Shaokun Zhang, Hangyu Li, Erkang Zhu, Yue Wang, Yin Tat Lee, Richard Peng, Qingyun Wu, and Chi Wang. 2024b. MathChat: Converse to tackle challenging math problems with LLM agents. *arXiv preprint*.

Qi Xin, Quyu Kong, Hongyi Ji, Yue Shen, Yuqi Liu, Yan Sun, Zhilin Zhang, Zhaorong Li, Xunlong Xia, Bing Deng, and 1 others. 2024. Bioinformatics agent (bia): unleashing the power of large language models to reshape bioinformatics workflow.

Sarjana Yadav, Sneha Senapati, Samir Kumar, Shashank K. Gahlaut, and Jitendra P. Singh. 2022. GLAD based advanced nanostructures for diversified biosensing applications: Recent progress. *Biosensors*, 12(12):1115.

Yanjun Yang, Jiaheng Cui, Dan Luo, Jackelyn Murray, Xianyan Chen, Sebastian Hulck, Ralph A Tripp, and Yiping Zhao. 2024. Rapid detection of sars-cov-2 variants using an angiotensin-converting enzyme 2-based surface-enhanced raman spectroscopy sensor enhanced by covari deep learning algorithms. *ACS sensors*, 9(6):3158–3169.

Yanjun Yang, Hao Li, Les Jones, Jackelyn Murray, James Haverstick, Hemant K. Naikare, Yung-Yi C. Mosley, Ralph A. Tripp, Bin Ai, and Yiping Zhao. 2023. Rapid detection of SARS-CoV-2 RNA in human nasopharyngeal specimens using surface-enhanced raman spectroscopy and deep learning algorithms. *ACS Sensors*, 8(1):297–307.

Fuda Ye, Shuangyin Li, Yongqi Zhang, and Lei Chen. 2024. R^2AG: Incorporating retrieval information into retrieval augmented generation. *arXiv preprint*.

Antonio Jimeno Yepes, Yao You, Jan Milczek, Sebastian Laverde, and Renyu Li. 2024. Financial report chunking for effective retrieval augmented generation. *Preprint*, arXiv:2402.05131.

Da Yin, Faeze Brahman, Abhilasha Ravichander, Khyathi Chandu, Kai-Wei Chang, Yejin Choi, and Bill Yuchen Lin. 2024. Agent lumos: Unified and modular training for open-source language agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12380–12403.

Haining Zhang, Jingyuan Huang, Xiaoge Zhang, and Chak-Nam Wong. 2024a. Autonomous optimization of process parameters and in-situ anomaly detection in aerosol jet printing by an integrated machine learning approach. *Additive Manufacturing*, 86:104208.

Huan Zhang, Yu Song, Ziyu Hou, Santiago Miret, and Bang Liu. 2024b. HoneyComb: a flexible LLM-based agent system for materials science. *arXiv preprint*.

Yu Zhang, Xiusi Chen, Bowen Jin, Sheng Wang, Shuiwang Ji, Wei Wang, and Jiawei Han. 2024c. A comprehensive survey of scientific large language models and their applications in scientific discovery. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages

8783–8817, Miami, Florida, USA. Association for Computational Linguistics.

Yiping Zhao, Amit Kumar, and Yanjun Yang. 2024. Unveiling practical considerations for reliable and standardized SERS measurements: lessons from a comprehensive review of oblique angle deposition-fabricated silver nanorod array substrates. *Chemical Society Reviews*, 53(2):1004–1057.

Zhiying Zhu, Yiming Yang, and Zhiqing Sun. 2024. HaluEval-wild: Evaluating hallucinations of language models in the wild. *arXiv preprint*.

A Details about IntraBench

The benchmark was created through a systematic process: first selecting representative science literature, then manually crafting questions that span different scientific four task-oriented categories: study subject & experimental setup, data characteristics & collection, technical approach & details, conclusions & results. Special attention was given to ensuring professional quality and coverage across multiple subfields.

A.1 Research Fields in IntraBench

Surface-enhanced Raman Spectroscopy (SERS) *Surface-enhanced raman spectroscopy*(SERS) represents a significant advancement in applied physics, particularly within the domain of optical physics. At its core, SERS is an extension of Raman spectroscopy, a technique that analyzes the inelastic scattering of light to provide insights into molecular vibrations. The enhancement in SERS arises when molecules are adsorbed onto nanostructured metallic surfaces, such as gold or silver, leading to a substantial increase in the Raman signal. This amplification is primarily attributed to localized surface plasmon resonances—coherent oscillations of conduction electrons at the metal surface excited by incident light—which generate intense electromagnetic fields near the surface, thereby boosting the Raman scattering efficiency of nearby molecules. The remarkable sensitivity of SERS enables the detection of analytes at extremely low concentrations, down to the single-molecule level. This capability is particularly valuable in various engineering applications where identifying trace amounts of substances is crucial. For instance, in public safety, SERS-based sensors have been developed for the rapid and accurate detection of biohazardous materials, including chemical agents, viruses, and bacteria(Thrift and Ragan, 2019; Rathnayake et al., 2024; Yang et al., 2024, 2023; Erzina et al., 2022). Such applications benefit from SERS's ability to provide specific molecular fingerprints, facilitating the identification of dangerous substances even in complex environments.

Remote Sensing Land cover and land use classification is a central research direction in remote sensing science, aiming to automatically identify and spatially represent surface types and human activity patterns based on multi-source remote sensing data(Gong et al., 2020b; Liu et al., 2020; Gong et al., 2013; Li et al., 2021b; Gong et al., 2020a). This field widely utilizes medium- and high-resolution satellite imagery from optical sensors (e.g., Landsat, Sentinel-2) and synthetic aperture radar (e.g., Sentinel-1), employing techniques such as supervised classification, object-based analysis, time-series processing, and deep learning for high-accuracy mapping. In recent years, deep convolutional neural networks such as U-Net, DeepLab, and HRNet have been extensively applied to semantic segmentation of high-resolution imagery, significantly improving the extraction accuracy of fine-scale urban features such as buildings, roads, and impervious surfaces. In addition, the integration of optical and radar data with auxiliary variables—such as nighttime lights and population density—has enhanced model performance in complex urban environments and facilitated dynamic monitoring of land use changes. These studies are of great value for supporting global change research, environmental assessment, and sustainable urban planning.

Infectious-disease Modeling Infectious-disease modeling and forecasting play a central role in public health by enabling researchers and policymakers to anticipate the spread and burden of infectious diseases. Among the tools available, mathematical modeling—particularly compartmental models such as SIR and SEIR—has proven indispensable for simulating disease transmission dynamics, assessing intervention strategies, and guiding resource allocation(Khan and Atangana, 2022; Biswas et al., 2020; Anggriani et al., 2022; Iboi et al., 2020; Ndaïrou et al., 2020). The COVID-19 pandemic has exemplified the value of these models, as they have been critical for estimating infection trajectories, evaluating the timing and effectiveness of control measures, and informing real-time policy responses during a rapidly evolving global crisis. Accurate modeling has thus been essential for understanding and managing the impact of COVID-19 across different settings and timeframes.

Human Factor Human factors is a cornerstone of industrial engineering, focusing on the design and evaluation of systems that balance human well-being and overall system performance. Human factors has addressed challenges in physical workload, cognitive workload, and human–machine interaction across domains such as manufacturing, transportation, and healthcare. A critical area is the classification and prediction of human fatigue and

risk. This is because prolonged exposure to repetitive motions, awkward postures, or high-intensity workloads can impair human performance, increase error rates, and elevate injury risk. Recent studies have leveraged wearable sensors, biomechanical modeling, and machine learning methods to detect early indicators of fatigue and quantify risks, enabling proactive intervention(Lamooki et al., 2022; Escobar-Linero et al., 2022; Ansari et al., 2022; Ding et al., 2025; Su et al., 2024a). Such approaches underscore the evolving role of human factors research in advancing resilient, human-centered industrial systems.

Additive Manufacturing Additive manufacturing (AM), rooted in materials science and manufacturing engineering, refers to a family of processes that build components layer by layer directly from digital models. Over the past decades, AM has evolved from rapid prototyping into a transformative production technology, enabling complex geometries, lightweight structures, and customized products across aerospace, biomedical, and energy sectors. While the promise of AM lies in its design freedom and material efficiency, ensuring consistent quality remains a central challenge. Quality control in AM encompasses detecting defects such as porosity and dimensional inaccuracy, which can critically impact the structural integrity and performance of printed parts. Researchers increasingly employ in-situ monitoring, computer vision, and machine learning-based defect detection to address variability AM processes(Zhang et al., 2024a; Li et al., 2021a; Seifi et al., 2019; Segura et al., 2021; Gaikwad et al., 2020). Such quality-control efforts are pivotal to advancing AM toward reliable, industrial-scale deployment.

A.2 Research Literature in `IntraBench`

Table 7 contains all 25 papers, five per field, used in `IntraBench`.

A.3 Research Questions in `IntraBench`

In total, `IntraBench` comprises 63 research questions, systematically categorized into four task-oriented categories: (1) Study subject & experimental setup, (2) Data characteristics & collection, (3) Technical approach & details, and (4) Conclusions & results. These questions are designed to elicit critical information from scientific papers across five domains. Specifically, the *SERS in chemistry physics* dataset contributes 13 questions, the *infectious-disease modeling* in public health dataset contributes 11 questions, the *remote sensing* in earth science dataset contributes 12 questions, the *human performance sensing* in engineering dataset contributes 13 questions, and the *additive manufacturing* in material science dataset contributes 14 questions. The complete lists are provided in Table 8, Table 9, Table 10, Table 11, and Table 12, respectively.

Across all domains, the 63 questions remain balanced across the first three task-oriented categories—Study subject & experimental setup, Data characteristics & collection, and Technical approach & details—each covering foundational aspects of experimental design, data acquisition, and analytical methodology. The remaining questions belong to the Conclusions & results category, emphasizing the outcome evaluation and performance reporting aspects of scientific research.

While the overall structure 63 consistent, each domain reflects distinct emphases. The *infectious-disease modeling* dataset prioritizes experimental setup due to the importance of compartmental structures and intervention parameters. The *remote sensing* dataset highlights data characteristics such as sensor type, spatial resolution, and temporal coverage. The *SERS in physics* dataset shows a balanced distribution aligned with typical SERS workflows from substrate preparation to data-driven analysis. The *humanfactor in engineering* dataset emphasizes human performance measurement and multimodal sensing setups, reflecting the diversity of sensor placements and physiological variables in human studies. The *Additive manufacturing in material science* dataset focuses on process monitoring and machine learning evaluation within additive manufacturing workflows, emphasizing material–process–defect relationships and reproducible modeling strategies.

To ensure usability and reproducibility, each research question was expert-annotated with detailed explanatory notes.

A.4 Main Result Report over Task-oriented Categories

To further analyze performance differences among backbone large language models (LLMs), we evaluate `IntrAgent` across four *task-oriented categories*: (1) *Study subject & experimental setup (S&E)*, (2) *Data characteristics & collection (D&C)*, (3) *Technical approach & details (T&D)*, and (4) *Conclusions & results (C&R)*.

Title	Ref.
Public Health - Infectious-disease Modeling	
Mathematical modeling and analysis of COVID-19: a study of new variant Omicron	(Khan and Atangana, 2022)
COVID-19 pandemic in India: a mathematical model study	(Biswas et al., 2020)
A mathematical COVID-19 model considering asymptomatic and symptomatic classes with waning immunity	(Anggriani et al., 2022)
Mathematical modeling and analysis of COVID-19 pandemic in Nigeria	(Iboi et al., 2020)
Mathematical modeling of COVID-19 transmission dynamics with a case study of Wuhan	(Ndaïrou et al., 2020)
Physics - Surface Enhanced Raman Spectroscopy	
Quantification of analyte concentration in the single molecule regime using convolutional neural networks	(Thrift and Ragan, 2019)
Machine learning enabled multiplex detection of periodontal pathogens by surface-enhanced Raman spectroscopy	(Rathnayake et al., 2024)
Rapid detection of SARS-CoV-2 variants using an angiotensin-converting enzyme 2-based surface-enhanced Raman spectroscopy sensor enhanced by CoVari deep learning algorithms	(Yang et al., 2024)
Rapid detection of SARS-CoV-2 RNA in human nasopharyngeal specimens using surface-enhanced Raman spectroscopy and deep learning algorithms	(Yang et al., 2023)
Quantitative detection of α1-acid glycoprotein (AGP) level in blood plasma using SERS and CNN transfer learning approach	(Erzina et al., 2022)
Earth Science - Remote Sensing	
Annual maps of global artificial impervious area (GAIA) between 1985 and 2018	(Gong et al., 2020b)
Annual dynamics of global land cover and its long-term changes from 1982 to 2015	(Liu et al., 2020)
Finer resolution observation and monitoring of global land cover: first mapping results with Landsat TM and ETM+ data	(Gong et al., 2013)
Mapping essential urban land use categories in Beijing with a fast area of interest (AOI)-based method	(Li et al., 2021b)
Mapping essential urban land use categories in China (EULUC-China): preliminary results for 2018	(Gong et al., 2020a)
Engineering - Human Factor	
A data analytic end-to-end framework for the automated quantification of ergonomic risk factors across multiple tasks using a single wearable sensor	(Lamooki et al., 2022)
Assessing human situation awareness reliability considering fatigue and mood using EEG data: a Bayesian neural network-Bayesian network approach	(Ding et al., 2025)
Automatic driver cognitive fatigue detection based on upper body posture variations	(Ansari et al., 2022)
Enhancing data privacy in human factors studies with federated learning	(Su et al., 2024a)
Worker's physical fatigue classification using neural networks	(Escobar-Linero et al., 2022)
Material Science - Additive Manufacturing	
Autonomous optimization of process parameters and in-situ anomaly detection in aerosol jet printing by an integrated machine learning approach	(Zhang et al., 2024a)
Geometrical defect detection for additive manufacturing with machine learning models	(Li et al., 2021a)
Layer-wise modeling and anomaly detection for laser-based additive manufacturing	(Seifi et al., 2019)
Online droplet anomaly detection from streaming videos in inkjet printing	(Segura et al., 2021)
Toward the digital twin of additive manufacturing: integrating thermal simulations, sensing, and analytics to detect process faults	(Gaikwad et al., 2020)

Table 7: Literature across five scientific domains used in our benchmark.

Task-oriented Category	Research Question Q
Study subject & experimental setup	What are the main analytes type studied? What are the material and structure, or morphology of the SERS substrates used? How many analytes are investigated?
Data characteristics & collection	What is the excitation laser wavelength used for SERS measurements? What is the spectral range collected for the analysis of the analytes? How many spectra are collected per analyte under each experimental condition? How many experimental replications are conducted to ensure reproducibility?
Technical approach & details	What is the primary machine learning task addressed in this study? Which machine learning algorithm is implemented? What data splitting strategy is applied, and the parameters? How many epochs are used during model training?
Conclusions & results	What performance metrics are employed to evaluate the machine learning models? What are the reported performance values?

Table 8: Research Questions in IntraBench: *SERS in chemistry physics* (Phys)

Task-oriented Category	Research Question Q
Study subject & experimental setup	Into how many compartments is the population divided in the model? What is the initial susceptible population of the model? What is the initial infected population of the model? How many interventions are addressed in the paper?
Data characteristics & collection	What is the source location or country of origin for the data used in this study?
Technical approach & details	What is the model used in this paper? What is the transmission rate? What is the disease-induced mortality rate? What values of the basic reproduction number were considered in the model?
Conclusions & results	What are the novel contributions of the paper? What are the limitations of the paper?

Table 9: Research Questions in IntraBench: *infectious-disease modeling* in Public Health (PH)

These evaluations reveal how each backbone contributes to different aspects of literature reasoning. Figure 3 visualizes the results across seven LLMs. Two overall trends emerge. First, IntrAgent maintains balanced performance across all categories regardless of the backbone, confirming its robustness and model-agnostic design. Second, although individual models show slight strengths in specific areas, the overall variation remains moderate, indicating that the contextual reading and iterative synthesis strategy of IntrAgent mitigates backbone dependency.

In detail, GPT-4.1 achieves the highest overall balance (S&E = 66.8, D&C = 72.7, T&D = 83.4, C&R = 64.3), especially excelling in technical reasoning. o4-mini performs comparably (65.7, 83.3, 77.5, 72.3), outperforming GPT-4.1 in D&C and C&R. Gemini 2.5 Pro maintains uniformly high results (76.7, 76.0, 75.0, 64.8), reflecting strong generalization. DeepSeek-R1 (56.4, 71.8, 82.6, 67.9) and o3 (60.0, 76.7, 82.5, 68.0) show stable interpretative abilities, while Llama-3.1-70B (62.2, 65.6, 77.9, 75.0) exhibits strength in conclusion-oriented reasoning.

Overall, GPT-4.1 and o4-mini demonstrate the most consistent and well-rounded performance, while Gemini 2.5 Pro and Llama-3.1-70B show domain-specific advantages. These results suggest that IntrAgent 's architecture effectively harmonizes reasoning depth and retrieval accuracy across heterogeneous LLMs, ensuring adaptability without overfitting to any specific model.

B Additional Experiments and Reports

B.1 Main Experiment - Domain-level Accuracy Breakdown

Table 2 in the main text reports averaged accuracies for every *baseline–LLM* combinations. For

Task-oriented Category	Research Question Q
Study subject & experimental setup	What is the number of land-cover / land-use classes classified in this study? What is the spatial extent of the study area? What is the geographic type of the study area?
Data characteristics & collection	What is the temporal scope of the data used? What type of remote sensing data is used? Which specific satellite data is used? What is the spatial resolution of the primary imagery used? Are auxiliary features used beyond raw spectral bands?
Technical approach & details	What type of model is implemented in this study? What performance metrics are reported?
Conclusions & results	Is any comparative analysis included? What is the reported overall accuracy (OA)?

Table 10: Research Questions in `IntraBench`: *remote sensing* in Earth Science (ES)

Task-oriented Category	Research Question Q
Study subject & experimental setup	What are the subjects' occupational roles? What specific task or activity are the subjects performing? What is the study context or environment in which participants perform the tasks? If the data are referenced from prior work, please indicate the source.
Data characteristics & collection	What are the primary sensing modalities or measurement instruments employed in the study to capture human performance and physiological responses? What is the anatomical or body placement of the sensors used in the study? What is the sampling rate (Hz)? What is the total number of participants involved in the study?
Technical approach & details	How are the participants' physical, cognitive, or perceptual states assessed or reflected in the study? What is the primary modeling objective in this study? What is the data partitioning strategy used during model training, and what are the parameters? What is the number of epochs used during model training (i.e., how many complete passes through the entire training dataset)?
Conclusions & results	Which performance metrics are used to assess the effectiveness of the machine learning models? What are the reported values for the performance metrics used to evaluate the machine learning models?

Table 11: Research Questions in `IntraBench`: *Human performance sensing in engineering* (Engr)

each pair we compute the simple arithmetic mean over the five domain datasets shown below, so every domain carries equal weight: *SERS* in physics, *infectious-disease modeling* in public health, *remote sensing* in earth science, *human performance sensing* in engineering, *additive manufacturing* in material science.

While the main table focuses on these aggregated averages, the following tables (Table 13, 14, 15, 16, 17) present the detailed per-domain accuracies for all *baseline–LLM* combinations. Each table corresponds to one of the five domains listed above. Boldface highlights the best score within each domain, allowing readers to examine where specific retrieval strategies or LLM backbones perform most effectively.

B.2 Statistical Analysis

B.2.1 Significance Testing of `IntrAgent` Performance

To rigorously assess whether `IntrAgent` achieves statistically significant performance gains over existing baselines, we conducted a one-sided Wilcoxon signed-rank test across all five domains and seven backbones ($n = 35$ paired samples per baseline). For each baseline, accuracies were paired with those of `IntrAgent` on identical backbones within the same domain.

Task-oriented Category	Research Question Q
Study subject & experimental setup	What type of additive manufacturing process is studied? What type of material is used for printing? What kind of shape or product is printed? What is the primary defect being studied?
Data characteristics & collection	What sensors are used to measure the process? What is the sampling rate (Hz)? If relevant, what is the spatial resolution (μm)?
Technical approach & details	What is the machine learning objective in this study? What machine learning algorithm is used? How are the data split during machine learning? How many replications are conducted during machine learning, if any? How many epochs are used during machine learning, if any?
Conclusions & results	What metrics are used to evaluate the machine learning models? What are the values for these metrics?

Table 12: Research Questions in IntraBench: *Additive manufacturing in material science* (MS)

	Baseline	GPT-4o	GPT-4.1	DS-R1	o3	o4-mini	Gemini-2.5 Pro	Llama-3.1-70B
RAG	Vanilla RAG all-MiniLM-L6-v2	60.0	66.2	64.6	58.5	60.0	60.0	67.7
	Vanilla RAG E5-mistral-7b-instruct	61.5	70.7	63.8	63.1	64.6	64.6	66.2
	Vanilla RAG GritLM-7B	60.0	67.6	58.5	58.5	60.0	58.5	64.6
	Contextual RAG E5-mistral-7b-instruct	58.5	66.2	58.5	58.5	60.0	53.9	58.5
	Contextual RAG GritLM-7B	53.9	67.6	64.6	58.5	60.0	60.0	58.5
	DRAGIN	38.5	43.0	47.7	43.1	46.2	44.6	46.2
	R^2AG	58.5	56.9	58.5	56.9	56.9	56.9	55.4
	LongRAG	64.6	70.7	64.6	56.9	58.5	55.4	56.9
Agent	LUMOS	52.3	50.8	50.8	53.8	56.9	56.9	56.9
	PaperQA2	49.2	50.8	52.3	49.2	52.3	50.8	52.3
	Agentic-Hybrid-RAG	60.0	63.1	61.5	55.4	55.4	55.4	53.8
	SciMaster	58.5	58.5	63.1	53.8	58.5	55.4	55.4
	IntrAgent (Ours)	75.4	78.5	78.5	76.9	81.5	78.5	72.3

Table 13: Accuracies (%) on IntraBench, Physics

	Baseline	GPT-4o	GPT-4.1	DS-R1	o3	o4-mini	Gemini-2.5 Pro	Llama-3.1-70B
RAG	Vanilla RAG all-MiniLM-L6-v2	52.7	60.0	65.6	58.2	61.8	58.2	54.5
	Vanilla RAG E5-mistral-7b-instruct	52.7	61.7	65.6	54.6	56.4	54.6	58.2
	Vanilla RAG GritLM-7B	52.7	61.7	58.2	56.4	52.7	54.5	58.2
	Contextual RAG E5-mistral-7b-instruct	56.4	58.3	65.6	58.2	54.5	56.4	56.4
	Contextual RAG GritLM-7B	58.3	58.3	54.6	56.4	58.2	56.4	54.5
	DRAGIN	41.8	40.0	47.3	41.8	47.3	43.6	43.6
	R^2AG	54.5	58.2	61.8	50.9	49.1	50.9	50.9
	LongRAG	50.9	54.6	60.0	47.3	49.1	50.9	47.3
Agent	LUMOS	43.6	47.3	49.1	49.1	47.3	47.3	52.7
	PaperQA2	38.2	45.5	47.3	47.3	45.5	47.3	50.9
	Agentic-Hybrid-RAG	54.5	58.2	60.0	52.7	52.7	50.9	50.9
	SciMaster	58.2	54.5	60.0	50.9	50.9	49.1	50.9
	IntrAgent (Ours)	58.2	69.1	69.1	65.5	67.3	70.9	69.1

Table 14: Accuracies (%) on IntraBench, Public Health

Hypotheses

$$H_0 : \text{Median}(\texttt{IntrAgent} - \text{Baseline}) = 0,$$
$$H_1 : \text{Median}(\texttt{IntrAgent} - \text{Baseline}) > 0.$$

Here H_0 indicates no improvement of IntrAgent over the baseline, while H_1 tests for significantly higher performance, result shown 18.

Interpretation Across all baselines, the median performance improvements range from $+10.9$ to $+27.3$ percentage points. The Holm–Bonferroni–adjusted p-values ($p_{\text{adj}} < 10^{-6}$) indicate that these gains are highly significant and not attributable to random variation. Even after correction for multiple comparisons, all 12 tests reject the null hypothesis H_0 at $\alpha = 0.05$. These

	Baseline	GPT-4o	GPT-4.1	DS-R1	o3	o4-mini	Gemini-2.5 Pro	Llama-3.1-70B
RAG	Vanilla RAG all-MiniLM-L6-v2	60.0	58.3	60.0	58.3	61.7	55.0	60.0
	Vanilla RAG E5-mistral-7b-instruct	60.0	60.0	58.3	58.3	61.7	51.7	60.0
	Vanilla RAG GritLM-7B	61.7	60.0	58.3	53.3	51.7	51.7	61.7
	Contextual RAG E5-mistral-7b-instruct	58.3	61.8	60.0	51.7	51.7	58.3	61.7
	Contextual RAG GritLM-7B	58.3	58.2	60.0	48.3	56.7	58.3	60.0
	DRAGIN	40.0	43.3	40.0	43.3	45.0	46.7	46.7
	R^2AG	58.3	56.7	58.3	55.0	55.0	53.3	60.0
	LongRAG	60.0	63.3	63.3	53.3	53.3	53.3	58.3
Agent	LUMOS	46.7	51.7	46.7	53.3	56.7	51.7	55.0
	PaperQA2	51.7	50.0	53.3	50.0	45.0	48.3	55.0
	Agentic-Hybrid-RAG	58.3	58.3	58.3	56.7	55.0	55.0	60.0
	SciMaster	56.7	56.7	60.0	56.7	58.3	60.0	61.7
	IntrAgent (Ours)	63.3	70.0	70.0	70.0	69.1	70.1	65.0

Table 15: Accuracies (%) on IntraBench, Earth Science

	Baseline	GPT-4o	GPT-4.1	DS-R1	o3	o4-mini	Gemini-2.5 Pro	Llama-3.1-70B
RAG	Vanilla RAG all-MiniLM-L6-v2	64.6	61.5	60.0	58.5	55.4	61.5	52.3
	Vanilla RAG E5-mistral-7b-instruct	60.0	58.5	61.5	60.0	60.0	60.0	55.4
	Vanilla RAG GritLM-7B	63.1	56.9	61.5	61.5	63.1	63.1	55.4
	Contextual RAG E5-mistral-7b-instruct	63.1	64.3	60.0	61.5	56.9	63.1	52.3
	Contextual RAG GritLM-7B	66.2	64.3	60.0	60.0	58.4	63.1	60.0
	DRAGIN	47.7	50.8	52.3	46.2	44.6	44.6	46.2
	R^2AG	60.0	61.5	63.1	60.0	56.9	56.9	56.9
	LongRAG	67.7	69.2	70.8	63.1	66.2	63.1	61.5
Agent	LUMOS	55.4	58.5	64.6	58.5	56.9	58.5	53.8
	PaperQA2	50.8	52.3	58.5	53.8	47.7	52.3	52.3
	Agentic-Hybrid-RAG	64.6	61.5	66.2	61.5	63.1	61.5	58.5
	SciMaster	64.6	60.0	66.2	61.5	61.5	60.0	56.9
	IntrAgent (Ours)	80.0	81.5	78.5	81.5	75.4	76.9	66.2

Table 16: Accuracies (%) on IntraBench, Engineering

	Baseline	GPT-4o	GPT-4.1	DS-R1	o3	o4-mini	Gemini-2.5 Pro	Llama-3.1-70B
RAG	Vanilla RAG all-MiniLM-L6-v2	64.3	60.0	71.4	68.6	68.6	74.3	61.4
	Vanilla RAG E5-mistral-7b-instruct	62.9	70.0	70.0	65.7	64.3	68.6	62.9
	Vanilla RAG GritLM-7B	64.3	70.0	71.4	68.6	64.3	64.3	67.1
	Contextual RAG E5-mistral-7b-instruct	67.1	68.6	70.0	65.7	68.6	62.9	65.7
	Contextual RAG GritLM-7B	67.1	65.7	68.6	68.6	70.0	70.0	62.9
	DRAGIN	44.3	45.7	47.1	45.7	51.4	50.0	44.3
	R^2AG	65.7	64.3	65.7	60.0	58.6	60.0	57.1
	LongRAG	67.1	65.7	68.6	64.3	64.3	62.9	62.9
Agent	LUMOS	52.9	57.1	65.7	61.4	64.3	60.0	61.4
	PaperQA2	48.6	45.7	58.6	58.6	55.7	57.1	58.6
	Agentic-Hybrid-RAG	61.4	60.0	65.7	61.4	62.9	61.4	60.0
	SciMaster	57.1	58.6	67.1	62.9	61.4	61.4	60.0
	IntrAgent (Ours)	72.9	80.0	75.7	72.9	75.7	82.9	71.4

Table 17: Accuracies (%) on IntraBench, Material Science

findings confirm that IntrAgent consistently and significantly outperforms every baseline across domains and backbones, demonstrating robust, model-agnostic performance advantages.

B.2.2 Statistical Analysis of Paper Length and IntrAgent Performance

IntraBench spans a wide range of paper lengths, with an average of 15.2 pages (SD = 9.04), a median of 13 pages, a minimum of 6 pages, and a maximum of 49 pages. To examine whether paper length affects agent performance, we build a mixed-effects linear regression model with the per-paper accuracy of IntrAgent as the response and paper length as the predictor. Since each paper is evaluated using three backbone LLMs (GPT-4o, GPT-4.1, and DeepSeek-R1), we treat backbone choice as repeated measurements.

Baseline	Median Δ(%)	p_{raw}	p_{adj}
Vanilla RAG all-MiniLM-L6-v2	+11.7	2.9×10^{-11}	3.5×10^{-10}
Vanilla RAG E5-mistral-7b-instruct	+10.9	2.9×10^{-11}	3.5×10^{-10}
Vanilla RAG GritLM-7B	+11.7	2.9×10^{-11}	3.5×10^{-10}
Contextual RAG E5-mistral-7b-instruct	+12.8	2.9×10^{-11}	3.5×10^{-10}
Contextual RAG GritLM-7B	+12.9	5.8×10^{-11}	3.5×10^{-10}
DRAGIN	+27.3	2.9×10^{-11}	3.5×10^{-10}
R^2AG	+16.8	2.9×10^{-11}	3.5×10^{-10}
LongRAG	+12.3	2.9×10^{-11}	3.5×10^{-10}
LUMOS	+18.5	2.9×10^{-11}	3.5×10^{-10}
PaperQA2	+21.8	2.9×10^{-11}	3.5×10^{-10}
Agentic-Hybrid-RAG	+14.1	2.9×10^{-11}	3.5×10^{-10}
SciMaster	+14.6	1.8×10^{-7}	1.8×10^{-7}

Table 18: Wilcoxon signed-rank test comparing IntrAgent with all baselines across 35 paired samples (5 domains $\times$ 7 backbones). All tests are one-sided (H_1: IntrAgent $>$ baseline). Reported p-values are Holm–Bonferroni corrected for multiple comparisons. All comparisons remain significant at $p_{\text{adj}} < 0.001$.

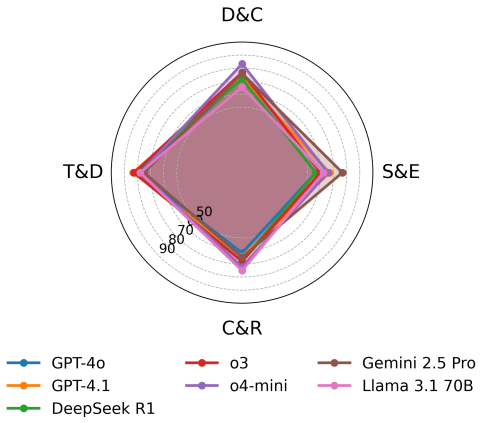

Figure 3: Radar plot comparing the performance of seven backbone LLMs—GPT-4o, GPT-4.1, DeepSeek-R1, o3, o4-mini, Gemini 2.5 Pro, and Llama-3.1-70B—across four research-question categories: Study subject & experimental setup (S&E), Data characteristics & collection (D&C), Technical approach & details (T&D), and Conclusions & results (C&R). GPT-4.1 and o4-mini show the strongest overall balance, while Gemini 2.5 Pro and Llama-3.1-70B demonstrate stable domain-specific strengths. The plot highlights that despite moderate variation across backbones, IntrAgent remains consistently robust across all task categories.

Hypotheses

$$H_0 : \beta_{\text{length}} = 0,$$
$$H_1 : \beta_{\text{length}} \neq 0.$$

Here, β_{length} denotes the regression coefficient associated with paper length. H_0 indicates that paper length has no effect on IntrAgent's performance, while H_1 tests whether paper length has a statistically significant effect.

Interpretation The model estimates a slope coefficient of -0.1444 for paper length, with $p = 0.463 > 0.05$ and a 95% confidence interval of $(-0.535,\ 0.246)$. Although the estimated coefficient is negative, its magnitude is small, and the p-value is well above the standard significance threshold of $\alpha = 0.05$. Therefore, we fail to reject H_0, indicating that paper length is not significantly associated with IntrAgent's performance in the current evaluation.

B.3 Execution Time Report

In this section, we report the computational runtime required to complete a single test instance on average. The runtime of IntrAgent is influenced by several factors, including the number of reasoning iterations until information sufficiency is achieved, API connection stability, and the response speed of the backbone LLM. Nevertheless, we provide an approximate runtime in seconds, averaged over multiple runs. (See Table19)

Since the number of runs can vary across experiments, we place greater emphasis on the median runtime on the default setting to provide a more representative estimate. IntrAgent achieves a median runtime of approximately 129 seconds, or about two minutes per paper per question. In comparison, the baseline methods require slightly less than one minute.

Given the nature of the IntraView task, our primary concern is accuracy rather than speed. Nevertheless, the agent remains significantly faster than human-level reading and reasoning over multiple full-text papers.

Method	Step	Time estimation (sec)
Evaluation protocol	MCQ mapping	5
IntrAgent	Format conversion	36
	Section Ranking stage	4
	Iterative Reading stage (1 iter)	42
	Median (Default setting)	129
	Average (Default setting)	259
Vanilla RAG	Plaintext conversion	2
	Short answer generation	35
DRAGIN	Plaintext conversion	2
	Paragraph-Level Corpus Construction	4
	Chain-of-Thought Construction(each one)	23
	Step-by-step Reasoning	55
R^2AG	Plaintext conversion	2
	Retriever Embedding	2
	Ranking	5
	Reranking	10
	Answer Generation with Retrieval Guidance	30
Lumos	Plaintext conversion	2
	Planning for a single Subgoal	6
	Converting Subgoal into a Retrieval Action	2
	Execution	2

Table 19: Estimated runtime (in seconds) for IntrAgent and baselines. Running with GPT-4o. Values are averaged over multiple runs per test instance.

C Ablation Studies Detail

C.1 Failure Case Study

C.1.1 Hallucination Failure Case from Missing Information Sufficiency Check

As an extension of our ablation study, we examine a specific failure mode caused by removing the information sufficiency check from the Iterative Reading stage. This case highlights the critical role of sufficiency verification in preventing answer hallucination.

We analyze a representative question from the *SERS* dataset based on the paper by Rathnayake et al. (Rathnayake et al., 2024): The correct section is successfully retrieved by section ranking stag: *Experimental - Surface entrapping of AGP and SERS measurements*. The following detail is extracted:

> "The measurements were performed on a ProRaman-L spectrometer with **785 nm excitation wavelength** using a $40\times$ objective, under the following conditions: power density of approximately $40\ \mu\mathrm{W}/\mu\mathrm{m}^2$, accumulation time of $40\ \mathrm{s}$, averaging over 3 spectra, and a spectral range of 400–3000 cm^{-1}."

This evidence directly supports the correct answer: **785 nm**.

In contrast, when the information sufficiency check is disabled, the agent outputs a fully hallucinated answer:

> "The excitation laser wavelength used for SERS measurements is confirmed to be **532 nm**. This is supported by the direct statement: 'The excitation wavelength for the SERS measurements is 532 nm' found in the text (Chunk 3, Section_title: Methodology)."

In this case, the cited sentence, section title, and numerical value are entirely fabricated and do not exist in the original document. The agent not only provides an incorrect answer but does so with unjustified confidence and fabricated attribution.

This failure mode demonstrates how critical the information sufficiency check is for ensuring factual grounding. Its absence increases the risk of hallucination, where the system generates plausible-sounding yet entirely unsupported responses.

C.1.2 Contextual Relationship Blindness from Domain-Specific Knowledge Gaps

Another failure pattern not resolved simply by using a more advanced LLM is **Contextual Relationship Blindness**. We believe this is partly due to gaps in domain knowledge: during the *Section*

Detail Extraction phase, the model fails to establish the correct relationship between the question and the corresponding sections in a given iteration. As a result, it may overlook crucial information, leading to deficiencies during the information sufficiency check stage and causing the system to loop to later sections, ultimately selecting "None of the Above."

For example, in our Earth Science dataset (specifically, the subfield of urban remote sensing), the main study targets are urban landscapes or buildings. Consider question #8 from the batch:

> Are auxiliary features used beyond raw spectral bands?

In remote sensing, it is well understood that the main features are spectral bands and radar bands. Auxiliary features are additional derived features, such as vegetation indices (NDVI, LAI, FAPAR), land surface temperature, albedo, emissivity, topographic information (e.g., DEM), or other external variables. However, when the LLM does not pick up the meaning or understand the term, even if the paragraph includes terms like NDVI, the relationship may still be overlooked. Cases like these highlighting `IntraBench` remain challenging. To improve the usability of the dataset for researchers who are not familiar with cross-domain terminology, we have annotated these terms in the benchmark; however, these annotations are **not** provided as inputs to the agent.

D Robustness of `IntrAgent` to Mapping Model and Input Variability Detail

D.1 Section Heading Rework

In Section 6.4, we investigated how `IntrAgent` performs when provided with subpar or non-standard section headings in scientific literature. Specifically, we examine whether the agent can maintain accuracy even when the section structure is rewritten in unfamiliar or distorted styles.

To conduct this study, we extracted all section headings from the paper by Thrift et al. (Thrift and Ragan, 2019), listed in the first column of Table 20. We then created three alternate versions of these headings: a simplified "Beginner Style", a distorted "Highly Noisy Version", and a stylized "Shakespearean Style", each shown in the subsequent columns.

For each variant, we replaced the original section titles in the document with the newly generated versions. `IntrAgent` was then applied to the modified texts using GPT-4o as the backbone LLM and GPT-4.1 as the mapping model, under default settings.

Original Title	Beginner Style	Highly Noisy Version	Shakespearean Style
A R T I C L E I N F O	Stuff About This Paper	ZXCV ARTICLE BLAH	Prologue of This Learned Work
A B S T R A C T	What This Thing's About	QWERT ABSTRACT THING	Abstract of Most Worthy Endeavor
1. Introduction	1. Starting Out	1. asdjkh Introduction blah	1. An Overture to the Matter at Hand
2. Materials and methods	2. Stuff We Used and Did	2. jkjhas Materials and doings	2. Of Materials Gathered and Deeds Undertaken
2.1. Bacteria and sample preparation	2.1. Germs and How We Got Them	2.1. ajskdfh Bacteria things and getting stuff	2.1. Of Bacterium's Harvest and Preparations Befitting Study
2.2. AFM imaging	2.2. Taking Tiny Pics	2.2. kjashdf AFM looking-at-things	2.2. Wherein the Art of Atomic Force Is Employed to Gaze Upon the Minuscule
2.3. SERS measurements	2.3. Laser Stuff on the Germs	2.3. asdkljfh SERS zappy laser stuff	2.3. Of Golden Gleams and SERS—The Spectral Oracle
2.4. Machine learning model development	2.4. Computer Things We Tried	2.4. zmxnv Machine learning magic box	2.4. Wherein the Machine, Apt in Learning, Is Taught to Discern
3. Results	3. What Happened	3. kjsdhf Results what happened	3. Revelations and Findings Most Curious
3.1. AFM analysis of microbial shape and dimension	3.1. Shapes of the Germs	3.1. qwekjh Germ shapes and sizes	3.1. Of Form and Measure—The Shape of the Microscopic Realm
3.2. SERS signature of periodontal pathogens	3.2. Shiny Lights from Germs	3.2. kjasdh SERS noise from germs	3.2. Upon the Light's Whisper: Signatures of the Mouth's Hidden Foes
3.3. ML-enabled identification of periodontal pathogens	3.3. Using Computer to Guess Germs	3.3. hgjdksh Computer guessing germ names	3.3. The Thinking Engine's Triumph in Unmasking the Pathogen
4. Discussion	4. What We Think It Means	4. lkjdfh What we think (maybe?)	4. A Discourse Upon the Meaning of These Happenings
5. Conclusion	5. Wrapping It Up	5. THE END (idk lol)	5. A Summation, As the Curtain Draws Near
CRediT authorship contribution statement	Who Did What	blorpflorp CRediT who did what maybe	Of Quills and Labors: A Testament to Those Who Contributed
Declaration of competing interest	We Don't Got Any Fights About This	nope-no-fightz Declaration or whatever	Of Conflicts, None Declared nor Concealed
Data availability	Where the Stuff Is	canhasdata Data place??	Where Might the Curious Find the Data?
Acknowledgement	Thanks and Shoutouts	thxbye Acknowlegmints	Gratitude, in Verse and Spirit
Appendix A. Supplementary data	Extra Stuff We Didn't Put Up There	Appendix A. More junk we had	Appendix the First: Scrolls of Supplement Yet Untold
References	Books and Papers We Looked At	Books n Stuff We Readed (aka refs)	Tomes and Records Consulted—A Catalog of the Wise

Table 20: Rewriting original section headings from Thrift et al. (Thrift and Ragan, 2019) into three alternative styles—Beginner Style, Highly Noisy Version, and Shakespearean Style—to evaluate the robustness of `IntrAgent` to variation in document structure.

D.2 Section Heading Rework - Highly Noisy Version Reasoning Example

This is a detailed reasoning example from the SERS dataset Thrift et al. (Thrift and Ragan, 2019), using the second section title rewrite variation, labeled as the Highly Noisy Version. The rewritten section headings are provided in Table 20.

With paper main title provided alongside with the research question for this case is: What are the material and structural properties, or the morphology, of the SERS substrates used?

After the Hierarchy Preservation step, the processed section headings are as follows:

> ZZXCV ARTICLE BLAH; QWERT ABSTRACT THING; asdjkh Introduction blah; 2. jkjhas Materials and doings - 2.1. ajskdfh Bacteria things and getting stuff; 2. jkjhas Materials and doings - 2.2. kjashdf AFM looking-at-things; 2. jkjhas Materials and doings - 2.3. asdkljfh SERS zappy laser stuff; 2. jkjhas Materials and doings - 2.4. zmxnv Machine learning magic box; 3. R kjsdhf Results what happened - 3.1. qwekjh Germ shapes and sizes; 3. R kjsdhf Results what happened - 3.2. kjasdh SERS noise from germs; 3. R kjsdhf Results what happened - 3.3. Mhgjdksh Computer guessing germ names; 4. lkjdfh What we think (maybe?); 5. THE END (idk lol); blorpflorp CRediT who did what maybe; nope-no-fightz Declaration or whatever; canhasdata Data place??; thxbye Acknowlegmints; more junk we had; Books n Stuff We Readed (aka refs)

In semantic section ranking, we derived the reasoning step:

> 1. 2. jkjhas Materials and doings - 2.3. asdkljfh SERS zappy laser stuff — This section specifically mentions "SERS" in its title, indicating it likely contains detailed information about the SERS substrates, including their materials and structure or morphology.
>
> 2. 2. jkjhas Materials and doings - 2.1. ajskdfh Bacteria things and getting stuff — As part of the "Materials and doings" main section, this subsection may discuss materials related to the experimental setup, possibly including the SERS substrates.
>
> 3. 2. jkjhas Materials and doings - 2.2. kjashdf AFM looking-at-things — This subsection likely involves materials characterization using AFM, which may include details about substrate morphology or materials.
>
> 4. asdjkh Introduction blah — The introduction often contains a summary of the materials and methods used, potentially offering an overview of the SERS substrate materials and structure.
>
> ...

This ultimately leads to correct section allocation, enabling detailed information gathering and passing the information sufficiency check without difficulty. It also results in accurate final mapping to the correct multiple-choice answer. This example highlights the model's ability to extract relevant information from scrambled headings by reasoning through their underlying intent, even in the presence of noise during section ranking.

E Prompts

Here is the list of prompts we used in our study.

E.1 Section Ranking Prompts

E.1.1 Hierarchy Preservation Prompt

Prompt: Hierarchy Preservation

```
hierarchy_preservation_prompt = """
You are an expert in scientific research
↪  and academic writing, proficient in
↪  analyzing the structural organization
↪  of research papers. Your task is to
↪  infer the hierarchical structure of a
↪  research paper based on a list of
↪  section titles provided in sequential
↪  order.

---

### **Task**
You are given a list of section titles
↪  extracted from a research paper, shown
↪  in the order they appear in the
↪  document:

{headings}

These titles are not annotated to indicate
↪  their level (e.g., section,
↪  subsection, sub-subsection). Your job
↪  is to infer the **tree structure** by
↪  determining which titles are:
- **Main sections**
```

```
- **Subsections** under a main section
- **Sub-subsections** under a subsection

---

### **Rules**
- Maintain the **original order** of the
↪  titles.
- Do **not** introduce any new titles or
↪  remove existing ones.
- Use the **first title** as the likely
↪  **main title** of the paper, unless
↪  it's clearly a preamble (e.g.,
↪  "Abstract", "Introduction").
- Infer nesting levels based on typical
↪  academic paper structure and semantic
↪  cues in the titles.
- Structure the output using this format:
  - `"Section Title"` (main section)
  - `"Section Title - Subsection Title"`
  ↪  (subsection)
  - `"Section Title - Subsection Title -
  ↪  Sub-subsection Title"`
  ↪  (sub-subsection)

---

### **Example**

Given:

["Experiment setup", "sample preparation",
↪  "device setup", "SERS measurements",
↪  "Result and Discussion"]

Expected output:

["Experiment setup",
 "Experiment setup - sample preparation",
 "Experiment setup - device setup",
 "Experiment setup - SERS measurements",
 "Result and Discussion"]

---

### **Expected Output Format**
- Provide the inferred tree structure in a
↪  Python list format.
- Follow the form: `["Section", "Section -
↪  Subsection", "Section - Subsection -
↪  Sub-subsection", ...]`
- Output **only the list**. Do not include
↪  explanations, reasoning steps, or
↪  extra comments.

---

Let's think step by step.
"""
```

E.1.2 Reasoning-Based Ranking Prompt

Prompt: Reasoning-Based Ranking

```
ranking_prompt = """

You are an AI research assistant with
↪  expertise in analyzing academic papers.
↪  Your task is to determine which
↪  sections of a paper are most likely to
↪  contain the answer to a given research
↪  question. The section titles are
↪  provided in a **tree structure**,
↪  where main sections and their
↪  subsections are denoted using a hyphen
↪  ("-"). Your goal is to rank these
↪  sections and subsections from **most
↪  relevant to least relevant** in
↪  relation to the research question.

---

**Paper Title**: {main_title_heading}

**Research Question**: {question}

**Section Titles**:
{formatted_headings}

---

### **Instructions**
- Analyze the research question and the
↪  structure of the section titles.
- Rank the section titles in order of
↪  likelihood to contain the answer —
↪  from most to least relevant.
- Subsections may be more specific than
↪  main sections; use their nesting to
↪  guide your ranking.
- For each section in the ranking, provide
↪  a **one-sentence explanation** for its
↪  placement.
- Maintain the original order of section
↪  titles when relevance is tied.
- Do **not** skip or omit any section
↪  titles from the list.
- Format the final ranking using only
↪  integer indices, enclosed in <<<#>>>
↪  brackets, based on the position of
↪  each title in the provided list
↪  (starting from 1).
- Do not repeat section titles or include
↪  any extra commentary after the final
↪  ranking.

---

### **Response Template**

Reasoning steps:

1. [Title 1] — [1-sentence justification]
2. [Title 2] — [1-sentence justification]
...

**Final ranking:** <<<2>>>, <<<1>>>,
↪  <<<3>>>, ... <<<n>>>

Let's think step by step.
"""
```

E.2 Iterative Reading Prompts

E.2.1 Action Loop Prompt

Prompt: Action Loop

```
main_prompt = """
You are an assistant designed to select
↪  the next Action based on the current
↪  observation.

Each observation contains:
- The current chunk index, indicating
↪  which chunk of the document you are
↪  working on.
- The total number of available chunks.
- A record of past Actions taken in
↪  previous iterations.
- If the most recent Action was EVALUATE,
↪  the observation will also include the
↪  evaluation result.

---

To ensure accuracy, you must follow these
↪  instructions:

{main_prompt_instructions}

---

Descriptions of all allowed actions:

- GET_CHUNK: Retrieve the next Knowledge
↪  Chunk. Use this when no chunk is
↪  currently loaded or when advancing to
↪  the next chunk.
- GET_DETAIL: Extract relevant details
↪  with respect to the Research Question
↪  from the currently loaded Knowledge
↪  Chunk.
- EVALUATE: Determine whether the
↪  extracted details from the current
↪  Knowledge Chunk are sufficient to
↪  answer the Research Question.
- TERMINATE: End the process once you have
↪  gathered enough relevant information
↪  to confidently answer the Research
↪  Question. (TERMINATE is only valid if
↪  the most recent evaluation yielded a
↪  “YES” result.)

---

Here are some examples:

- Past Action Taken: GET_CHUNK. I have
↪  already obtained the chunk, so I will
↪  GET_DETAIL.
- Past Action Taken: GET_CHUNK, GET_DETAIL.
↪  then I will now EVALUATE the summary.
- Past Action Taken: GET_CHUNK,
↪  GET_DETAIL, EVALUATE. The result does
↪  not seem to be going well, so I will
↪  GET_CHUNK again.
- Past Action Taken: GET_CHUNK, GET_DETAIL,
↪  EVALUATE. The result seems to be going
↪  well, so I will TERMINATE the process.
- Past Action Taken: GET_CHUNK, GET_DETAIL,
↪  EVALUATE, ... ,GET_CHUNK, GET_DETAIL,
↪  EVALUATE. This is chunk 10 out of 10.
↪  I was supposed to get more chunks, but
↪  I have already retrieved the last one.
↪  No more knowledge chunks are available,
↪  so I will TERMINATE the process.

---

Observation:

You are currently at chunk
↪  {current_chunk_index} out of
↪  {total_chunks_len} chunks.

Past actions taken:
{past_action}

{evaluation_result}

---

Response Format:

Your final answer must follow the exact
↪  format below. Do not include any text
↪  outside this format.

Format:

Reasoning Steps: [In one sentence, explain
↪  why this Action is the best next step.]

Action: [Choose one of the following:
↪  GET_CHUNK, GET_DETAIL, EVALUATE,
↪  TERMINATE]
"""
```

E.2.2 Action Loop - Default Balanced Instruction

For the instruction given in the loop above, depend on confidence level, different prompt was provided. This given Instruction subject to confidence level 2. This instruction is being used for default setting.

Prompt: Action Loop

```
main_prompt_instructions_confidence_level_2
↪  = """
1. You will be provided with a list of
↪  Knowledge Chunks, ordered by relevance
↪  to the research question. Earlier
↪  Knowledge Chunks are more likely to
↪  contain useful information.
2. Your task is to iteratively retrieve a
↪  Knowledge Chunk, extract relevant
↪  details, and evaluate whether you can
↪  answer the research question. Repeat
↪  this process until you can confidently
↪  terminate.
3. You will be given a list of predefined
↪  actions to select from.
4. In each iteration, you must select
↪  exactly one action based on the
↪  current observation.
```

```
5. The observation includes: the current
↪  Knowledge Chunk index, the total
↪  number of available Knowledge Chunks,
↪  and the list of past actions taken.
6. The available actions are: GET_CHUNK,
↪  GET_DETAIL, EVALUATE, and TERMINATE,
↪  as defined in the action list.
7. If the most recent action in the past
↪  actions taken was GET_CHUNK, your next
↪  action **must** be GET_DETAIL to
↪  extract information from the current
↪  Knowledge Chunk — do not perform this
↪  extraction yourself.
8. If the most recent action was
↪  GET_DETAIL, your next action **must**
↪  be EVALUATE to assess whether the
↪  gathered details are sufficient to
↪  answer the research question — do not
↪  perform the evaluation yourself.
9. You must select only one action at a
↪  time — never choose multiple actions
↪  in a single step.
10. Since Knowledge Chunks are ordered by
↪  relevance to the research question,
↪  earlier Knowledge Chunks are more
↪  likely to contain useful information.
↪  Use this ordering to guide your
↪  reading sequence.
11. You must TERMINATE the process once
↪  you have gathered enough information
↪  to confidently answer the research
↪  question.
12. Pay attention to the total number of
↪  Knowledge Chunks. If your most recent
↪  EVALUATE action was performed on the
↪  **last available Knowledge Chunk** and
↪  the result was insufficient, you must
↪  TERMINATE the process due to lack of
↪  additional information.
13. The expected workflow is: GET_CHUNK →
↪  GET_DETAIL → EVALUATE → TERMINATE (if
↪  evaluation result is sufficient). If
↪  not, continue to the next Knowledge
↪  Chunk. If you reach the final
↪  Knowledge Chunk and the evaluation is
↪  still insufficient, you must
↪  TERMINATE.
14. You may TERMINATE early if an EVALUATE
↪  action confirms that the gathered
↪  information is sufficient to answer
↪  the research question with high
↪  confidence.
15. After your reasoning, output your
↪  response in the specified format.
"""
```

E.2.3 Section Detail Extraction Prompt

Prompt: Action: Section Detail Extraction

```
get_detail_prompt = """
You are a research assistant helping
↪  extract detailed information relevant
↪  to the given Research Question based
↪  on a Knowledge Chunk given.

---

Research Question: {question}

Knowledge Chunk:
{chunk}

---

Task:
- Extract all key points from the current
↪  Knowledge Chunk **only as they relate
↪  to the Research Question**.
- Include all relevant information such as
↪  scientific terms, numerical data,
↪  experimental results, measurements,
↪  statistical indicators, conclusions,
↪  and any comparative or causal
↪  statements that explicitly support or
↪  address the Research Question.
- Do not infer or assume missing content.
↪  If essential information is absent,
↪  clearly state what is missing—avoid
↪  speculation or completion.
- When a sentence directly answers or
↪  supports the Research Question, quote
↪  it **verbatim** from the **original
↪  chunk**.
- If there are multiple key points,
↪  present them as a structured list in
↪  bullet point format.
- If there are no key points relevant to
↪  the Research Question, clearly state:
↪  *"This chunk does not provide relevant
↪  information."*
- Perform this task step by step, ensuring
↪  that each extracted detail is
↪  justified and directly traceable to
↪  the original content.

---

Response Format:

Your final answer must follow the exact
↪  format below. Ensure strict adherence
↪  to this structure. Do not include
↪  additional explanations outside of
↪  this format.

Format:
Reasoning Steps: [reasoning step by step
↪  goes here]
Details: The Section_title is
↪  <Section_title>.[Details based on the
↪  Research Question and chunk go here.
↪  Include all relevant findings and
↪  quote supporting sentences, present
↪  them as a structured list.]
"""
```

E.2.4 Default Balanced Information Sufficiency Check Prompt

The Information Sufficiency Check Prompt, confidence level 2. This prompt is being used for the default setting.

Prompt: Sufficiency Check Confidence Level 2

```
evaluation_prompt_confidence_level_2 = """
You are a research assistant tasked with
↪  evaluating whether the provided
↪  details are both sufficient and
↪  accurate to answer a given research
↪  question. Based solely on the current
↪  details, determine whether they
↪  contain all the necessary and correct
↪  information required to answer the
↪  research question. Your evaluation
↪  must include direct references to
↪  original sentences from the provided
↪  details to support your reasoning.

---

Research Question: {question}

Current Details:
{observation_stage}

---

Task:
- Assess whether the current details
↪  contain all essential and accurate
↪  information needed to answer the
↪  Research Question.
- Respond with "YES" **only if** the
↪  provided details fully and correctly
↪  address the research question with no
↪  missing elements or uncertainties.
  - In this case, provide a complete
  ↪  answer supported by quoted or
  ↪  clearly referenced content from the
  ↪  current Knowledge Chunk.
- If the information is incomplete,
↪  partially correct, or uncertain in any
↪  way, respond with "NO."
  - Clearly identify what specific
  ↪  information is missing or ambiguous.
  - Then provide the closest possible
  ↪  answer using only the available
  ↪  details.
  - If no relevant information is present
  ↪  at all, write: *"No information in
  ↪  given details."*

- After reasoning step by step, output
↪  both `Sufficiency` and
↪  `Detail_Answer`. Your `Detail_Answer`
↪  must include direct evidence from the
↪  knowledge chunk — avoid general
↪  summaries.

---

Response Format:

Your response must strictly follow the
↪  format below:

Sufficiency: [YES or NO]

Detail_Answer: The Section_title is
↪  <Section_title>. [Provide your best
↪  possible answer to the research
↪  question, using direct references from
↪  the details. Explain your reasoning
↪  step by step, ensuring each claim is
↪  supported by the provided content.]
"""
```

E.2.5 Detail Aggregation Final Answer Prompt

Prompt: Summary for Final Answer

```
full_set_answer_prompt = """
You are a research assistant tasked with
↪  synthesizing a final answer to the
↪  research question based on the
↪  evaluations of multiple knowledge
↪  chunks provided below. Each evaluation
↪  entry includes:
- A sufficiency judgment (YES or NO),
- A detailed answer (Detail_Answer),
- Referenced original sentences, each
↪  tagged with its Chunk number and
↪  Section title.

Use the complete set of evaluation entries
↪  to construct a coherent,
↪  well-supported, and evidence-based
↪  final answer to the research question.
↪  You must include direct references to
↪  the **original sentences**, along with
↪  their corresponding **Chunk number**
↪  and **Section title**, to justify each
↪  claim you make.

---

Research Question: {question}

Evaluation Entries:
{observation_stage}

---

Task:
- Synthesize the provided evaluation
↪  entries to generate a comprehensive
↪  and conclusive answer to the Research
↪  Question.
- Your answer must be fully supported by
↪  specific content from the evaluation
↪  entries.
- When making a claim, **explicitly cite**
↪  the source using the format: `"Quoted
↪  sentence from the original text"
↪  (Chunk #, Section_title:
↪  <Section_title>)`.
- If multiple entries support the same
↪  point, cite each one.
```

```
- Do **not** introduce new information or
↪  inferred content that is not present
↪  in the evaluation entries.
- Avoid vague generalizations — every
↪  component of the answer must be
↪  evidence-backed.
- After reasoning step by step, output
↪  `Final_Answer` using the exact format
↪  below.

---

Response Format:

Final_Answer: [Provide your final answer
↪  with supporting evidence from the
↪  evaluation entries. For every claim,
↪  cite the original sentences along with
↪  the Chunk number and Section title
↪  they came from. Explain your reasoning
↪  step by step, covering all required
↪  aspects of the Research Question.]
"""
```

E.3 Evaluation Protocol Prompts

E.3.1 Evaluation Protocol - Baseline Final Answer Prompt

Prompt: Final Answer - Choices Mapping

```
full_set_answer_prompt = """
You are a research assistant tasked with
↪  synthesizing a final answer to the
↪  research question based on the
↪  evaluations of multiple knowledge
↪  chunks provided below. Each evaluation
↪  entry includes:
- A sufficiency judgment (YES or NO),
- A detailed answer (Detail_Answer),
- Referenced original sentences, each
↪  tagged with its Chunk number and
↪  Section title.

Use the complete set of evaluation entries
↪  to construct a coherent,
↪  well-supported, and evidence-based
↪  final answer to the research question.
↪  You must include direct references to
↪  the **original sentences**, along with
↪  their corresponding **Chunk number**
↪  and **Section title**, to justify each
↪  claim you make.

---

Research Question: {question}

Evaluation Entries:
{observation_stage}

---

Task:
- Synthesize the provided evaluation
↪  entries to generate a comprehensive
↪  and conclusive answer to the Research
↪  Question.
- Your answer must be fully supported by
↪  specific content from the evaluation
↪  entries.
- When making a claim, **explicitly cite**
↪  the source using the format: `"Quoted
↪  sentence from the original text"
↪  (Chunk #, Section_title:
↪  <Section_title>)`.
- If multiple entries support the same
↪  point, cite each one.
- Do **not** introduce new information or
↪  inferred content that is not present
↪  in the evaluation entries.
- Avoid vague generalizations — every
↪  component of the answer must be
↪  evidence-backed.
- After reasoning step by step, output
↪  `Final_Answer` using the exact format
↪  below.

---

Response Format:

Final_Answer: [Provide your final answer
↪  with supporting evidence from the
↪  evaluation entries. For every claim,
↪  cite the original sentences along with
↪  the Chunk number and Section title
↪  they came from. Explain your reasoning
↪  step by step, covering all required
↪  aspects of the Research Question.]
"""
```

E.3.2 Evaluation Protocol - Final Answer to MCQ Answer Mapping

Prompt: Final Answer - Final Answer to MCQ Answer Mapping

```
prompt = f"""
You are a research assistant. Your task is
↪  to map a short answer to a research
↪  question, derived from a scientific
↪  research paper, to the most
↪  appropriate option among the provided
↪  answer choices in a multiple-choice
↪  question (MCQ) format.
We are given the following research
↪  question:
{question}

To support your understanding of the
↪  research question, here is some
↪  relevant contextual information:
{additional_note}

Here are the extracted answers to the
↪  research question, taken from the
↪  research paper:
{relevant_chunks}

Here are the answer choices:
{paper_choice}

**Instructions:**
```

```
1. You must select **only one answer
↪  choice** from the list that is either
↪  correct or most relevant to the
↪  research question.
2. Provide your answer as a single letter
↪  (e.g., A, B, C, D, E, F) enclosed in
↪  the format <<<ANS>>>. Do not include
↪  any other text in your final output.
3. You may include step-by-step reasoning
↪  to justify your decision, but your
↪  final output must consist of only one
↪  answer choice in the required format.
4. This is a science and
↪  technology-related research question.
↪  For cases where the short answer
↪  involves numerical values:
    - You must map the short answer to the
    ↪  most accurate and relevant answer
    ↪  choice.
    - If the short answer and an answer
    ↪  choice represent numerical values,
    ↪  apply the following decimal
    ↪  precision alignment rules:
        - If the short answer and an answer
        ↪  choice have the **same number
        ↪  of decimal places**, compare
        ↪  them directly without
        ↪  rounding.
        - If they have **different numbers
        ↪  of decimal places**, round the
        ↪  value with **more digits** to
        ↪  match the **shorter decimal
        ↪  precision**, then compare.
        - After alignment, select the
        ↪  answer choice that **exactly
        ↪  matches** the short answer.
        - If no answer choice matches after
        ↪  applying these rules, select
        ↪  **"F. None of the above."**
    - **Examples**:
        - Short answer = 1.1111, answer
        ↪  choices include 1.1 → round
        ↪  short answer to 1.1 → match →
        ↪  select 1.1
        - Short answer = 2.01, answer
        ↪  choices include 1.98 → both
        ↪  have two decimal places → no
        ↪  rounding → no match → select
        ↪  "F. None of the above"
5. Never provide multiple answers.
**Example of output format:**
*reasoning steps*
---
<<<C>>>
Let's think step by step.
"""
```

E.4 Confidence Level Ablation Study Prompts

E.4.1 Confidence Level 1 Ablation Study - Conservative Information Sufficiency Check Prompt

Prompt: Sufficiency Check Confidence Level 1

```
evaluation_prompt_confidence_level_1 = """
You are a research assistant tasked with
↪  evaluating whether the provided
↪  details are both sufficient and
↪  accurate to answer a given research
↪  question. Based solely on the current
↪  details, determine whether they
↪  contain all the necessary and correct
↪  information required to answer the
↪  research question. You must act
↪  conservatively — your evaluation
↪  should follow the strictest standard:
↪  any ambiguity, missing element, or
↪  lack of clarity must result in a "NO".

---

Research Question: {question}

Current Details:
{observation_stage}

---

Task:
- Assess whether the extracted details
↪  contain all **necessary elements** to
↪  fully and unambiguously answer the
↪  Research Question.
- Only respond with "YES" if **all
↪  aspects** of the research question are
↪  addressed with precise and complete
↪  evidence from the provided knowledge
↪  chunk.
  - No assumptions, inferred logic, or
  ↪  external knowledge are allowed.
- If **any part** of the required
↪  information is missing, vague,
↪  incomplete, or unclear, respond with
↪  "NO".
  - Clearly identify what is missing or
  ↪  uncertain.
  - Then provide the closest possible
  ↪  answer that can be supported solely
  ↪  by the available content from the
  ↪  Knowledge Chunk.
  - If no relevant information is present
  ↪  at all, state: *"No information in
  ↪  given details."*

- After step-by-step reasoning, output
↪  both `Sufficiency` and
↪  `Detail_Answer`. Your `Detail_Answer`
↪  must include direct evidence from the
↪  knowledge chunk — avoid general
↪  summaries.

---

Response Format:
```

```
Your response must strictly follow the
↪  format below:

Sufficiency: [YES or NO]

Detail_Answer: The Section_title is
↪  <Section_title>. [Provide your best
↪  possible answer to the research
↪  question, using direct references from
↪  the details. Explain your reasoning
↪  step by step, addressing all required
↪  components.]
"""
```

E.4.2 Confidence Level 3 Ablation Study - Aggressive Information Sufficiency Check Prompt

Prompt: Sufficiency Check Confidence Level 3

```
evaluation_prompt_confidence_level_3 = """
You are a research assistant tasked with
↪  evaluating whether the provided
↪  details are both sufficient and
↪  accurate to answer a given research
↪  question. You may act more
↪  aggressively and confidently. Based
↪  solely on the current details,
↪  determine whether they contain
↪  **enough relevant content to
↪  reasonably answer** the research
↪  question, even if some minor points
↪  are not fully explicit.

---

Research Question: {question}

Current Details:
{observation_stage}

---

Task:
- Assess whether the current details
↪  provide **most or all of the key
↪  information** needed to reasonably
↪  answer the Research Question.
- Respond with "YES" if the answer can be
↪  supported using the provided
↪  information, even if a few supporting
↪  details are missing, implicit, or only
↪  partially clear.
  - If confident, provide a full answer
  ↪  backed by the best available
  ↪  references from the Knowledge Chunk.
- Only respond with "NO" if **critical**
↪  information is missing or the answer
↪  would be too speculative.
  - Clearly explain what is missing or
  ↪  unclear.
  - Then provide the closest possible
  ↪  answer based only on the available
  ↪  content.
  - If no relevant information is present
  ↪  at all, state: *"No information in
  ↪  given details."*
- After step-by-step reasoning, output
↪  both `Sufficiency` and
↪  `Detail_Answer`. Your `Detail_Answer`
↪  must include direct evidence from the
↪  knowledge chunk — avoid general
↪  summaries.

---

Response Format:

Your response must strictly follow the
↪  format below:

Sufficiency: [YES or NO]

Detail_Answer: The Section_title is
↪  <Section_title>. [Provide your best
↪  possible answer to the research
↪  question, using direct references from
↪  the details. Explain your reasoning
↪  step by step, addressing all required
↪  components.]
"""
```

E.4.3 Confidence Level 1 Ablation Study - Action Loop Conservative Instruction

Prompt: Confidence Level Case Study Prompts 1

```
main_prompt_instructions_confidence_level_1
↪  = """
1. You will be provided with a list of
↪  Knowledge Chunks, ordered by relevance
↪  to the research question. Earlier
↪  Knowledge Chunks are more likely to
↪  contain useful information.
2. Your task is to iteratively retrieve a
↪  Knowledge Chunk, extract relevant
↪  details, and evaluate whether you can
↪  answer the research question. Repeat
↪  this process until you have evaluated
↪  all available Knowledge Chunks or
↪  gathered enough information to
↪  confidently terminate.
3. You will be given a list of predefined
↪  actions to select from.
4. In each iteration, you must select
↪  exactly one action based on the
↪  current observation.
5. The observation includes: the current
↪  Knowledge Chunk index, the total
↪  number of available Knowledge Chunks,
↪  and the list of past actions taken.
6. The available actions are: GET_CHUNK,
↪  GET_DETAIL, EVALUATE, and TERMINATE,
↪  as defined in the action list.
7. If the most recent action in the past
↪  actions taken was GET_CHUNK, your next
↪  action **must** be GET_DETAIL to
↪  extract information from the current
↪  Knowledge Chunk — do not perform this
↪  extraction yourself.
```

```
8. If the most recent action was
↪  GET_DETAIL, your next action **must**
↪  be EVALUATE to assess whether the
↪  gathered details are sufficient to
↪  answer the research question — do not
↪  perform the evaluation yourself.
9. You must select only one action at a
↪  time — never choose multiple actions
↪  in a single step.
10. Since Knowledge Chunks are ordered by
↪  relevance to the research question,
↪  earlier Knowledge Chunks are more
↪  likely to contain useful information.
↪  Use this ordering to guide your
↪  reading sequence.
11. You must TERMINATE the process only
↪  after either (a) all available
↪  Knowledge Chunks have been evaluated,
↪  or (b) you are confident that the
↪  research question can be answered
↪  based on the most recent evaluation.
12. Pay attention to the total number of
↪  Knowledge Chunks. If your most recent
↪  EVALUATE action was performed on the
↪  **last available Knowledge Chunk**,
↪  you must TERMINATE the process
↪  regardless of the outcome.
13. The expected workflow is: GET_CHUNK →
↪  GET_DETAIL → EVALUATE → TERMINATE (if
↪  evaluation result is sufficient). If
↪  not, continue to the next chunk. If
↪  you reach the final chunk and the
↪  evaluation is still insufficient, you
↪  must TERMINATE.
14. Do not TERMINATE early based solely on
↪  assumptions about relevance. Always
↪  continue until a conclusive evaluation
↪  result is available or all chunks are
↪  exhausted.
15. After your reasoning, output your
↪  response in the specified format.
"""
```

E.4.4 Confidence Level 3 Ablation Study - Action Loop Aggressive Instruction

Prompt: Confidence Level Case Study Prompts 3

```
main_prompt_instructions_confidence_level_3
↪  = """
1. You will be provided with a list of
↪  knowledge chunks, ordered by relevance
↪  to the research question. Earlier
↪  chunks are more likely to contain
↪  useful information.
2. Your task is to iteratively retrieve a
↪  knowledge chunk, extract relevant
↪  details, and evaluate whether you can
↪  answer the research question. Repeat
↪  this process until you can confidently
↪  terminate.
3. You will be given a list of predefined
↪  actions to select from.
4. In each iteration, you must select
↪  exactly one action based on the
↪  current observation.
5. The observation includes: the current
↪  chunk index, the total number of
↪  available chunks, and the list of past
↪  actions taken.
6. The available actions are: GET_CHUNK,
↪  GET_DETAIL, EVALUATE, and TERMINATE,
↪  as defined in the action list.
7. If the most recent action in the past
↪  actions taken was GET_CHUNK, your next
↪  action **must** be GET_DETAIL to
↪  extract information from the current
↪  knowledge chunk — do not perform this
↪  extraction yourself.
8. If the most recent action was
↪  GET_DETAIL, your next action **must**
↪  be EVALUATE to assess whether the
↪  gathered details are sufficient to
↪  answer the research question — do not
↪  perform the evaluation yourself.
9. You must select only one action at a
↪  time — never choose multiple actions
↪  in a single step.
10. Since knowledge chunks are ordered by
↪  relevance to the research question,
↪  earlier chunks are more likely to
↪  contain useful information. Use this
↪  ordering to guide your reading
↪  sequence.
11. You may TERMINATE the process as soon
↪  as you believe that additional
↪  knowledge chunks are unlikely to
↪  provide significantly better
↪  information, even if you are not fully
↪  confident.
12. If your most recent EVALUATE action
↪  was performed on the **last available
↪  chunk**, you must TERMINATE the
↪  process regardless of the outcome.
13. The expected workflow is: GET_CHUNK →
↪  GET_DETAIL → EVALUATE → TERMINATE. You
↪  may also TERMINATE early based on the
↪  assumption that remaining chunks have
↪  diminishing relevance.
14. Terminate aggressively if your
↪  evaluation suggests NO from
↪  continuing, even if high confidence
↪  has not yet been achieved.
15. After your reasoning, output your
↪  response in the specified format.
"""
```

E.5 Contextual RAG Prompt

Prompt: Contextual RAG Prompt

```
prompt = """
You are an expert technical writer
↪  specialised in contextual retrieval.

<document>
{whole_document}
</document>

Here is the chunk we want to situate
↪  within the whole document
<chunk>
{chunk}
</chunk>
```

```
Please give a short succinct context to
↪  situate this chunk within
the overall document for the purposes of
↪  improving search retrieval
of the chunk. Answer only with the
↪  succinct context and nothing else.
"""
```

F Baseline Evaluation Details

F.1 Vanilla RAG

Vanilla RAG refers to the original implementation proposed by Lewis et al. (Lewis et al., 2020). In our setup, we first convert the PDF versions of papers into plain text. Subsequently, we tokenize the text into fixed-length chunks of 500 tokens with an overlap of 50 tokens between consecutive chunks. We select 500 tokens as it provides a balance between semantic coherence and memory efficiency, allowing enough context to be preserved within each chunk. The 50-token overlap ensures that important information occurring at chunk boundaries is not lost, improving retrieval continuity and robustness.

For embedding generation and similarity computation, we use the `all-MiniLM-L6-v2304` model to encode each chunk. During retrieval, we adopt a top-k strategy, defaulting to the top 3 chunks most similar to the given research question, based on cosine similarity scores. This top-k selection aligns with our evaluation protocol, where these top 3 chunks are passed into the generation module to produce short-form answers.

For final answer generation, we evaluate selected backbone LLMs under identical input conditions to ensure a fair comparison. Each model generates an answer conditioned on the retrieved top-3 chunks and the research question.

F.2 Contexual RAG

To enhance retrieval precision beyond conventional dense embedding approaches, we implemented a Contextual Retrieval-Augmented Generation (Contextual RAG) framework (Anthropic, 2024). Unlike vanilla RAG, which embeds document chunks in isolation, Contextual RAG explicitly augments each chunk with a succinct, model-generated context situating it within the broader document structure. This implementation effectively bridges semantic and structural understanding, allowing the retrieval component to capture both what is being discussed and where it fits contextually within the paper. This contextual enrichment helps disambiguate semantically similar fragments, improving the embedding model's ability to capture fine-grained relationships between queries and relevant content.

Our pipeline first extracts raw text given literature, followed by cleaning and token-level chunking with a 500-token window and 50-token overlap to preserve local continuity. For each chunk, a lightweight contextualization module built on GPT-4o generates concise summaries that describe how the chunk fits within the overall document; prompt used see E.5. These augmented chunks are then encoded using models—either E5-Mistral-7B-Instruct or GritLM-7B— for efficient batch inference. Query and chunk embeddings are computed with cosine similarity to identify the top-k most relevant text segments per question, enabling high-precision retrieval aligned with the document's narrative structure.

F.3 DRAGIN

We adopt *DRAGIN* (Dynamic Retrieval Augmented Generation based on the real-time Information Needs of LLMs)(Su et al., 2024b) as the baseline framework for dynamic retrieval-augmented generation. *DRAGIN* explicitly determines both *when to retrieve* and *what to retrieve* based on the model's internal uncertainty and token-level salience signals, enabling fine-grained control over evidence acquisition during generation.

To adapt *DRAGIN* to our customized scenario involving a small-scale paragraph-level corpus and chain-of-thought-style few-shot prompting, we introduce the following modifications:

Paragraph-Level Corpus Construction. We replace the default Wikipedia corpus (`psgs_w100.tsv`) used in the official *DRAGIN* release with a custom paragraph-level corpus (`DRAGIN_paragraphs.tsv`) tailored to our domain. Each entry consists of a unique paragraph ID, a document title, and the paragraph text. This design enables the retrieval module to operate at a finer granularity, which is essential for scenarios involving a small collection of five domain-specific documents.

Few-Shot Prompting with Summary-Driven CoT Chains. We generate ten chain-of-thought (CoT) reasoning exemplars from the five domain-specific documents. For each test instance, `IntrAgent` summarizes the top five retrieved para-

graphs to construct a condensed context representation. These summaries serve as inputs to guide CoT generation and are incorporated into the `demo` field during the `inference()` stage. For example:

> "The spatial extent of the study area is approximately 966 km², derived from the Residential category's area (398 km²) and its proportion (41.20%) of the total impervious surface..."

This procedure ensures that the few-shot prompt incorporates domain-relevant knowledge in a structured format, strengthening the alignment between the model's reasoning path and the underlying evidence.

Dynamic Retrieval at Inference Time. During inference, *DRAGIN* actively monitors for hallucination-prone segments and triggers retrieval of relevant evidence in real time. Retrieved paragraphs are incrementally integrated into the generation context, enabling iterative refinement of the model's output. Unlike short-form QA systems, *DRAGIN* produces complete chain-of-thought reasoning sequences, along with auxiliary metadata such as retrieval count and token usage for diagnostic analysis.

Following CoT generation, we initiate a final synthesis stage. The full reasoning trace, rather than the retrieved paragraphs themselves, is incorporated into a standardized prompt alongside the original question. This prompt is then passed to an external language model (e.g., GPT-4-turbo, GPT-4o, or DeepSeek-Chat), which is instructed to summarize the reasoning and produce a concise final answer. This decoupled architecture ensures interpretability during intermediate steps while maintaining fluency and coherence in the final output.

F.4 R^{2}AG

We adopt *R*^{2}*AG* (Reranking-augmented Retrieval-Augmented Generation) as one of our baseline(Ye et al., 2024). *R*^{2}*AG* enhances generation quality by incorporating a reranking module that explicitly guides evidence selection, thus mitigating retrieval noise and improving relevance alignment in the final output.

To implement the *R*^{2}*AG* pipeline in our setting, we follow the procedures below:

Retriever Fine-Tuning and Dense Indexing. We fine-tune a Sentence-BERT retriever using positive and negative question-paragraph pairs. Specifically, we leverage the DuReader benchmark, which provides human-annotated relevance labels for question-passage pairs, ensuring high-quality supervision during training. Positive samples are passages annotated as relevant to the given question, while negatives are randomly selected or chosen as hard negatives from the same context. After training, the retriever encodes all corpus paragraphs into dense embeddings, which are then indexed using FAISS (`IndexFlatIP`) for efficient top-k similarity search during retrieval.

Construction of*R*^{2}*AG* Training Samples. Each training instance comprises a question, reference answer, k retrieved paragraph embeddings, and a binary relevance label for each paragraph. Labels are heuristically assigned based on similarity with the gold answer using token overlap or ROUGE. These samples are used to jointly supervise both generation and reranking.

Relevance-Aware Generation via RFormer. During training, paragraph embeddings are processed by *R-Former*, a lightweight transformer encoder that outputs a relevance-guided vector. This vector is projected to match the hidden dimension of a frozen LLM (Here we used LLaMA) and injected into the prompt via a placeholder token. The model jointly optimizes generation loss (over the answer) and binary classification loss (over paragraph relevance).

Inference At inference time, *R*^{2}*AG* retrieves top-k passages, processes them via RFormer, and injects the reranking-aware signal into the prompt before decoding with the LLM.

F.5 LongRAG

We adopt *LongRAG* (Jiang et al., 2024) as one of our baselines. *LongRAG* enhances retrieval-augmented reasoning through a dual-perspective design consisting of a *Hybrid Retriever* for coarse-to-fine semantic recall, an *LLM-augmented Information Extractor* for global context recovery, and a *CoT-guided Filter* for relevance refinement. To adapt *LongRAG* to our scientific literature corpus and domain-specific QA task, we apply several modifications.

First, each paper is segmented into semantically coherent chunks of approximately 200 words, using sentence boundaries as the minimal division unit. Each chunk is encoded with the `intfloat/multilingual-e5-large` dual encoder and stored in a FAISS index for dense retrieval. In the fine-grained re-ranking stage, the cross-encoder

`nreimers/mmarco-mMiniLMv2-L12-H384-v1` is used with default settings (`chunk_size = 200`, `top-k = 7`).

Second, the retrieved chunks are mapped back to their original paragraphs to restore paragraph-level semantic continuity. When multiple chunks correspond to the same paragraph, only the most relevant one is retained. The resulting paragraphs are passed to an LLM-based extractor, which generates a global contextual representation capturing structural and semantic relationships across the paper.

Third, we retain LongRAG's two-stage CoT-guided filtering mechanism to eliminate redundant or irrelevant content. The model produces intermediate reasoning traces to identify evidence-bearing segments and synthesizes an intermediate answer by combining globally summarized and filtered information. In our adaptation, GPT-3.5-Turbo-16k serves as the backbone model for extraction, filtering, and generation.

Finally, a more capable external LLM—such as GPT-4o, GPT-4.1, DeepSeek-R1, Gemini 2.5 Pro, o3, o4-mini, or Llama-3.1-70B-Instruct-Turbo—is employed to refine intermediate outputs and produce the final synthesized answers. Accuracy is reported as the primary evaluation metric, consistent with our multiple-choice benchmark protocol. The models used in each stage are summarized in Table 21.

F.6 Agent LUMOS

This section details our customized version of the LUMOS iterative framework(Yin et al., 2024), adapted specifically for the multiple-choice question (MCQ) task. Unlike the original LUMOS framework designed for open-domain QA, our version retains its iterative structure with separate **Planning**, **Grounding**, and **Execution** modules. At each step, the system generates subgoals, formulates paragraph retrieval actions, retrieves relevant evidence, and accumulates results. Once enough evidence is collected, we use an external LLM (GPT-4o, GPT-4-turbo, or DeepSeek-Chat) to synthesize the final answer based on all retrieved content.

Our method retains LUMOS's modular design, consisting of a planning module, grounding module, and execution module. The models used in each stage are summarized in Table 22:

The baseline system adopts an iterative multi-stage reasoning framework composed of four core modules: planning, grounding, execution, and final synthesis.

The process begins by initializing an intermediate result dictionary with the original question and an initial context. In each iteration, the planning module generates a reasoning subgoal based on the question, accumulated results, and previous actions. This is achieved using a pretrained model, `lumos_complex_qa_plan_iterative`, which is built on LLaMA 2–7B. The iteration process terminates either when the planning module outputs an explicit stop signal (e.g., "Yes, I will stop planning.") or when a fixed maximum of 5 steps is reached.

Once a subgoal is generated, the system proceeds to the grounding stage, where the subgoal is translated into a structured paragraph retrieval action conditioned on the current reasoning context.

The execution module is responsible for retrieving the most relevant paragraph. It segments the context into overlapping chunks and selects the top match based on semantic similarity using a custom retriever. To evaluate the effect of evidence granularity, we test retrieval based on the top-5, top-7, and top-9 most relevant paragraphs.

After sufficient evidence has been collected through iterations, the system enters the final synthesis stage. All retrieved content is compiled into a standardized prompt, including the original question and numbered evidence chunks. This prompt is then passed to an external large language model—such as GPT-4-turbo, GPT-4o, or DeepSeek-Chat—to generate the final answer. The model is instructed to synthesize an answer *strictly based on the retrieved content*, ensuring transparency and faithfulness.

This modular design supports step-wise and interpretable reasoning, separates control logic from retrieval, and uses the LLM exclusively in the final answer synthesis stage.

F.7 PaperQA2

We reproduce and adapt the *PaperQA2* framework for our controlled setting involving a small fixed corpus of input PDFs(Skarlinski et al., 2024), rather than a large open-domain scientific literature database. The original PaperQA2 pipeline includes dynamic *Paper Search* and *Citation Traversal* modules, which rely on inter-paper citation graphs and large-scale retrieval. Since our corpus consists of a limited number of static papers without citation links, these modules are disabled in our adapta-

Stage	Model / Tool
Dual-Encoder Retrieval	`intfloat/multilingual-e5-large`
Cross-Encoder Re-ranking	`nreimers/mmarco-mMiniLMv2-L12-H384-v1`
Information Extraction (Global)	`GPT-3.5-Turbo-16k`
CoT-guided Filtering (Local)	`GPT-3.5-Turbo-16k`
Answer Generation (Intermediate)	`GPT-3.5-Turbo-16k`
Final Answer	External LLMs: `GPT-4o`, `GPT-4.1`, `DeepSeek-R1`, `Gemini 2.5 Pro`, `o3`, `o4-mini`, `Llama-3.1-70B-Instruct-Turbo`

Table 21: Models used in each stage of the *LongRAG* baseline.

Module	Role	Model Used
Planning	Generate reasoning subgoals	`LLaMA 2-7B`
Grounding	Convert subgoals into retrieval actions	/
Execution	Execute retrieval over provided context	`dpr-reader-multiset-base`
Final Answer	Synthesize final answer using retrieved content	External LLM (GPT-4-Turbo, GPT-4o, or DeepSeek-Chat)

Table 22: Models used in each stage of the *LUMOS* baseline.

	GPT-4o				GPT-4.1				DeepSeek-R1			
	Phys	PH	ES	Avg.	Phys	PH	ES	Avg.	Phys	PH	ES	Avg.
LUMOS (3 chunk)	43.1	36.4	41.7	40.4	41.5	41.8	40.0	41.1	38.5	40.0	38.3	38.9
LUMOS (5 chunk†)	52.3	43.6	46.7	47.5	50.8	47.3	51.7	49.9	50.8	49.1	46.7	48.9
LUMOS (7 chunk)	58.5	49.1	51.7	53.1	56.9	52.7	55.0	54.9	55.4	56.4	51.7	54.5
LUMOS (9 chunk)	61.5	52.7	53.3	55.8	60.0	56.4	56.7	57.7	58.5	61.8	55.0	58.4
`IntrAgent` (ours)	**75.4**	**58.2**	**63.3**	**65.6**	**78.5**	**69.1**	**70.0**	**72.5**	**72.3**	**67.3**	**63.3**	**67.6**

Table 23: Accuracy (in percentage) of LUMOS across the three domains in `IntraBench` —SERS in chemistry physics (Phys), *infectious-disease modeling* in public health (PH), and *remote sensing* in earth science (ES)—under varying chunk configurations, with GPT-4o, GPT-4.1, and DeepSeek-R1 backbones. LUMOS achieves consistent performance gains as chunk size increases, peaking at 55.8%, 57.7%, and 58.4% in the Phys, PH, and ES domains, respectively. The default 5-chunk setup strikes a balance between accuracy and efficiency across all models.

Note. LUMOS with 5 chunks† is the default setting.

tion. Instead, we preserve the core reasoning-based retrieval and summarization components of PaperQA2, ensuring a faithful yet context-appropriate baseline implementation.

Each question–document pair is processed through a streamlined multi-stage pipeline consisting of: (1) **Document Parsing**, where each input PDF is converted to structured text using `Grobid`; (2) **Chunking & Indexing**, where parsed text is segmented into semantically coherent units and indexed with `FAISS`; (3) **Dense Retrieval**, using `intfloat/multilingual-e5-large` to identify the most relevant chunks to the input query; (4) **RCS (Contextual Summarization)**, where `GPT-4o` performs reasoning-based summarization and relevance scoring to produce high-quality evidence summaries; and (5) **Answer Generation**, where the top-ranked evidence summaries are consolidated and passed to external LLMs (`GPT-4o`, `GPT-4.1`, `DeepSeek-R1`, `Gemini 2.5 Pro`, `o3`, `o4-mini`, `Llama-3.1-70B-Instruct-Turbo`) for final response synthesis. All models operate with temperature fixed at 0.0 to ensure deterministic factual outputs.

The dense retriever (`intfloat/multilingual-e5-large`) encodes both query and chunk representations under cosine similarity. For each question, the top-k = 30 retrieved chunks are summarized by the RCS stage. The final answer generation step utilizes one of the listed external LLMs, which directly synthesize concise, evidence-grounded answers. The overall model configuration and stage-wise mapping are summarized in Table 24.

F.8 Agentic-Hybrid-Rag

This section presents the implementation and evaluation setup of the *Agentic-Hybrid-Rag* baseline(Nagori et al., 2025), adapted to our experimental environment featuring a small, offline

Stage	Model / Tool
Document Parsing	`Grobid` (structure-aware scientific text parser)
Dense Retrieval	`intfloat/multilingual-e5-large` (dual encoder for chunk retrieval)
RCS (Contextual Summarization)	`GPT-4o` (reasoning-based summarization and relevance scoring)
Final Answer	External LLMs: `GPT-4o`, `GPT-4.1`, `DeepSeek-R1`, `Gemini 2.5 Pro`, `o3`, `o4-mini`, `Llama-3.1-70B-Instruct-Turbo`

Table 24: Models used in each stage of the *PaperQA2* baseline.

corpus consisting of a few domain-specific PDF documents. Since the original *Agentic-Hybrid-Rag* framework relies on large-scale online bibliographic sources (e.g., PubMed, ArXiv, Google Scholar) and a server-based Neo4j knowledge graph, we develop a lightweight variant suitable for a fully local, closed-data scenario.

The customized version retains *Agentic-Hybrid-Rag*'s dual-retrieval architecture—combining **graph-based retrieval** and **vector-based retrieval**—but replaces all cloud-dependent components with locally executable modules. The system is organized into three main stages: **Document Preprocessing**, **Dual Retrieval**, and **Final Answer Synthesis**. The models and resources used are summarized in Table 25.

Each PDF in the corpus is first converted to structured text using `pdfminer.six`. Section headers and paragraphs are segmented based on typographic cues, and both the extracted text and associated metadata (title, author, year, and keywords) are serialized into JSON format. The resulting dataset thus supports both symbolic (graph) and semantic (vector) indexing without external dependencies.

To emulate HybridRAG's structured reasoning component without a full Neo4j server, we construct an in-memory knowledge graph using `NetworkX`. Nodes represent papers, authors, and keywords, while edges encode relations such as `written_by` and `has_keyword`. Given a question involving bibliographic relations (e.g., "Which study by Author X discusses…?"), the agent issues rule-based lookups that traverse this lightweight graph and retrieve the corresponding document set.

For semantic retrieval, all document sections are embedded using the `all-MiniLM-L6-v2` sentence transformer and indexed in FAISS. During inference, queries are encoded into the same embedding space, and the top-k passages are selected based on cosine similarity. To approximate HybridRAG's hybrid search strategy, BM25 lexical scores are linearly combined with dense similarity scores before reranking. This design ensures that both surface-level and semantic signals are leveraged despite the small data scale.

The retrieved passages from both retrieval modes are merged into a unified context, ordered by relevance, and passed to an answer generator. For fairness across baselines, we use the same instruction-tuned model (`Mistral-7B-Instruct`) for all answer synthesis tasks. The model is explicitly prompted to reason *only over the retrieved evidence* to avoid hallucination.

The baseline is evaluated on the same set of expert-curated questions used for `IntrAgent`. For each query, the router decides between graph or vector retrieval depending on the question type. We report accuracy using the LLM-grounded multiple-choice evaluation metric described in Section 5.2. This adaptation allows HybridRAG to operate fully offline while preserving its dual-retrieval reasoning capability, thus serving as a reproducible and interpretable baseline for comparison with `IntrAgent`.

F.9 SciMaster

We adopt *SciMaster* (Chai et al., 2025) as the baseline framework for tool-integrated agentic reasoning. *SciMaster* enables a reasoning model to dynamically interact with external environments through Python-based code execution, combining tool-augmented reasoning with a multi-stage inference-time workflow involving *Solver*, *Critic*, *Rewriter*, and *Selector* roles. To adapt *SciMaster* to our task setting involving domain-specific scientific literature and structured information extraction, we introduce several modifications.

First, we perform **Domain-Specific Query Adaptation**. Each research question from our benchmark is reformulated into SciMaster's standard input template, consisting of a user query followed by a reasoning block enclosed within `<think>` and `</think>` tags. This ensures that the baseline reasoning process remains consistent with the original workflow while aligning the query semantics with scientific reading tasks. The same backbone model, *DeepSeek-R1-0528*, is employed

Stage	Implementation / Model Used
Graph-based Retrieval	`NetworkX`(Query structured metadata relations)
Vector-based Retrieval	`all-MiniLM-L6-v2` + BM25 + FAISS (Retrieve semantically similar passages using hybrid sparse–dense retrieval)
Final Answer	External LLMs: `GPT-4o`, `GPT-4.1`, `DeepSeek-R1`, `Gemini 2.5 Pro`, `o3`, `o4-mini`, `Llama-3.1-70B-Instruct-Turbo` (Synthesize a concise, evidence-grounded response based on all retrieved information)

Table 25: Models used in each stage of the *Agentic-Hybrid-Rag* baseline.

with a temperature of 0.6 and a context length of 64k tokens, identical to the configuration reported in the original paper.

We replace the original general-purpose modules $\texttt{web}_s earchand$

G Experiments Compute Resources and Model Choices

We conducted all experiments using API access to GPT-4o-20240806, GPT-4.1-20250414, and DeepSeek-R1-20250120, o3-20250416, o4-mini-20250416, gemini-2.5-pro-20250617, and Llama-3.1-70B-Instruct-Turbo-20250723 – using their default settings.

Other experiments compute resources indicated in Table 26.

Component	Specification
GPU Model	NVIDIA RTX 3090
GPU Memory	24 GB per GPU
CPU Cores	22
System Memory	64 GB

Table 26: Hardware Configuration Used for Training and Inference

H Detailed Impact of Mapping Model Across Domains

To complement our primary analysis of mapping model robustness (Section Robustness of `IntrAgent` to Mapping Model and Input Variability), we present a comprehensive evaluation across all three scientific domains in `IntraBench`: SERS in chemistry physics (Phys), infectious-disease modeling in public health (PH), and remote sensing in earth science (ES). This analysis investigates how different mapping models perform when paired with short-form answers generated by `IntrAgent` and vanilla RAG under different backbone LLMs.

Specifically, we follow a standard evaluation protocol: short-form answers are first generated by two methods using three backbone LLMs—GPT-4o, GPT-4.1, and DeepSeek-R1. These responses are then mapped to multiple-choice (MCQ) selections using three separate mapping models (GPT-4o, GPT-4.1, and DeepSeek-R1), resulting in 18 unique backbone-mapping combinations. Each configuration is evaluated using a consistent prompt template. Accuracy scores for each domain and setting are reported in Table 27.

Regardless of the mapping model used, `IntrAgent` consistently outperforms the RAG baseline across all domains and backbone configurations on average. The performance gains remain substantial, demonstrating the robustness of `IntrAgent` 's short-form answers across different downstream mapping strategies. Across all nine mapping–backbone pairs, `IntrAgent` consistently exceeds the vanilla RAG baseline: averaged over the three scientific domains, its accuracy gains are 4.8%, 10.0%, and 7.7% when GPT-4o is the mapping model (for GPT-4o, GPT-4.1, and DeepSeek-R1 backbones, respectively); 8.0%, 11.0%, and 4.9% when GPT-4.1 performs the mapping; and 5.6%, 5.0%, and 8.7% when DeepSeek-R1 is used as the mapping model.

In addition, we revisited the results of hu-

	Backbone	GPT-4o				GPT-4.1				DeepSeek-R1			
Mapping	Method	Phys	PH	ES	Avg.	Phys	PH	ES	Avg.	Phys	PH	ES	Avg.
GPT-4o	IntrAgent	73.8	52.7	60.0	62.2	76.9	63.6	68.3	69.6	73.8	67.3	63.3	68.1
	RAG	63.1	50.9	58.3	57.4	66.2	52.7	60.0	59.6	63.1	58.2	60.0	60.4
GPT-4.1	IntrAgent	75.4	58.2	63.3	65.6	78.5	69.1	70.0	72.5	72.3	67.3	63.3	67.6
	RAG	60.0	52.7	60.0	57.6	66.2	60.0	58.3	61.5	64.6	63.6	60.0	62.7
DeepSeek-R1	IntrAgent	67.7	58.2	58.3	61.4	70.8	58.2	63.3	64.1	76.9	67.3	55.0	66.4
	RAG	61.5	50.9	55.0	55.8	66.2	52.7	58.3	59.1	63.1	58.2	51.7	57.7

Table 27: Accuracy (in percentage) of our IntrAgent and the Vanilla RAG baseline across three scientific domains in IntraBench —SERS in chemistry physics (Phys), infectious-disease modeling in public health (PH), and remote sensing in earth science (ES). Each row block corresponds to a mapping model (GPT-4o, GPT-4.1, or DeepSeek-R1), and each column block shows performance under a different backbone LLM. Short-form answers from each backbone are mapped to MCQ choices using each mapping model.

Method	**Top-1 Acc.**	**Top-2 Acc.**	**Top-3 Acc.**
Similarity-based (RAG)	25.0 (14/56)	32.0 (18/56)	50.0 (28/56)
Reasoning-based (IntrAgent)	87.5(49/56)	92.9(52/56)	94.6(53/56)

Table 28: Section title ranking accuracy (in percentage) on the physics dataset (Top-1, Top-2, and Top-3). Ground truth is based on expert-annotated sections.

man mapping annotation in collaboration with our physics domain expert. In a prior study, RAG achieved a score of 61 out of 65 when evaluated using GPT-4.1 as the mapping model. Upon closer inspection of the four mismatches, we found that in two cases the model's selections were arguably more accurate than those of the human annotator, indicating possible human error when the task is challenging (which causes potential ambiguities in answer interpretation). Nevertheless, even with minor fluctuations in performance across different mapping models, we consistently observe a substantial performance gap between the best baseline and our approach. This further reinforces our conclusion that IntrAgent achieves state-of-the-art performance in IntraView.

I Similarity-Based Section Selection

In addition to evaluating baseline approaches, we further investigate how to identify the section of a paper most relevant to a given research question by analyzing section titles. To this end, we conduct an additional experiment comparing two section selection strategies. The first is the reasoning-based ranking from the Section Ranking stage developed in IntrAgent. The second is a retrieval-based approach that selects the section whose title exhibits the highest cosine similarity with the research question, using a vanilla RAG framework.

This experiment focuses on the physics dataset, where domain experts have annotated the section titles from which correct answers are expected to originate. We first collect the section titles ranked by the reasoning-based method. For comparison, we replace this with the retrieval-based strategy described above, computing cosine similarity scores between each question and all section titles, and ranking them in descending order. Both methods are evaluated by comparing their top-3 ranked section titles against the expert-annotated ground truth. Results are reported in Table 28.

Note that in some cases where the correct answer is *F (none of the above)*—indicating that the relevant content does not appear in the paper—we exclude those questions from the evaluation, as there is no valid section title to compare. As a result, the physics dataset contains 56 questions that are valid for Section Ranking comparison.

We observe that our method achieves a top-1 accuracy of 87.5%, a top-2 accuracy of 92.9%, and a top-3 accuracy of 94.6%, in comparison to the RAG-based method, which yields an accuracy of 25.0% at top-1, 32.0% at top-2, and 50.0% at top-3 on the physics dataset.

We also evaluate end-to-end task performance using the standard protocol. When replacing our reasoning-based section ranking with the RAG-based reordering strategy, the overall accuracy drops to 66.2%, compared to 78.5% achieved by IntrAgent using GPT-4.1 as both the backbone and mapping model. While this substitution leads to a performance decline, the decrease is not sub-

stantial. These results highlight the robustness of our iterative reading stage: even when section titles fail to generate an ideal ranking, the model continues to perform strongly by reasoning over multiple retrieved sections.