ENHANCING PHYSICS PROBLEM SOLVING IN LLM AGENTS WITH KNOWLEDGE GRAPHS AND SYMBOLIC MATH TOOLS

Sviatoslav Gladkykh
ID: 2128780
Data and Artificial Intelligence cluster
Eindhoven University of Technology
s.gladkykh@student.tue.nl

August 25, 2025

ABSTRACT

Large Language Models (LLMs) are fluent in general conversations [18], yet they often fail on more specific, domain-focused tasks such as university-level physics questions that require both conceptual understanding and precise mathematical reasoning [31]. This work explores whether two external supports - structured retrieval from a domain knowledge graph and algebraic checking with a symbolic-math system can reliably close that gap. We will construct a knowledge graph from the *OpenStax University Physics* textbook [15], and potentially extend it with other open knowledge sources such as *Light and Matter* [5], allowing broader coverage and richer representations. The agent will query this graph through a relation-aware retrieval scheme. When the agent needs to manipulate equations or evaluate expressions, it will hand those steps to a custom verifier, ensuring arithmetic accuracy and valid units. Performance will be measured on the SciEval [22] and MMLU [11] benchmarks, across multiple controlled configurations, evaluating each component individually, in combination, and under alternative tool setups such as vector-based and graph-based retrieval. We aim to show that the combined system raises answer accuracy beyond both a standard general-purpose LLM and a specialized physics-fine-tuned LLM. Success would demonstrate a resource-efficient route for adapting general LLMs to technical domains, offer clearer reasoning traces than end-to-end fine-tuning, and provide a practical foundation for reliable assistants in science, technology, engineering, and mathematics applications where labeled data is limited and mathematical precision is essential.

1 Overall aim and goals

1.1 Motivation and Challenges

Motivation. The study is motivated by the following closely related goals:

- *Explainability*: each answer should be accompanied by a transparent trace of the facts and algebraic steps that produced it, moving the system away from the 'black box' behavior typical in current LLMs.
- *Efficiency*: if a lightweight agent, built from a physics knowledge graph and a symbolic-math verifier can match the performance of far larger fine-tuned models, it would point to a more economical route to dependable domain expertise and clarify the separate value of structured retrieval and external computation.
- *Transferability*: because the design focuses on interchangeable tools rather than task-specific adaptation, the same pattern should be adaptable to other science and engineering domains that demand precise, explainable outputs.

- *Skill development*: the work offers a timely opportunity for hands-on experience with agent frameworks, graph-construction pipelines, and function-calling interfaces that are becoming central to applied NLP research.

Key challenges. Several obstacles are likely to determine how far these aims can be realized:

- A suitable benchmark must first be identified — one that covers both numerical and conceptual physics questions, supports component-level comparisons, and offers reliable baselines from existing systems.
- The knowledge source must be rich enough to span those benchmarks yet structured well enough (preferably in XML or JSON) to enable reliable extraction of entities, relations, and formulas.
- Turning that source into a clean knowledge graph and linking it to a relation-aware retriever requires a carefully designed extraction pipeline.
- The symbolic verifier should be exposed through an interface that the agent can invoke with minimal prompt overhead while receiving clear, machine-readable results.
- All components then need to be orchestrated so the agent decides sensibly between graph lookups and mathematical checks, without drifting into endless loops or hallucinated tool calls.
- Whatever works for the initial physics chapters, in principle, should extend to additional topics and, ideally, to other STEM fields with different notation and patterns of reasoning.

1.2 Broad Literature Analysis

This subsection gives background for the project by linking prior work on knowledge retrieval, tool-using agents, and shared benchmarks and sources. We keep the focus broad and descriptive here; technical details follow in Section 3.1.

Knowledge retrieval. Recent physics QA studies explore two complementary ways to bring structure into reasoning. One line builds a knowledge graph from the question itself and uses it to break the task into smaller sub-questions, so the graph serves as a framework for reasoning rather than as a reusable store [1]. In contrast, work on retrieval from an existing graph aims to deliver only the most helpful slice of a large KG: instead of returning many loosely related nodes, the retriever brings back compact subgraphs with entities, relations, and nearby links, then aligns and filters them so the model sees a clean context [26]. A related approach goes one step further and asks the model to propose candidate triples which are then checked against the KG before use; this reduces hallucinations and makes provenance clear, but it depends on the graph being complete enough and on the base model being able to suggest schema-compatible triples [21]. As a text-centric counterpoint, physics RAG systems retrieve short passages from textbooks or reference material and fine-tune an open model to use that context, yielding more precise and grounded answers and offering a strong baseline against which graph-based retrieval can be compared [7].

Agents with tools. A simple and effective pattern for quantitative problems is to have the model map the prompt to equations, hand those to a symbolic solver, and then explain the result. This setup largely removes arithmetic and algebra mistakes while keeping the language model in charge of interpretation and presentation [10], and it matches our plan to verify physics steps with a symbolic tool. Broader scientific agents extend this idea by teaching models to choose and call external functions (for example from SymPy, NumPy, or SciPy) on demand; results show that well-orchestrated tool use can let smaller models outperform larger ones without tools, which is appealing for efficiency [17]. A recent physics-focused system improves answers by inspecting an initial solution and then applying targeted fixes: retrieving the right formula, recomputing numbers with code, or reframing the question [12]. The gains reported there support combining knowledge retrieval with external computation; our work takes a complementary route by integrating these checks during reasoning to keep solutions transparent and efficient.

Benchmarks, datasets, and sources. For evaluation, the SciEval [22] benchmark provides a multi-level test suite that includes physics questions covering knowledge recall, application, calculation, and problem solving, with both multiple-choice and free-response formats and dynamic subsets to reduce leakage. The MMLU [11] benchmark, widely used for assessing general-purpose language models, includes a physics subset that tests conceptual understanding and reasoning in a standardized setting, enabling comparisons with other academic disciplines. The SCIMAT-2 [14] dataset offers millions of science and math problems, including a substantial portion in physics, supplying diverse material for training and ablation studies. The PHYSICS benchmark [8] further complements these resources, consisting of more than one thousand university-level problems across domains such as mechanics, electromagnetism, quantum mechanics, and thermodynamics. Although formally classified as multimodal, the majority of its questions are text-only, making it suitable for the initial text-based setup while still allowing later extensions to multimodal reasoning.

As a grounded source of facts and formulas, the *OpenStax University Physics* textbook series [15] is widely used in calculus-based courses; their public XML repository makes parsing, entity and relation extraction, and formula capture practical for building a physics knowledge graph with citations. Complementary resources such as the *Light and Matter* series [5], which includes both algebra-based and calculus-based texts, and open encyclopedic sources like *Wikipedia*, can provide additional coverage and enrichment beyond OpenStax. While these sources vary in structure and level of detail, they broaden the range of available knowledge and may enhance the completeness of the resulting knowledge graph. A case study in bootstrapping ontology graphs from textbooks [25] demonstrates how to automatically generate ontology graphs from OpenStax (including *OpenStax University Physics*) with human validation, illustrating a workable path from chapter text to reusable concept–relation structures.

Overall, prior work indicates that structured retrieval, textbook grounding, and external tools can raise accuracy and reliability on physics questions. These strands inform our component choices and provide baselines and comparisons for the study.

1.3 Formulation of the Problem and Objectives

Overall aim. The project aims to design and evaluate a physics-aware LLM agent that combines (i) structured retrieval from a textbook-derived knowledge graph and (ii) a symbolic-math verifier for calculations. The agent will be orchestrated as a tool-using workflow and evaluated on college-level physics benchmarks to assess reliability, transparency, and efficiency relative to strong baselines.

Key objectives and research questions. The study focuses on three primary questions.

RQ1. How does retrieval from a structured knowledge graph compare to vector-based retrieval for physics question answering?

RQ2. What is the relative and combined impact of using knowledge retrieval and mathematical tools on the accuracy and reliability in multi-step derivations while ensuring physical consistency?

RQ3. How does the proposed physics-informed LLM agent perform against general-purpose LLMs, physics-fine-tuned models, and state-of-the-art methods under the same evaluation setup?

These are the principal questions; additional, component-level questions may be investigated as needed or if time permits.

Component-level sub-questions.

- *Knowledge acquisition and retrieval design:* Which data-extraction and storage choices from the selected physics knowledge sources yield the most useful and well-structured KG for problem solving, and which retriever configuration returns the most decision-relevant context to the LLM?
- *External computation:* Which symbolic or numeric libraries (e.g., SymPy variants, units toolkits) offer the best trade-off between reliability, ease of invocation, and integration with the agent's prompting and tool-calling interface?
- *Benefit analysis by topic:* Which types of physics topics or question categories benefit most from knowledge-graph-backed retrieval or from the use of an external verifier?
- *Provenance and citation:* How can retrieved subgraphs be linked back to the source textbook passages to enable citation, fact checking, and hallucination mitigation?

Time-permitting research questions.

- Can an auxiliary corpus of solved Q&A be leveraged for few-shot guidance or case-based retrieval to further improve the agent's task execution?
- Can the agent be extended to multimodal physics tasks (e.g., diagrams, plots), and what uplift results on multimodal benchmarks relative to generic and physics-tuned multimodal LLMs?
- Can the approach scale to master's/PhD/olympiad-level problems with an appropriately enriched knowledge base, and how does it compare with existing solutions on very challenging benchmarks?
- Can verifiable rewards be defined for each reasoning step of the agent, enabling reinforcement learning to further improve performance and consistency?

2 Research approach

2.1 Overall methodology and decomposition

The project will follow a staged approach that moves from preparation and baseline construction to iterative refinement, extended evaluation, and in-depth analysis. Each stage is designed to address the main research questions, with results from earlier steps informing the design choices in later ones.

Part I: Preparation and baseline design. We will begin by finalizing the selection of benchmarks and knowledge bases to be used in the project. An evaluation pipeline will be created and applied to a range of models, both general-purpose and physics-fine-tuned, in order to select the most suitable base model(s) for the agent and for comparative baselines. In parallel, we will review and select frameworks for knowledge graph construction, storage, and retrieval; for agent orchestration; and for symbolic verification. One or more physics topics with substantial overlap between the chosen textbook and benchmarks will be identified for the initial baseline experiments.

Part II: Baseline implementation and evaluation. The selected knowledge base will be parsed into two formats: a structured knowledge graph and a chunked vector store. A first baseline agent will then be implemented, combining two retrieval mechanisms (KG-based and vector-based) with a symbolic verifier. This configuration will be evaluated on the selected benchmarks, and the results will guide initial adjustments to the component setups.

Part III: System extension and comparative setups. The knowledge bases will be expanded to cover additional topics and, if appropriate, supplemented with other reliable sources. Multiple agent configurations will be developed in line with the research questions, including versions using each component individually, in combination, and with alternative tool integrations. Each configuration will be benchmarked, and the findings will inform further refinements.

Part IV: Detailed analysis, optional extensions, and thesis writing. A thorough analysis of the final results will be carried out, including an examination of error sources, topic-specific performance, and the reasoning traces produced by tool calls. If time permits, additional research directions will be explored: incorporating an auxiliary Q&A corpus for few-shot guidance, adding multimodal task support, and scaling the approach to more complex problems and richer knowledge sources. The work will conclude with the writing of the thesis, synthesizing the findings, methodology, and contributions into a coherent final document.

2.2 Methods and techniques

The project will be implemented primarily in Python, with source code managed using the *Git* version control system to ensure reproducibility, openness, and effective tracking of changes. The core of the system will be an agent-based architecture built with the *LangChain* and *LangGraph* frameworks. Reasoning and tool usage will follow the *ReAct* [28] paradigm, with the possibility of incorporating structured reasoning prompts such as Chain-of-Thought (CoT) [24], Tree-of-Thought (ToT) [27], or Graph-of-Thought (GoT) [3] as an initial reasoning step before tool invocation.

For structured knowledge storage and retrieval, we plan to use the *Neo4j* graph database and/or Microsoft's *GraphRAG* [6] framework, depending on performance and integration considerations. The symbolic verification component will be implemented using either individual libraries such as *SymPy*, *SciPy*, and *NumPy*, or a combination of these. In some cases, the agent may write and execute custom Python code directly, combining multiple libraries or using basic Python operations to achieve the required computation.

The choice of base language model for the agent will be determined after initial benchmarking. This selection will balance performance on physics-specific tasks, model size, and available computational resources. For larger models beyond local compute capacity, we may employ API-based access (e.g., OpenAI API) to ensure flexibility.

The knowledge graph will be built from the *OpenStax University Physics* [15] XML source, which offers well-structured content suitable for entity, relation, and formula extraction. We will explore both manual XML parsing, using regular expressions and tree traversal, and LLM-assisted parsing, potentially combining the two approaches to maximize accuracy and coverage. Formula representation conversion between formats such as MathML, LaTeX, and *SymPy* expressions will be performed using tools such as *sbmlmath* and *SymPy*, enabling consistent storage, retrieval, and symbolic manipulation. In addition to OpenStax, supplementary knowledge sources such as the *Light and Matter* series [5] or selected *Wikipedia* pages may be incorporated to broaden coverage and enrich the graph with additional concepts, relations, and examples.

Where applicable, we will also maintain a chunked vector store of textbook content to allow for direct comparison between vector-based retrieval and graph-based retrieval. This dual approach will help in evaluating the impact of

structured versus unstructured retrieval methods on problem-solving accuracy. Possible options for the vector store include open-source frameworks such as *FAISS*, *Weaviate*, *Milvus*, or *Chroma*, with the final choice to be determined during the initial phase of the research.

2.3 Research plan and timeline

The project will follow the four stages outlined in Subsection 2.1, moving from preparation and baseline development to extended experimentation, in-depth analysis, and thesis writing. Each stage produces tangible outputs that inform the next, ensuring that the work proceeds in a structured and iterative manner. The timeline below reflects the planned allocation of time for each part of the methodology, along with the key expected results and deliverables.

Table 1: Planned timeline, milestones, and deliverables for the MSc Graduation project.

Period	Description	Deliverables
Sep 2025 – Oct 2025	**Part I: Preparation and baseline design.** Finalize benchmarks and knowledge bases, set up evaluation pipeline, select base models and frameworks, define initial topics.	Baseline design plan, documentation of choices.
Oct 2025 – Dec 2025	**Part II: Baseline implementation and evaluation.** Parse KB into KG and vector store, implement baseline agent with dual retrieval and symbolic verification, run first benchmarks.	Baseline agent code, evaluation report.
Dec 2025 – Feb 2026	**Part III: System extension and comparative setups.** Expand KB, add alternative tools and retrieval setups, benchmark multiple configurations.	Extended system code, comparative evaluation report.
Feb 2026 – May 2026	**Part IV: Analysis, extensions, thesis writing.** Error analysis, topic-specific results, tool trace study, optional extensions, thesis writing.	Final analysis report, thesis draft.
May 2026 – Jun 2026	**Defense.** Finalize thesis, prepare and deliver defense presentation.	Defended MSc thesis.

2.4 Identified risks and their mitigation

In Section 1.1 we outlined several challenges relevant to this research. Here, we present potential risks that may arise during the development phase and describe mitigation strategies to address them.

Computational resources. The availability and capacity of university-provided computational resources may be limited. If local infrastructure is not sufficient for running the selected model(s), the plan is to use API-based inference (e.g., via the OpenAI API), which runs on dedicated cloud servers. These services may incur costs, but they are expected to remain within a manageable budget for the project.

Parsing the knowledge base. While the *OpenStax University Physics* textbook provides a structured XML format, some important quantities, concepts, and explanations may be embedded in plain text. The material also contains in-text images and diagram references that must be filtered. If manual parsing using XML tree traversal and regular expressions does not provide adequate coverage, LLM-assisted extraction will be used either to supplement manual methods or, where necessary, to process specific sections entirely. As an additional fallback, or to broaden coverage, other open resources such as the *Light and Matter* series [5] or selected *Wikipedia* pages may be incorporated to complement OpenStax and fill gaps in the knowledge graph.

Knowledge graph performance. The structured knowledge graph may produce results that are comparable to, or worse than, vector-based retrieval alone. In such cases, the first step will be to investigate potential causes, such as graph construction quality or retrieval strategy. It is important to note that vector-based retrieval can also be implemented within a knowledge graph by storing vector embeddings for relevant nodes or documents, enabling a hybrid retrieval approach. If neither pure graph-based retrieval nor hybrid graph–vector retrieval offers improvements, a detailed analysis will be provided, and subsequent experiments may rely on a standalone vector-based retrieval setup instead.

Agent effectiveness. The final agent configuration may fail to deliver a substantial improvement in physics problem-solving accuracy compared to a general-purpose baseline. To mitigate this risk, comparison models will be selected early in the project, ensuring that the chosen base model is not fine-tuned for physics and does not include built-in mathematical solvers. This will help isolate and measure the contributions of both the external knowledge base and the symbolic verification component. Even if accuracy gains are limited, the agent may still provide improvements in other important metrics such as reasoning transparency, traceability of intermediate steps, interpretability of outputs, and consistency in applying physical principles, which could represent valuable outcomes in their own right.

2.5 Knowledge utilisation, valorisation, and expected contributions and impact

Potential. The primary contribution of this project lies in advancing methods for integrating structured knowledge retrieval, agent-based reasoning, and symbolic computation in the context of physics problem solving. The outcomes will be relevant to several academic areas, including natural language processing, information retrieval, knowledge graph construction, and intelligent tutoring systems. Within physics education research, the proposed approach may inform the design of AI-assisted learning tools that deliver accurate, transparent, and verifiable solutions. Beyond academia, the techniques developed may be of interest to organisations working on educational technology, automated reasoning, and scientific knowledge management. The modularity of the system, particularly in its retrieval and tool orchestration components, makes it applicable to other domains that rely on domain-specific corpora and structured knowledge, such as engineering, biomedical sciences, and law.

Implementation. Results from the project will be disseminated through open-source code releases, enabling other researchers and practitioners to replicate and extend the work. Where possible, the knowledge graph and vector store derived from openly licensed sources such as *OpenStax* and supplementary sources such as *Light and Matter* and Wikipedia will be shared to support reproducibility and facilitate further research. Potential end-users, including educators and developers of digital learning environments, could adopt the retrieval and reasoning framework to create interactive systems that assist learners in solving complex problems with step-by-step explanations and validated computations. Scientific and technical communities may benefit from the improved retrieval–reasoning synergy demonstrated in the project, which can be adapted to other knowledge-intensive tasks. Dissemination may be supported by academic publications, presentations at relevant conferences, and engagement with open research forums, ensuring the work reaches both the scientific community and interested external stakeholders.

3 Evidence that your research can succeed

This section provides a detailed review of prior research most closely aligned with the objectives of this project. We focus on studies that combine language model reasoning with structured knowledge retrieval, particularly those employing knowledge graphs and symbolic verification tools for problem solving. Special attention is given to agent architectures that orchestrate multiple tools, retrieval strategies, or reasoning frameworks, as these represent the closest analogues to the proposed system. We also review relevant benchmarks and datasets used to evaluate such systems, with emphasis on those containing physics-specific material or enabling cross-domain comparisons. Where applicable, we highlight concrete design patterns, empirical findings, and technical choices from prior work that directly inform our planned methodology. By identifying approaches that have demonstrated measurable benefits in related contexts, this analysis not only situates the project within the broader research landscape but also provides tangible evidence that the proposed methods have a strong potential to succeed.

3.1 Background and In-depth Literature Analysis

Datasets and benchmarks. A key component of this research is the careful selection of datasets and benchmarks that can be used both for evaluation and, where appropriate, for fine-tuning. This subsection examines several resources in depth, assessing their structure, coverage, and suitability for the intended use cases, with the aim of providing clear evidence of their applicability to the project.

One of the most relevant resources is *SCIMAT-2* [14], an open-source dataset containing millions of mathematics and science problems at pre-college and college level, including a substantial number of physics questions. The official repository provides well-structured Python utilities for generating unlimited question–answer pairs with randomized input values. The dataset is hierarchically organized: the two broad domains of mathematics and science are each subdivided into subjects, which are further divided into specific skills being assessed. This fine-grained structure is particularly valuable for our purposes, as it allows us to target specific physics topics for initial experiments and to ensure that the assessed concepts are also represented in the knowledge graph. In addition, the availability of randomization functions makes it possible to generate large, diverse problem sets for robust benchmarking or fine-tuning.

Another benchmark of high relevance is *SciEval* [22], a comprehensive multi-disciplinary evaluation suite comprising approximately 18,000 questions. It spans the core scientific domains of physics, chemistry, and biology, and is explicitly designed to test large language models along four dimensions: basic knowledge, knowledge application, scientific calculation, and research/problem-solving skills. These dimensions are derived from Bloom's taxonomy and reflect the different cognitive levels required in scientific reasoning. SciEval is also well structured, with questions organized by subject, question type, ability category, and topic. For physics, even narrower topics contain enough questions to support statistically meaningful evaluations. The benchmark's inclusion of both multiple-choice and free-response formats, along with a *dynamic subset* mechanism to mitigate leakage, makes it an especially rigorous evaluation tool for this project.

The *Massive Multitask Language Understanding* benchmark (MMLU) [11] is another widely adopted evaluation suite, consisting of multiple-choice questions across a broad range of academic disciplines. Within physics, MMLU includes categories such as *college physics*, *conceptual physics*, and *high school physics*, covering a spectrum of difficulty levels. However, MMLU's physics subset is relatively small compared to SciEval, and its lack of fine-grained topic or skill annotations limits its usefulness for targeted early-stage experiments. Nevertheless, its popularity in related research and broad coverage make it valuable for cross-study comparisons and for later stages of the project, especially if we extend evaluation to the full range of physics topics or develop methods to cluster questions by topic.

The *PHYSICS* benchmark [8] represents one of the most advanced resources currently available for scientific problem solving. It consists of 1,297 university-level problems covering six major domains: classical mechanics, electromagnetism, quantum mechanics, thermodynamics and statistical mechanics, atomic physics, and optics. All questions are open-ended and require multi-step derivations, with solutions expressed in either numeric or symbolic form. Automated verification using symbolic mathematics tools (e.g., SymPy) ensures objective evaluation. Given its depth and rigor, PHYSICS is especially valuable for the later stages of this project, where the aim is to assess the reasoning capability of the agent in complex and knowledge-intensive scenarios.

The *Massive Multi-discipline Multimodal Understanding* benchmark (MMMU) [29] is a large-scale evaluation suite that tests model performance on multimodal college-level questions. It contains over 11,000 questions spanning diverse academic disciplines, of which 433 are physics questions requiring reasoning over both textual and visual inputs such as diagrams, plots, and tables. All questions are multiple-choice, with detailed annotations for topic and difficulty. The physics subset provides an excellent opportunity to evaluate multimodal retrieval and reasoning strategies, particularly if our system is extended to handle image-based tasks.

The *OlympiadBench* dataset [9] provides a complementary challenge, consisting of 456 high-difficulty physics problems from international physics olympiads and related competitions. These questions are open-ended and often accompanied by diagrams, requiring advanced reasoning, problem decomposition, and symbolic manipulation. Unlike more general-purpose benchmarks, OlympiadBench directly tests the system's ability to operate at the upper limit of pre-university problem solving. This makes it a valuable addition when testing the scalability of the approach toward highly competitive settings.

Several other benchmarks were considered but found less suitable for the current project scope. For example, *SciBench* offers college-level problems but does not separate questions by subject, limiting its diagnostic value. *MMMU-Pro* is too small, with only 60 physics questions, while *ScienceQA* primarily targets elementary and middle school levels and is therefore too simple for the objectives of this study. Similarly, *JEEBench* contains only 123 questions, making it statistically insufficient, and *SciQ*, although large, is open-ended without structured topics, which reduces comparability across models. Finally, while *GPQA* provides expert-level multiple-choice questions, its scale is limited, and its primary focus is beyond the level of most of the benchmarks considered here. A consolidated overview of all considered datasets, along with their characteristics and selection decisions, is provided in Table 2.

Overall, *SciEval* emerges as the most suitable starting point for this project, as it is widely used, sufficiently large and complex, and organized into well-defined topics that are essential for the initial phases of the research. In addition, both *PHYSICS* and *SCIMAT-2* may serve as complementary benchmarks in the early stages, given their adequate knowledge level and good coverage of physics topics, though their more limited adoption raises some concerns regarding broader applicability. Once a more comprehensive knowledge base has been constructed, *MMLU* can provide valuable cross-study comparisons, despite the fact that its subcategories contain relatively few questions; its strength lies in assessing physics knowledge across different complexity levels. Finally, benchmarks such as *PHYSICS*, *MMMU*, and *Olympiad-Bench* will be particularly relevant if multimodal capabilities are added to the agent, enabling evaluation across text, diagrams, and advanced competition-level problem settings.

Physics Knowledge Sources. In evaluating possible sources for constructing a structured physics knowledge base, several open-source options were considered. The *Light and Matter* [5] series by Benjamin Crowell is a free, openly available collection of introductory physics textbooks covering mechanics, electromagnetism, optics, and modern

Table 2: Comparison of benchmarks with physics subsets: modalities, knowledge levels, question types, size, topic availability, and selection decision.

Benchmark	Multimod.	Knowledge Level	Question Type	#Physics Questions	Topic present	Decision
SciEval [22]	Text only	high school–undergraduate	multiple-choice	1478	Yes	Accepted (for text-only)
SCIMAT-2 [14]	Text only	pre-college–college	open (numeric/formula)	∼1041 types	Yes	Accepted (for text-only)
MMLU [11]	Text only	high school–college	multiple-choice	102–235 (subsets)	No	Accepted (for text-only)
PHYSICS [8]	Combined	university-level	open (numeric/formula)	1297	Yes	Accepted (for multi-modality)
SciBench [23]	Combined	college	open (numeric/formula)	N/A (789 total)	Yes	Rejected (few topics, not split by subject)
MMMU [29]	Images only	college	multiple-choice	408	Yes	Accepted (for multi-modality)
MMMU-Pro [30]	Images only	college	multiple-choice	60	No	Rejected (too few questions)
ScienceQA [16]	Combined	elementary–middle school	multiple-choice	1923	Yes	Rejected (too simple)
Olympiad-Bench [9]	Images only	olympiad	open ended	456	Yes	Accepted (for multi-modality and olympiad-level)
JEEBench [2]	Text only	pre-engineering college	mixed types	123	Yes	Rejected (too few questions)
GPQA [19]	Text only	domain experts	multiple-choice	187	Yes	Rejected (too few questions)
SciQ [13]	Text only	middle–high school	open ended	N/A (13.7K total)	No	Rejected (open ended)

physics. Its modular structure and clear explanations make it attractive for conceptual grounding, but the material is primarily algebra-based, with only optional calculus sections, and it is less comprehensive in scope compared to a full calculus-based course. In addition, while the material is available in PDF, HTML, and LaTeX source formats, LaTeX still requires substantial pre-processing to extract entities, formulas, and relations in a structured form suitable for direct integration into a knowledge graph.

The *CK-12 FlexBooks* [4] and *The People's Physics Book* similarly offer free and modular high-school and early-college-level physics resources. While they provide useful summaries, worked examples, and practice problems, their depth and mathematical rigor are limited compared to advanced university-level material. Moreover, CK-12's format, although accessible via API, is designed for interactive web use, and content consistency can vary across topics.

Motion Mountain: The Adventure of Physics [20] by Christoph Schiller is an ambitious, engaging, and wide-ranging open textbook series covering classical and modern physics. It is notable for its unconventional style, breadth of topics, and inclusion of curiosities and paradoxes. However, while it is freely available for personal use, its restrictions on modification may limit direct reuse in our pipeline. Additionally, its continuous narrative style and PDF-only distribution pose challenges for structured parsing and targeted fact or relation extraction.

OpenStax University Physics [15] emerged as one of the most suitable primary sources for our purposes. This three-volume, calculus-based textbook is widely adopted in two- and three-semester university physics courses. Volume I covers mechanics, oscillations, and waves; Volume II addresses thermodynamics, electricity, and magnetism; and Volume III focuses on optics and modern physics. The text is designed to connect theory with application, presenting rigorous mathematical treatments alongside accessible explanations and numerous worked examples that illustrate problem-solving strategies. Crucially, OpenStax provides the entire content in a public `GitHub` repository as structured XML, making it directly amenable to parsing, entity/relation extraction, and formula conversion for building a high-quality physics knowledge graph.

A particularly relevant precedent is the case study *A Case Study in Bootstrapping Ontology Graphs from Textbooks* [25], in which researchers developed a pipeline to automatically extract taxonomies and relations from OpenStax textbooks, including all three volumes of *OpenStax University Physics*. Their approach involved generating an ontology graph

of physics concepts. Nodes such as *force*, *mass*, and *acceleration* with labeled relations like *force is proportional to→ acceleration*, validated via crowd-sourcing. This work demonstrates a concrete and successful methodology for transforming textbook content into a structured, queryable graph, aligning closely with our own objectives.

We also note the potential utility of Wikipedia as a supplementary knowledge source for expanding coverage to topics beyond the scope of the OpenStax text, such as more advanced or niche areas of physics. Wikipedia's breadth, regular updates, and open accessibility make it a valuable candidate for targeted enrichment. However, its encyclopedic style lacks the pedagogical sequencing of textbooks, and the level of detail can be inconsistent across topics, necessitating careful curation to ensure relevance and correctness. Despite these limitations, its structured interlinking of concepts could complement the knowledge graph with additional entities and relations when needed. In a similar way, the *Light and Matter* series [5] may provide additional utility as a modular, openly available textbook resource, offering both algebra-based and calculus-based treatments of core physics topics that can complement OpenStax material. A comparative overview of the considered knowledge sources, including their accessibility, coverage, strengths, and limitations, is presented in Table 3.

Table 3: Comparison of physics knowledge sources evaluated for constructing a structured knowledge base.

Source	**Accessibility Format**	**Coverage & Depth**	**Strengths**	**Limitations**
Light and Matter (Crowell) [5]	PDF, HTML, LaTeX source	Introductory physics (algebra-based and calculus-based)	Extensive modular series (introductory and calculus-based), free and open-source	Requires preprocessing to extract structured entities, relations, and formulas suitable for KG integration
CK-12 FlexBooks; The People's Physics Book [4]	Web-based modular textbooks, API (CK-12)	High-school to early-college topics	Free, includes examples and practice problems, flexible access (API for CK-12)	Limited mathematical rigor, inconsistent across topics, web-only format for CK-12
Motion Mountain (Schiller) [20]	PDF (narrative style)	Broad conceptual coverage (classical and modern physics)	Engaging, whimsical approach, wide topic breadth	Not structured, PDF-only, licensing restricts reuse and modification
OpenStax University Physics [15]	XML (GitHub repository)	Full calculus-based three-volume physics (mechanics to modern physics)	Structured XML format, widely adopted, rigorous mathematical exposition, rich worked examples	Requires parsing pipeline and ontology design for KG construction
Wikipedia (including Wikidata, DBpedia, YAGO)	HTML + structured APIs/SPARQL endpoints	Broad, evolving coverage of physics concepts	Large-scale, cross-linked, regularly updated knowledge graph sources; suitable for supplementary enrichment (e.g., Wikidata has 1.65 billion statements)	Not pedagogically structured; variable quality and detail; requires careful curation

Related work demonstrating similar mechanisms. Several recent studies offer strong precedent for the components and functionality proposed in this project.

First, *Solving Math Word Problems by Combining Language Models With Symbolic Solvers* [10] addresses the tendency of LLMs to make arithmetic or algebraic errors. The authors present a method in which the model translates a natural-language problem into a formal, declarative representation (e.g., equations), passes it to a symbolic solver (SymPy) to compute a correct solution, and then verbalizes the result in natural language. The approach achieved comparable accuracy to Program-Aided Language (PAL) on the GSM8K benchmark and outperformed it by 20 percentage points on a more challenging Algebra dataset. This work provides solid evidence that offloading mathematical steps to a symbolic engine can correct LLM errors, a strategy we adopt and extend for physics.

Second, the *SciPhyRAG* [7] approach improves physics question answering by combining dense retrieval of unstructured text with fine-tuning. Using a Vicuna-7B model trained on retrieval-augmented prompts, the method achieved a +16.7% BERTScore increase and +35% ROUGE-2 improvement on high-school and university-level physics questions. This demonstrates that retrieving contextual text substantially improves performance, supporting our plan to use structured KG retrieval as a potentially even stronger alternative.

Third, the work on *Improving Physics Reasoning in Large Language Models Using Mixture of Refinement Agents (MoRA)* [12] provides evidence that combining retrieval and symbolic computation can substantially boost accuracy. MoRA uses multiple specialized agents to correct different errors after an initial solution: formula retrieval via GraphRAG, numeric recomputation via Python code, and reframing of misinterpreted prompts. This refinement strategy achieved up to a 16% absolute accuracy gain on SciEval and MMLU subsets using open-source LLMs. Although MoRA closely resembles our design in combining knowledge retrieval with verification tools, there are key differences: MoRA relies on a larger model's guidance to identify mistakes, while our approach integrates tool usage directly during reasoning (e.g., via CoT, ToT, or GoT decomposition), rather than post-hoc correction. We also aim to construct a more lightweight agent that achieves efficiency without relying on a larger supervising model.

Taken together, these works demonstrate the core benefits of structured retrieval and symbolic verification, supporting the feasibility of our proposal. However, our approach advances novelty and efficiency by embedding these capabilities within the reasoning process itself, rather than treating them as external fixes or relying on oversized models.

3.2 Preliminary studies and analyses

In the preparatory stage of this research, several activities were undertaken to validate the feasibility of the proposed approach, gain practical experience with relevant frameworks, and outline the core system design.

First, during the literature review and dataset analysis, we examined the structure and topical coverage of the selected benchmarks, to confirm their applicability. The analysis verified that both datasets exhibit sufficient topical diversity and a rich distribution of physics questions across multiple subfields, ensuring that we can begin with a small, well-defined subset of topics and later expand. We generated sample questions and answers from both benchmarks and tested them with an LLM (`gpt-4o-mini`) for exploratory purposes, confirming that the questions are well-understood and that the model can produce coherent answers without format or comprehension issues. Formal benchmarking of candidate models is planned for the initial phase of the main project.

Second, to gain practical skills in agent development, we completed a *LangGraph* course that covered the framework's fundamentals and included the implementation of small-scale agents such as a RAG agent, a basic calculator agent, and a document editor agent. In parallel, we developed a simple *LangChain* agent capable of performing basic arithmetic via Python command-line execution, tested on a small set of mathematical problems.

Third, we explored the *GraphRAG* library by Microsoft to understand how graph-based retrieval works in practice, how knowledge graphs can be constructed from various sources, and how information can be effectively retrieved. This exploration also informed the design of an initial ontology for our planned physics knowledge graph. At the current stage, the proposed ontology includes the following elements:

Node types:

1. **Concept/Law:** name, meaning, use cases, optional examples.
2. **Formula:** LaTeX representation, computational (SymPy) form, optional note, optional examples.
3. **Quantity/Variable:** symbol(s), plain name, standard unit(s).
4. **Constant:** name, symbol, numerical value.
5. **Topic:** chapter or section title.

Relation types:

- Concept → Formula (e.g., a law is associated with an equation).
- Concept → Concept (e.g., prerequisite or related concepts).
- Formula → Variable (e.g., a formula uses certain variables).
- Variable → Unit (e.g., m is measured in kilograms).
- Variable → Constant (e.g., typical value or standard constant).
- Concept → Topic (e.g., a concept belongs to a specific topic).

This ontology is not final and will likely be refined during the early phases of the research as we begin parsing the knowledge base and constructing the first knowledge graph.

Finally, we have produced a preliminary system architecture diagram (Figure 1) that outlines the major components of the proposed solution and the interactions between them. The design features a LangGraph-based ReAct agent, a knowledge graph retriever connected to a Neo4j backend, symbolic math solvers, and a base LLM with an internal memory state, orchestrated through a chain-of-thought (CoT) or related structured reasoning approach.

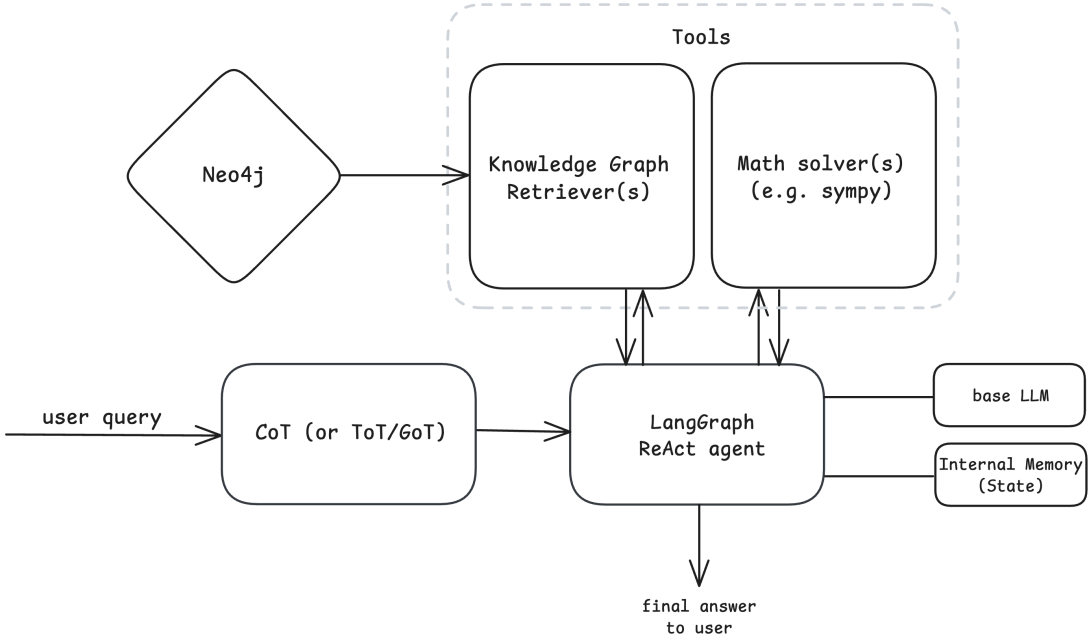

Figure 1: Preliminary system architecture showing the main components (Neo4j knowledge graph, retrievers, math solvers, LangGraph ReAct agent, and base LLM) and their interactions.

4 Other Information

4.1 Data management

Responsible data management is integral to this project. We follow FAIR principles (findable, accessible, interoperable, reusable), apply clear versioning to all artifacts (data, code, configurations), and document provenance from source to derived outputs. No personal or sensitive data are involved; the project works only with openly available scientific texts, benchmarks, and model outputs. All secrets (e.g., API keys) will be kept outside version control and managed via environment variables.

1. **Will this project involve re-using existing research data?**
   Yes. We will re-use publicly available resources: the *OpenStax University Physics* [15] XML source as the primary knowledge base, and the *SciEval* [22], *MMLU* [11] benchmarks for evaluation. To extend coverage,

we may also use supplementary sources such as the *Light and Matter* series [5] and selected *Wikipedia* pages, as well as additional benchmarks including *PHYSICS* [8], *SCIMAT-2* [14], and multimodal datasets where relevant.
Constraints on re-use: All third-party resources will be used strictly under their published terms of use. Where redistribution of raw content is restricted, we will *not* republish those raw datasets. Instead, we will release processing scripts, metadata, and, where permitted, non-reconstructible derivatives (e.g., indices, annotations, aggregated statistics). Provenance (source, version/commit, access date) will be recorded for every dataset snapshot.

2. **Will data be collected or generated that are suitable for reuse?**
Yes. The project will generate several reusable artifacts:
   - A textbook-derived **physics knowledge graph** (primarily from OpenStax, with possible enrichment from *Light and Matter* or Wikipedia) with nodes (concepts/laws, formulas, quantities/variables, constants, topics) and typed relations, exported both as a Neo4j dump and an interchange format (e.g., JSON-LD/RDF).
   - **Retrieval indices** built from openly shareable sources (e.g., vector stores over OpenStax content and supplementary sources) plus configuration for KG retrieval; if indices could enable text reconstruction beyond permitted use, we will distribute recipes/scripts to rebuild them instead.
   - **Experimental artifacts**: prompts, agent configurations, tool-call traces, and evaluation logs/metrics for benchmark runs.
   - **Processing code and ETL pipelines** for parsing XML, building the KG, creating indices, and running evaluations.

   Each artifact will include machine- and human-readable metadata (README, schema, version, hash), so others can reproduce or extend the results.

3. **After the project has been completed, how will the data be stored for the long-term and made available to third parties? Are there possible restrictions to data sharing or embargo reasons?**
Long-term storage and access:
   - *Code and pipelines* will be released in a public repository (with a permissive open-source license agreed with the supervisor) and archived for long-term findability and citation.
   - *OpenStax-derived KG* and *shareable indices/metadata* (including any enrichment from *Light and Matter* or Wikipedia, where licensing permits) will be deposited alongside the code and archived, including full provenance to the textbook snapshot used.
   - *Evaluation outputs* (metrics, traces) will be shared openly where provider terms allow. If API-based models are used and provider terms constrain redistribution of verbatim outputs, we will share scripts, prompts, and seeds to regenerate results, and publish aggregated metrics/tables.

   Restrictions/embargo:
   - Raw third-party datasets with redistribution limits (e.g., benchmark question text if restricted) will not be republished. Users will be instructed to obtain them from the original sources and use our scripts to reproduce results.
   - If needed for double-blind submission or coordinated release, derived artifacts may be placed under a short embargo until acceptance, after which they will be made public.

   Backups will be maintained during the project. At close-out, all public artifacts will be archived with stable identifiers and complete documentation to ensure independent reuse.

4.2 Motivation for choice of research group & supervisor

I selected this project because it aligns closely with my personal research interests, which include the development of LLM-based agents with planning capabilities, task-specific adaptation and fine-tuning, and the integration of knowledge bases into generative processes. The proposed research brings together multiple domains: reasoning agents, retrieval from structured sources, and symbolic/numeric verification into a single, coherent objective. This combination not only has strong potential for industrial applications but also offers a foundation for future research directions, enabling further robustness, scalability, and integration with emerging tools and frameworks.

Early on in the planning phase, I identified Professor Mykola Pechenizkiy as my preferred supervisor due to a substantial overlap in research interests and his extensive experience in guiding similar projects and theses. Professor Pechenizkiy supervises a diverse group of PhD students, many of whom explore topics directly relevant to this work, creating a stimulating research environment. Among them is Bram Schut, who recently began his PhD under Professor

Pechenizkiy's supervision. His doctoral research focuses on LLMs integrated with knowledge graphs, conducted in collaboration with ASML, where they are developing an agent capable of retrieving structured knowledge from a knowledge graph and verifying physics principles using external tools. Bram will act as a tutor for my graduation project; we have already established regular meetings to exchange findings, discuss methodologies, and address technical challenges. His guidance has also been instrumental in familiarizing me with key tools and frameworks such as Neo4j and LangGraph, ensuring a smooth start to the project's technical development.

References

[1] K. Addala, K. D. P. Baghel, D. Jain, N. Gupta, R. R. Vyalla, C. Kirtani, A. Anand, and R. R. Shah. Knowledge graphs are all you need: Leveraging kgs in physics question answering, 2025.

[2] D. Arora, H. Singh, and Mausam. Have LLMs advanced enough? a challenging problem solving benchmark for large language models. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7527–7543, Singapore, Dec. 2023. Association for Computational Linguistics.

[3] M. Besta, N. Blach, A. Kubicek, R. Gerstenberger, M. Podstawski, L. Gianinazzi, J. Gajda, T. Lehmann, H. Niewiadomski, P. Nyczyk, and T. Hoefler. Graph of thoughts: Solving elaborate problems with large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17682–17690, Mar. 2024.

[4] CK-12 Foundation. Ck-12 flexbooks and the people's physics book. `https://www.ck12.org/`, 2025. Open educational resources covering high school physics topics; accessed 2025-07-29.

[5] B. Crowell. *Light and Matter*. Open Textbook Library / Marshall University, 2010. PDF and HTML available; algebra-based introductory physics.

[6] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, D. Metropolitansky, R. O. Ness, and J. Larson. From local to global: A graph rag approach to query-focused summarization, 2025.

[7] A. A. et al. Sciphyrag - retrieval augmentation to improve llms on physics q &a. In *Big Data and Artificial Intelligence*, pages 50–63, Cham, 2023. Springer Nature Switzerland.

[8] K. Feng, Y. Zhao, Y. Liu, T. Yang, C. Zhao, J. Sous, and A. Cohan. Physics: Benchmarking foundation models on university-level physics problem solving, 2025.

[9] C. He, R. Luo, Y. Bai, S. Hu, Z. L. Thai, J. Shen, J. Hu, X. Han, Y. Huang, Y. Zhang, J. Liu, L. Qi, Z. Liu, and M. Sun. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems, 2024.

[10] J. He-Yueya, G. Poesia, R. E. Wang, and N. D. Goodman. Solving math word problems by combining language models with symbolic solvers, 2023.

[11] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding, 2021.

[12] R. Jaiswal, D. Jain, H. P. Popat, A. Anand, A. Dharmadhikari, A. Marathe, and R. R. Shah. Improving physics reasoning in large language models using mixture of refinement agents, 2024.

[13] M. G. Johannes Welbl, Nelson F. Liu. Crowdsourcing multiple choice science questions. 2017.

[14] N. Kollepara, S. K. Chatakonda, and P. Kumar. Scimat: Science and mathematics dataset, 2021.

[15] S. J. Ling, J. Sanny, and W. Moebs. *University Physics Volume 1*. OpenStax, 2016. `https://openstax.org/details/books/university-physics-volume-1`.

[16] P. Lu, S. Mishra, T. Xia, L. Qiu, K.-W. Chang, S.-C. Zhu, O. Tafjord, P. Clark, and A. Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[17] Y. Ma, Z. Gou, J. Hao, R. Xu, S. Wang, L. Pan, Y. Yang, Y. Cao, A. Sun, H. Awadalla, and W. Chen. Sciagent: Tool-augmented language models for scientific reasoning, 2024.

[18] OpenAI. Gpt-4 technical report, 2024.

[19] D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael, and S. R. Bowman. Gpqa: A graduate-level google-proof qa benchmark, 2023.

[20] C. Schiller. *Motion Mountain – The Adventure of Physics*. Self-published, 2008. Multiple volumes; conceptual and narrative style.

[21] X. Su, Y. Wang, S. Gao, X. Liu, V. Giunchiglia, D.-A. Clevert, and M. Zitnik. Kgarevion: An ai agent for knowledge-intensive biomedical qa, 2025.

[22] L. Sun, Y. Han, Z. Zhao, D. Ma, Z. Shen, B. Chen, L. Chen, and K. Yu. Scieval: A multi-level large language model evaluation benchmark for scientific research, 2024.

[23] X. Wang, Z. Hu, P. Lu, Y. Zhu, J. Zhang, S. Subramaniam, A. R. Loomba, S. Zhang, Y. Sun, and W. Wang. Scibench: Evaluating college-level scientific problem-solving abilities of large language models, 2024.

[24] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[25] C. Wu, M. Jiang, J. Li, and J. Han. A case study in bootstrapping ontology graphs from textbooks. In *Proceedings of the 3rd Conference on Automated Knowledge Base Construction (AKBC)*, 2021.

[26] D. Xu, X. Li, Z. Zhang, Z. Lin, Z. Zhu, Z. Zheng, X. Wu, X. Zhao, T. Xu, and E. Chen. Harnessing large language models for knowledge graph question answering via adaptive multi-aspect retrieval-augmentation, 2025.

[27] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023.

[28] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao. React: Synergizing reasoning and acting in language models, 2023.

[29] X. Yue, Y. Ni, K. Zhang, T. Zheng, R. Liu, G. Zhang, S. Stevens, D. Jiang, W. Ren, Y. Sun, C. Wei, B. Yu, R. Yuan, R. Sun, M. Yin, B. Zheng, Z. Yang, Y. Liu, W. Huang, H. Sun, Y. Su, and W. Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi, 2024.

[30] X. Yue, T. Zheng, Y. Ni, Y. Wang, K. Zhang, S. Tong, Y. Sun, B. Yu, G. Zhang, H. Sun, Y. Su, W. Chen, and G. Neubig. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark, 2025.

[31] Y. Zhang, Y. Ma, Y. Gu, Z. Yang, Y. Zhuang, F. Wang, Z. Huang, Y. Wang, C. Huang, B. Song, C. Lin, and J. Zhao. Abench-physics: Benchmarking physical reasoning in llms via high-difficulty and dynamic physics problems, 2025.