EduMod-LLM: A Modular Approach for Designing Flexible and Transparent Educational Assistants

Meenakshi Mittal[1], Rishi Khare[1], Mihran Miroyan[1], Chancharik Mitra[2], Narges Norouzi[1]

[1]University of California, Berkeley [2]Carnegie Mellon University
meenakshi.mittal@berkeley.edu, rishi.khare@berkeley.edu, miroyan.mihran@berkeley.edu,
cmitra@cs.cmu.edu, norouzi@berkeley.edu

Abstract

With the growing use of Large Language Model (LLM)-based Question-Answering (QA) systems in education, it is critical to evaluate their performance across individual pipeline components. In this work, we introduce EduMod-LLM, a modular function-calling LLM pipeline, and present a comprehensive evaluation along three key axes: function calling strategies, retrieval methods, and generative language models. Our framework enables fine-grained analysis by isolating and assessing each component. We benchmark function-calling performance across LLMs, compare our novel structure-aware retrieval method to vector-based and LLM-scoring baselines, and evaluate various LLMs for response synthesis. This modular approach reveals specific failure modes and performance patterns, supporting the development of interpretable and effective educational QA systems. Our findings demonstrate the value of modular function calling in improving system transparency and pedagogical alignment.

Website — https://chancharikmitra.github.io/EduMod-LLM-website/

1 Introduction

Modern LLMs have demonstrated impressive capabilities across a range of Natural Language Processing (NLP) tasks, including question-answering, knowledge-base retrieval, and summarization. A variety of methods and architectures have advanced progress in these areas including Chain-of-Thought (CoT) prompting (Wei et al. 2022b), Reinforcement Learning (RL) for reasoning (Guo et al. 2025; Jaech and OpenAI 2024), RAG (Lewis et al. 2020; Karpukhin et al. 2020), robust function calling capabilities (OpenAI 2023; Lu et al. 2023; Schick et al. 2023), and LLM token probability scoring for retrieval and automated evaluation (Lin et al. 2024; Zhang et al. 2024). However, it is unclear how these advances translate to real-world educational settings, where questions often require alignment with instructional goals, compliance with course policies, and sensitivity to student learning trajectories.

While LLMs have achieved impressive results on standardized benchmarks, such as solving complex mathematical problems in American Mathematics Competitions (AMC) and American Invitational Mathematics Examination (AIME) competitions (Lewkowycz et al. 2022; Trinh et al. 2023), their effectiveness in a live classroom environment is uncertain. Classroom questions are rarely clearly structured; they are often ambiguous, contextually grounded, and embedded in course-specific terminology. Similarly, retrieval paradigms have advanced from basic encoder-based embedding approaches (Reimers and Gurevych 2019) to fine-tuned generative models (Luo et al. 2024; Wang, Jiang, and Joty 2023) that achieve strong performance on benchmarks like Massive Text Embedding Benchmark (MTEB) (Muennighoff et al. 2022). Yet, these methods often struggle with the heterogeneous nature of educational data, ranging from textbooks, assignments, student Q&A, and course logistics. Moreover, evaluation techniques such as token probability scoring (Lin et al. 2024; Zhang et al. 2024), although promising, remain unsuitable for educational use cases where pedagogical alignment, e.g., avoiding direct answers during assessments and adhering to instructional norms of the course, is as critical as factual correctness (Dorodchi et al. 2019; Denny et al. 2024). Existing benchmarks fail to capture these instructional subtleties and leave a gap in understanding how LLM components perform under real educational constraints.

To address this gap, we introduce **EduMod-LLM**, a **modular LLM-based framework** for student QA, designed to evaluate the individual and combined impact of three critical components: *function calling strategies*, *retrieval methods*, and *generative language models*. This modular design improves transparency by enabling the precise diagnosis of failure modes and performance bottlenecks using real student questions from a large-scale university course. By systematically isolating and analyzing each module, we provide new insights into how to build more interpretable and pedagogically-aligned educational QA systems.

Our contributions in this work are threefold:

1. **LLM-as-a-Judge Evaluation Framework:** We design and validate an LLM-based evaluation module aligned with experienced Teaching Assistants (TAs). This approach provides scalable and automated assessments of response quality, factual accuracy, and pedagogical appropriateness to student questions. This evaluation represents a first-of-its-kind assessment of *LLMs and Large Reasoning Models (LRMs) as reward models for course-*

specific educational QA, distinguishing itself from general reward-model benchmarks (Li et al. 2024; Yasunaga, Zettlemoyer, and Ghazvininejad 2025; Li et al. 2024)

2. **Modular Pipeline Evaluation:** We develop a novel modular LLM pipeline that allows each component—function calling, retrieval, and response generation-to be independently and jointly evaluated. This enables fine-grained comparisons and targeted improvements in performance, establishing how modularity is especially important in educational QA.
3. **Structure-Aware Retrieval:** We introduce a novel retrieval method that reflects the hierarchical organization of educational content. By summarizing and indexing materials, such as textbooks and assignments, according to their natural structure (chapters, sections, and problem numbers), our method improves both retrieval precision and interpretability.

Our experimental results reveal several key findings. Firstly, our LLM-as-a-Judge results suggest that non-reasoning models in particular such as DeepSeek-V3 (et. al. 2024a) can reliably mimic TA evaluation standards, enabling scalable and expert-aligned quality control. Secondly, we find that our novel multi-step function-calling approaches match and outperform SoTA methods, even those with TA-designed function-calling rules. Secondly, we find that our novel structure-aware retrieval and carefully selecting GPT-4.1 as the base LLM contributes significantly to better responses, indicating the importance of specializing retrieval methods and LLMs for educational QA tasks. Collectively, these findings offer actionable design principles for building robust, transparent, and instructionally sound AI-ED systems.

2 Related Work

2.1 LLM-powered Virtual Assistants in Education

Large classroom settings often suffer from logistical inefficiencies and repeated questions, which creates a strong use case for integrating LLM into educational workflows (MacWilliam and Malan 2012; Mirhosseini, Henley, and Parnin 2023; Zamfirescu-Pereira et al. 2024; Mitra et al. 2024b; Kazemitabaar et al. 2024). Research shows that student motivation is correlated with their help-seeking behaviors (Cheong, Pajares, and Oberman 2004). Digital tools can help bridge this gap (Ali 2021; Campillo-Ferrer, Miralles-Martínez, and Sánchez-Ibáñez 2020), making LLMs such as GPT-4 (OpenAI et al. 2023), LLaMA (et. al. 2023), and Gemini (et. al. 2024b) promising candidates for educational assistance (Gan et al. 2023; Kazemitabaar et al. 2024; Zamfirescu-Pereira et al. 2024; Liu et al. 2024; Mitra et al. 2024b; Wang et al. 2024). These approaches are often unified pipelines that lack modularity and transparency necessary for iterative improvement. Educational contexts impose unique demands: responses must be factually correct, pedagogically appropriate, interpretable by instructors, and aligned with instructional policies. These constraints challenge black-box LLMs and require transparent and controllable QA systems for classroom use.

2.2 Transparent LLM Pipelines

As LLMs become more complex, understanding and controlling their behavior become a central concern. Traditional explainability methods, such as feature attribution techniques (Sundararajan, Taly, and Yan 2017; Lundberg and Lee 2017), are difficult to scale to models with billions of parameters. More recent techniques offer structural insights using task vectors or a sparse attention mechanism to attribute latent components to high-level behavior (Hendel et al. 2023; Todd et al. 2023; Huang et al. 2024; Mitra et al. 2024a). Complementary approaches in natural language include generating self-explanations or summaries (Huang et al. 2023; Zhao et al. 2023), as well as prompting strategies such as CoT prompting and In-Context Learning (ICL) (Wei et al. 2022a; Brown et al. 2020). These methods improve transparency, but few have been systematically evaluated in educational settings.

Our work builds on this line of research by focusing on the faithfulness (Jacovi and Goldberg 2020) and plausibility (Shen et al. 2022) of the output of the LLM models in the specific context of student question answering. We extend this literature by designing a modular pipeline that is both interpretable and aligned with the instructor's expectations.

3 Methods

EduMod-LLM is a modular framework for student question-answering to analyze the strengths and weaknesses of different pipeline components systematically. By isolating the **functional-calling**, **retrieval**, and **answer generation** modules, our framework (shown in Figure 1 enables detailed analysis of where failure occurs and how each component contributes to the final response quality. In the following, we describe the dataset (subsection 3.1), the architecture of each module (subsection 3.2, Section 3.3, and subsection 3.4), and our automated pedagogical quality evaluation framework (subsection 3.4).

3.1 Data

Evaluating our approach on real student questions asked in a course environment is a key factor in assessing the effectiveness of different SOTA LLM-based approaches in education. We collected historical student EdSTEM data from an upper-division undergraduate data science course at an R-1 institution in the USA, during the Spring 2024 semester. The course served 1133 undergraduates and 71 graduate students, supported by 59 staff members, covering topics ranging from data processing to machine learning and probabilistic modeling. Ground truth responses to these questions are generated via a recent SOTA, TA-in-the-loop educational assistant Edison, which was integrated into the course's EdSTEM discussion forum.

Target Course. The dataset was collected from DATA 100 (Principles and Techniques of Data Science), a single upper-division undergraduate data science course at the University of California, Berkeley. The course spans 17 weeks and covers topics from data processing to machine learning and probabilistic modeling. It includes approximately 11 weekly

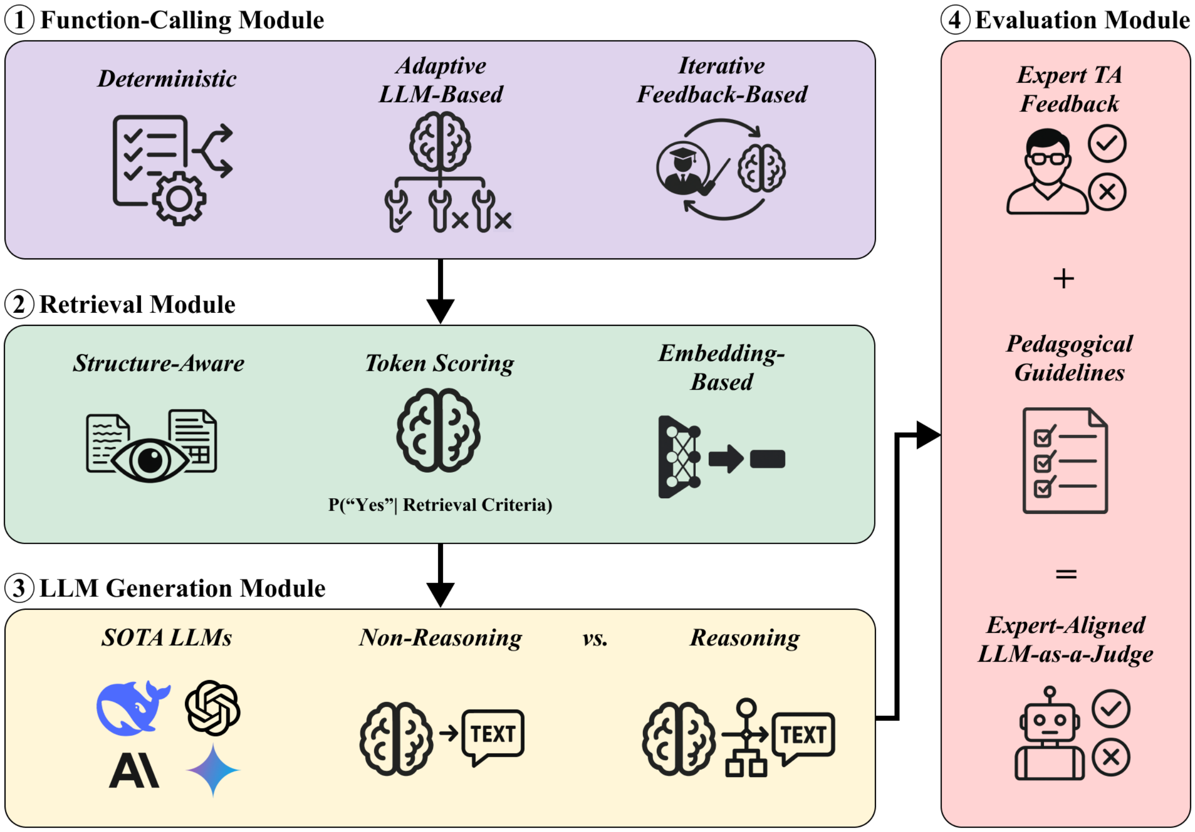

Figure 1: EduMod-LLM is a modular approach to developing an LLM-based pipeline for answering student questions. We explore design considerations in function-calling, retrieval, and LLM response generation via a high-quality and scalable evaluation module that leverages expert TA insights. This approach enables flexibility to courses and technical constraints as well as transparency for iterative improvement.

programming assignments, 13 lab exercises, and 2 exams (midterm and final).

Students use the EdSTEM discussion forum for Q&A related to conceptual understanding, assignments, and logistics throughout the semester. To clarify briefly, conceptual questions are those related to course concepts and not directly referencing assignments. Logistics questions relate to those about deadlines, course resources, and generally any other course policies.

When students submitted new questions, they could opt in to receiving assistance from Edison. TAs reviewed the questions and, when appropriate, triggered Edison to generate draft responses, which TAs could then edit and post. Our final dataset contains 1,000 student-consented questions, selected randomly from the forum, spanning the full 17 weeks to ensure temporal diversity across assignments and topics. We also collected their corresponding TA-written responses.

3.2 Function-Calling Pipeline

Educational QA requires access to multiple information sources, such as assignments, textbooks, logistics documents, and past student Q&A. Prior work has implemented RAG using these sources (Mitra et al. 2024b; Miroyan et al. 2025). We reformulate each source-specific retrieval strategy as a **function** to allow LLM to dynamically decide which to call during response generation. We provide the LLM with the following function:

- `qa_retrieval(query, top_k)`: Retrieves `top_k` similar Q&A pairs from prior semesters' EdSTEM forums.
- `textbook_retrieval(query)`: Retrieves relevant chunks from the textbook and course notes.
- `assignment_retrieval(query)`: Retrieves relevant portions of homework, labs, and projects (including any relevant solutions).
- `logistics_retrieval(query)`: Retrieves matching content from course logistics (e.g., syllabus, policies).

The LLM uses OpenAI's Function-calling (FC) API (OpenAI 2025) to choose functions, generate input arguments (e.g., `query`), and uses the returned outputs when composing its own response. Unless otherwise specified, the model can call zero or more functions in each turn.

We implement and compare the following pipelines:

- **Baseline** (`Edison`): A SoTA deterministic function-calling pipeline (Miroyan et al. 2025) that requires TA hand-designed rules based on course-specific metadata.

- **Pipeline 1** (`fc`): A default FC baseline with optional function invocation.
- **Pipeline 2** (`fc_categorize`): Prepends a student-selected category to guide the LLM to perform function selection.
- **Pipeline 3** (`fc_forced`): Requires the model to select at least one function.
- **Pipeline 4** (`fc_iterative`): Requires the model to select at least one function, then allows further function calls based on retrieved content.
- **Pipeline 5** (`fc_feedback`): After an initial function call (required) and response generation, the output is reviewed by a judging model (LLM-as-a-Judge), which provides feedback. The model then re-enters the FC loop with this feedback.
- **Pipeline 6** (`fc_multihop`): Decomposes function selection and argument generation into two LLM calls to simulate a multi-hop reasoning process.

We evaluate these pipelines along two axes: **function selection accuracy** (compared to expert-labeled ground-truth) and **final response quality** (via human and LLM-based evaluation).

3.3 Structure-Aware Retrieval

We introduce an interpretable hierarchical retrieval pipeline designed to extract logically coherent question-and-answer pairs from raw assignment documents. This approach addresses two key challenges: (1) eliminating the overhead and scalability limitations of requiring instructors to manually extract and upload individual questions and answers, and (2) improving retrieval quality beyond vector-based methods, which often operate on shallow and fixed-length chunks.

Chunking We split documents into fixed-length chunks. GPT-4o is used to detect question headers and generate a complete list of question boundaries while removing any extraneous information. These headers are then used to segment the document into coherent question-and-answer blocks.

Retrieval Next, we recursively summarize the chunks into a hierarchical (k-ary tree) structure. Each parent node summarizes its children, enabling coarse-to-fine traversal. At inference time, the LLM selects top-level summaries from a table of contents, then performs a beam-k search through the document tree to locate the most relevant content (chunk). This design improves both the relevance of the retrieved materials and the interpretability of the retrieval module, making it possible to trace which material was used and why.

3.4 Expert TA-aligned LLM-as-a-Judge Module:

To evaluate generated answers at scale, we implement an LLM-based evaluation module aligned with expert TA grading standards. Building on the taxonomy from prior work (Miroyan et al. 2025), we collaborated with two expert TAs to define detailed rubrics for three key dimensions.

> Evaluation Criteria
>
> **Factuality:** Assesses the accuracy of the information.
> **Relevance:** Evaluates alignment with (1) course material and (2) the specific student question.
> **Style:** Examines (1) clarity, (2) appropriate verbosity (concise for assignments, detailed for concepts), and (3) absence of direct solutions.

We encode these criteria into few-shot prompts that are used by multiple LLMs to simulate expert assessment. Our validation shows a high agreement between these LLM judges and the TA scores, supporting their use for large-scale evaluation while preserving pedagogical alignment.

4 Experiments and Results

We evaluate our modular LLM pipeline along three key axes: **function selection accuracy** in Subsection 4.2; **retrieval performance** in Subsection 4.2; and **response quality** in Subsection 4.5. Across all experiments, we use a consistent set of metrics:

- For FC, we report **F1 score** against expert-annotated function labels.
- For QA response quality, we report **Likert-scale scores** for **factuality** (1-5), **relevance** (1-5), and **style** (1-3), using both expert TA labels and an LLM-as-a-Judge module.
- For retrieval, we report **Recall@k** for k= $\{1, 3, 5\}$ for document-level relevance.

A subset of 180 out of 1,000 student questions was manually labeled by two expert TAs for answer quality evaluation and function selection.

4.1 LLM-as-a-Judge Evaluation

To scale evaluation beyond manual annotations, we implement an LLM-as-a-judge module aligned with TA preferences. We collaborated with two expert TAs to develop a refined rubric for factuality, relevance, and pedagogical style, and embedded these criteria into few-shot prompts for consistent scoring.

Our expert-labeled validation set contained 180 samples, selected to balance coverage across topics and question types. While this relatively small sample provides high-quality human judgments, it may not capture the full diversity of 1,000 questions.

To mitigate sampling effects, we used five random seeds for model runs and averaged results with 95% confidence intervals, which reduced variance across evaluations. Future extensions will increase the annotated subset and explore active-learning–based selection to ensure broader coverage.

We compare various LLMs in their ability to serve as judges, using two metrics: **Exact Match** with TA scores and **Mean Absolute Error (MAE)**. As shown in Figure 2, DeepSeek-v3 achieves the best alignment with expert TAs, with over 70% exact match across all categories (close to 90% for factuality) and the lowest MAE across

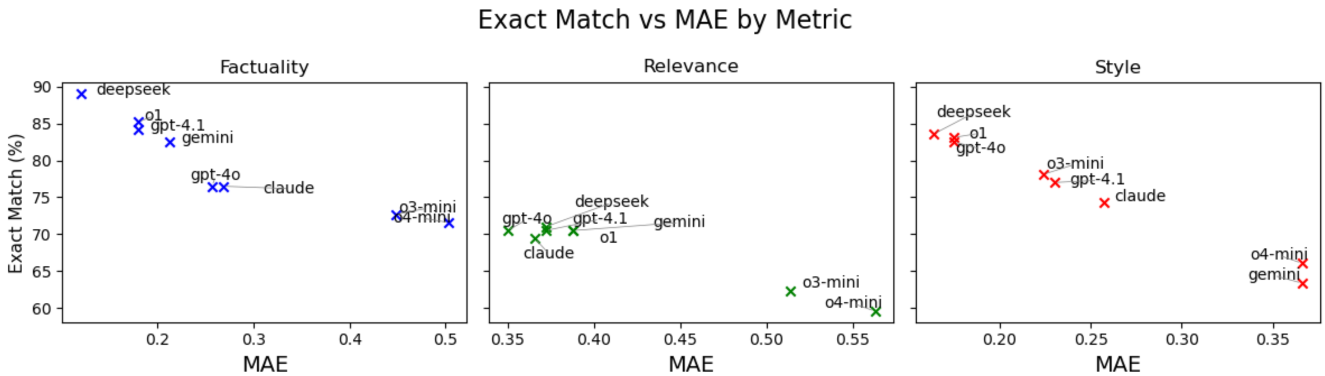

Figure 2: We report exact Match vs. MAE for LLM-as-a-Judge model across factuality, relevance, and style. DeepSeek generally achieves the best alignment with TA responses when measured across exact match and MAE.

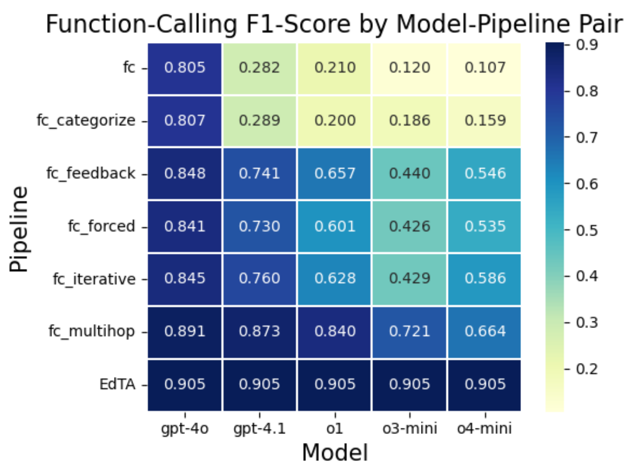

Figure 3: We evaluate function-calling F1 scores for models and pipelines. GPT-4o and GPT-4.1 achieve the highest accuracy across all models, while fc_multihop is the best . The rule-based Edison does not rely on LLM function-calling and thus achieves the same score for each model.

all categories. This consistent performance across all dimensions makes DeepSeek-v3 particularly suitable for the multi-faceted evaluation in educational contexts, where responses must be factual, relevant, and stylistically appropriate.

Other models (GPT-4o, Claude, Gemini, and o3/o4-mini) exhibit partial strength but do not match DeepSeek's consistency. This confirms that LLM-based evaluation, when guided by a domain-specific rubric, can serve as a scalable and accurate proxy for assessment in educational contexts.

4.2 Function-Calling Evaluation

We assess the accuracy of FC pipelines by comparing their selected functions to ground-truth labels established by two expert TAs. Each experiment is run five times to account for LLM variability, and we compute the mean F1 score for function selection.

As shown in Figure 3, GPT-4o consistently achieves the highest accuracy across all pipelines, with F1 scores exceeding 0.80 in most cases. More advanced pipelines such as fc_multihop outperform simpler methods across all models, which confirms the value of the use of iterative and feedback-aware tool usage. Most importantly, the deterministic Edison pipeline performs comparatively (F1-0.905). This suggests that in structured educational settings, flexible LLM-based function selection can outperform TA-designed, rule-based function selection.

4.3 Function-Calling and Response Quality

To understand how function-calling quality relates to response quality, we analyzed the relationship between function selection accuracy and response metrics. Table 2 presents a comparison of different function-calling approaches using GPT-4o (previously identified as the best-performing model for function calling) across our evaluation dimensions. The results demonstrate the impact of function-calling strategies on response quality metrics. Our automated fc_multihop demonstrates slight improvements across all three metrics of factuality, relevance, and style, importantly, matching or excedding the performance of the Edison approach (using rule-based methods developed with TA insights). This is particularly notable as the fc_multihop approach does not require pre-determined question-category information from dataset metadata, which was necessary in creating the hand-designed function-calling in Edison. The fc_iterative approach also shows strong performance, though not quite reaching the levels of fc_multihop. These findings confirm that advanced LLM-based function-calling approaches can achieve comparable response quality to hand-designed rule-based methods (which leverage question metadata) while offering significantly greater flexibility as a result. Such flexibility enables more adaptable deployments across different courses and forums.

4.4 Structure-Aware Retrieval Evaluation

We evaluate different retrieval approaches for accessing course materials relevant to student questions. For this eval-

Method	**Recall@1**	**Recall@3**	**Recall@5**
gemini-embedding-exp-03-07	0.441 ± 0.091	0.487 ± 0.091	0.523 ± 0.091
text-embedding-ada-002	0.460 ± 0.091	0.640 ± 0.090	0.721 ± 0.086
vector_gen	0.451 ± 0.091	0.676 ± 0.086	0.757 ± 0.081
hier_gen	$\mathbf{0.784 \pm 0.077}$	$\mathbf{0.874 \pm 0.059}$	$\mathbf{0.946 \pm 0.041}$

Table 1: We report retrieval performance across different methods. Recall@1, 3, and 5 are reported with 95% confidence intervals.

Model	**Experiment**	**Factuality**	**Relevance**	**Style**	**FC F1 Score**
GPT-4o	fc	4.508 ± 0.147	4.902 ± 0.091	2.918 ± 0.074	0.8055
GPT-4o	fc_categorize	4.541 ± 0.156	4.902 ± 0.091	2.934 ± 0.057	0.8066
GPT-4o	fc_feedback	4.508 ± 0.156	4.885 ± 0.074	2.902 ± 0.074	0.8481
GPT-4o	fc_forced	4.574 ± 0.147	4.885 ± 0.115	2.967 ± 0.041	0.8410
GPT-4o	fc_iterative	4.623 ± 0.131	4.918 ± 0.074	2.951 ± 0.058	0.8454
GPT-4o	Edison*	4.541 ± 0.164	4.918 ± 0.082	2.934 ± 0.057	**0.9049**
GPT-4o	fc_multihop	$\mathbf{4.689 \pm 0.115}$	$\mathbf{4.951 \pm 0.058}$	$\mathbf{2.951 \pm 0.058}$	0.8913

Table 2: For all FC pipelines GPT-4o is used as the base LLM and the quality of generated responses are evaluated across multiple axes with 95% confidence intervals. FC F1 score refers to the F1 score performance of the function calling approach. *Note: Edison requires an additional TA-designed, deterministic function-calling approach.

uation, we curated a set of 111 assignment-based student questions from course forums. To establish ground truth, we conducted human annotation to identify relevant documents for each question, treating unselected documents as negatives. This enabled us to calculate meaningful recall metrics for each retrieval method.

We compare to the following baselines:

- **Vector Retrieval + Generative Ranking** (`vector-gen`): Vector-based retrieval, retrieving a fixed 15 chunks, then ranking these chunks using a generative scoring metric. This metric uses the probability score from applying softmax to the output logit of the token ”yes” in the GPT-4o response to the question: ”Is this document relevant for answering the student question?” (Lin et al. 2024; Zhang et al. 2024) before returning the top-k chunks. This probability is used directly as an alignment score for retrieval.
- **Our LLM-based Hierarchical Retrieval** (`hier_gen`): The hierarchical generative retrieval method described earlier to retrieve the top-k chunks.
- **text-embedding-ada-001**: A vector-based retrieval pipeline based on Azure AI Search that performs hybrid search by combining vector similarity with keyword-based ranking, retrieving the top-k chunks.
- **gemini-embedding-exp-03-07**: A powerful embedding retrieval approach that leverages Gemini as a base model, enabling top-5 performance on the MTEB (Muennighoff et al. 2022) benchmark.

Table 1 presents the results of our evaluation. For each method, we computed recall rates at different cutoff thresholds (k=1, k=3, and k=5), measuring the proportion of relevant documents successfully retrieved in the top-k results.

Our structure-aware hierarchical retrieval approach (`hier_gen`) substantially outperforms all compared methods across all recall metrics, achieving 0.784 Recall@1 and 0.946 Recall@5. This represents a significant, almost 30% improvement over state-of-the-art embedding-based methods (`gemini-embedding-exp-03-07` and `text-embedding-ada-002`). These results underscore the significance of tailoring retrieval methods to the educational domain, where documents have diverse layouts and organizational structures. Despite the sophisticated representation learning in current SOTA embedding models, they still fall short when handling the unique challenges of educational content retrieval. By accounting for structural information in chunking and summarization, our approach achieves more effective retrieval, ultimately contributing to better response quality in educational QA systems. This finding suggests that educational retrieval represents a distinct problem domain requiring specialized approaches beyond general-purpose embedding models.

4.5 LLM Response Generation Evaluation

After identifying optimal FC, retrieval, and evaluation configurations, we evaluated which LLM performs best at response generation for student questions. Table 3 presents the performance of various models across our three evaluation dimensions, along with automated metrics. While all mod-

Model	Factuality	Relevance	Style
GPT-4o	4.533 ± 0.040	4.930 ± 0.018	2.980 ± 0.010
O4-Mini-High	4.574 ± 0.038	4.933 ± 0.020	2.985 ± 0.008
Claude-3.7-Sonnet	4.551 ± 0.041	4.897 ± 0.024	2.995 ± 0.005
Gemini-1.5-Pro	4.495 ± 0.041	4.920 ± 0.020	2.994 ± 0.005
DeepSeek-V3	4.586 ± 0.039	4.921 ± 0.021	2.997 ± 0.004
DeepSeek-R1	4.603 ± 0.037	4.944 ± 0.017	2.989 ± 0.007
Edison* (GPT-4o)	4.553 ± 0.070	4.912 ± 0.039	2.997 ± 0.005
GPT-4.1 (Ours)	$\mathbf{4.648 \pm 0.036}$	$\mathbf{4.955 \pm 0.014}$	$\mathbf{2.999 \pm 0.002}$

Table 3: We evaluate across 1000 samples using `fc_multihop` with GPT-4o function-calling. We report on our metrics scored using DeepSeek-V3 as a judge with 95% confidence intervals. *Note: Edison requires an additional TA-designed, deterministic function-calling approach.

els perform reasonably well, GPT-4.1 stands out with superior scores across all dimensions, particularly in relevance (4.955) and style (2.999).

This result indicates that reasoning models don't necessarily surpass non-reasoning models in student QA retrieval-augmented generation, likely due to tradeoffs between better instruction-following/context utilization abilities of non-reasoning models compared to the deliberate thinking of reasoning models. The evaluation was successfully scaled to 1,000 questions using our LLM-as-a-Judge approach, demonstrating the value of tailored automated metrics for large-scale evaluation of educational AI assistants.

5 Conclusion

We present EduMod-LLM a modular function-calling framework to improve the interpretability of LLM-generated responses to student questions on educational course discussion boards. Through comprehensive evaluation of each component, we have gained significant insights into designing effective educational QA systems.

LLM-as-a-Judge experiments show that non-reasoning models can effectively follow TA evaluation guidelines to provide reliable assessments that correlate strongly with expert human judgments. This indicates that LLM-as-a-Judge approaches can scale fine-grained and domain-specific evaluation of the quality of responses to student questions.

Our function-calling experiments reveal that correct function selection impacts response quality. Our multihop approach `fc_multihop` with GPT-4o yields responses that match or exceed those outputted by the TA-designed `Edison` pipeline in relevance, factuality, and style, with less hand-engineering and course forum metadata to deploy.

Our structure-aware retrieval module demonstrates that domain-specific adaptations substantially outperform even top-performing general-purpose embedding models from the MTEB benchmark. This 30% improvement underscores that educational content retrieval represents a distinct problem domain where the hierarchical and structured nature of course documents demands specialized approaches beyond semantic similarity.

In LLM response generation, we found that GPT-4.1 surpassed the performance of most models (particularly GPT-4o and Gemini-1.5-Pro), including specialized reasoning models. This reveals that while LRMs show extraordinary performance on mathematics benchmarks, they may struggle to surpass LLMs in contextually-rich educational environments. Finally, we find that our additions of multi-hop function calling, structure-aware retrieval, and GPT-4.1 response generation help surpass the prior art (particularly in response factuality) Edison (Miroyan et al. 2025), which leverages deterministic function-calling, vector retrieval, and GPT-4o-based pipeline.

Collectively, our findings suggest that modularity can be a valuable principle when designing educational AI systems, allowing for flexible component-wise evaluation and adaptation to the specific demands of educational contexts. The modular approach enables transparent assessment of each component's contribution to overall system performance on real student questions, moving beyond the limitations of static benchmarks. Our experiments highlight the primary importance of structure-aware retrieval, with function-calling and LLM selection serving as important but secondary factors. By independently optimizing function-calling, retrieval, and response generation, we've demonstrated superior performance compared to fixed-pipeline systems that obscure these critical distinctions.

Acknowledgements

This work is supported by the National Science Foundation Graduate Research Fellowship Program under NSF grant number: DGE2140739. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Ali, A. 2021. Using Padlet as a pedagogical tool. *Journal of Learning Development in Higher Education*, (22).

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; teusz Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. *ArXiv*, abs/2005.14165.

Campillo-Ferrer, J.-M.; Miralles-Martínez, P.; and Sánchez-Ibáñez, R. 2020. Gamification in higher education: Impact on student motivation and the acquisition of social and civic key competencies. *Sustainability*.

Cheong, Y. F.; Pajares, F.; and Oberman, P. S. 2004. Motivation and academic help-seeking in high school computer science. *Computer Science Education*.

Denny, P.; Becker, B. A.; Prather, J.; Hellas, A.; Fraser, N.; Luxton-Reilly, A.; MacNeil, S.; and Proulx, V. K. 2024. Computing Education in the Era of Generative AI. volume 67. ACM.

Dorodchi, M.; Al-Hossami, E.; Benedict, A.; Demeter, E.; and Liu, Y. 2019. Authentic Assessment of Student Learning in an Online Class: A Comparison of Automated and Manual Approaches. In *2019 IEEE Frontiers in Education Conference (FIE)*, 1–5. IEEE.

et. al., D.-A. 2024a. DeepSeek-V3 Technical Report. *ArXiv*, abs/2412.19437.

et. al., H. T. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

et. al., M. R. 2024b. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Gan, W.; Qi, Z.; Wu, J.; and Lin, J. C.-W. 2023. Large language models in education: Vision and opportunities. In *2023 IEEE international conference on big data (BigData)*.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Hendel, A.; et al. 2023. Task Vectors for Interpretable Language Model Analysis. In *Proceedings of the 2023 Conference on Neural Information Processing Systems*.

Huang, J.; et al. 2024. Sparse Attention Vectors for Interpretable LLMs. In *Proceedings of the 2024 Conference on Interpretability in AI*.

Huang, S.; Zhang, Z.; Lim, J. S.; Zhang, X.; and Zhou, Y. 2023. Can Large Language Models Explain Themselves? A Study of LLM-Generated Self-Explanations. *arXiv preprint arXiv:2310.11207*.

Jacovi, A.; and Goldberg, Y. 2020. Annotating and Measuring Faithfulness in Natural Language Explanations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jaech, A.; and OpenAI. 2024. OpenAI o1 System Card. *arXiv preprint arXiv:2412.16720*.

Karpukhin, V.; Oguz, B.; Min, S.; Lewis, P. S.; Wu, L.; Edunov, S.; Chen, D.; and Yih, W.-t. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *EMNLP (1)*, 6769–6781.

Kazemitabaar, M.; Ye, R.; Wang, X.; Henley, A. Z.; Denny, P.; Craig, M.; and Grossman, T. 2024. Codeaid: Evaluating a classroom deployment of an llm-based programming assistant that balances student and educator needs. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; Riedel, S.; and Kiela, D. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *arXiv preprint arXiv:2005.11401*.

Lewkowycz, A.; Andreassen, A.; Dohan, D.; Dyer, E.; Michalewski, H.; Ramasesh, V.; Slone, A.; Anil, C.; Schlag, I.; Gutman-Solo, T.; et al. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35: 2216–2230.

Li, L.; Wei, Y.; Xie, Z.; Yang, X.; Song, Y.; Wang, P.; An, C.; Liu, T.; Li, S.; Lin, B. Y.; Kong, L.; and Liu, Q. 2024. VL-RewardBench: A Challenging Benchmark for Vision-Language Generative Reward Models. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 24657–24668.

Lin, Z.; Li, B.; Pathak, D.; Li, J.; Xia, X.; Neubig, G.; Zhang, P.; and Ramanan, D. 2024. VQAScore: Evaluating Text-to-Visual Generation with Image-to-Text Generation. *arXiv preprint arXiv:2404.01291*.

Liu, R.; Zenke, C.; Liu, C.; Holmes, A.; Thornton, P.; and Malan, D. J. 2024. Teaching CS50 with AI: leveraging generative artificial intelligence in computer science education. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*.

Lu, P.; Peng, B.; Cheng, H.; Galley, M.; Chang, K.-W.; Wu, Y. N.; Zhu, S.-C.; and Gao, J. 2023. Chameleon: Plug-and-Play Compositional Reasoning with Large Language Models. *ArXiv*, abs/2304.09842.

Lundberg, S. M.; and Lee, S.-I. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Luo, H.; Sun, Q.; Xu, C.; Zhao, P.; Lou, J.; Tao, C.; Geng, X.; Lin, Q.; Chen, S.; and Zhang, D. 2024. NV-Embed: Multi-Stage Fine-Tuning of LLMs for Better Retrieval-Augmented Generation Over Heterogeneous Enterprise Documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

MacWilliam, T. M.; and Malan, D. J. 2012. Scaling office hours: managing live Q&A in large courses.

Mirhosseini, S.; Henley, A. Z.; and Parnin, C. 2023. What is your biggest pain point? an investigation of cs instructor obstacles, workarounds, and desires. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*.

Miroyan, M.; Mitra, C.; Jain, R.; Ranade, G.; and Norouzi, N. 2025. Analyzing Pedagogical Quality and Efficiency of LLM Responses with TA Feedback to Live Student Questions. In *Technical Symposium on Computer Science Education*.

Mitra, A.; et al. 2024a. Interpretable Sparse Attention Mechanisms for Large Language Models. In *Proceedings of the 2024 Conference on Machine Learning Interpretability*.

Mitra, C.; Miroyan, M.; Jain, R.; Kumud, V.; Ranade, G.; and Norouzi, N. 2024b. RetLLM-E: Retrieval-Prompt Strategy for Question-Answering on Student Discussion Forums. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Muennighoff, N.; Tazi, N.; Magne, L.; and Reimers, N. 2022. MTEB: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

OpenAI. 2023. Function Calling and other API updates. *OpenAI Blog*.

OpenAI. 2025. OpenAI docs: Function calling.

OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; et al. 2023. GPT-4 Technical Report.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Schick, T.; Dwivedi-Yu, J.; Dessì, R.; Raileanu, R.; Lomeli, M.; Zettlemoyer, L.; Cancedda, N.; and Scialom, T. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools. *ArXiv*, abs/2302.04761.

Shen, Y.; et al. 2022. Evaluating the Plausibility of Explanations in Large Language Models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*.

Todd, M.; et al. 2023. Function Vectors: Uncovering Latent Roles in Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Trinh, T.; Nushi, B.; Awadallah, A.; Cohen, B.; Morris, P.; Nouri, A.; and Liu, T. 2023. Solving math word problems with process-based and outcome-based feedback. *arXiv preprint arXiv:2306.01939*.

Wang, S.; Xu, T.; Li, H.; Zhang, C.; Liang, J.; Tang, J.; Yu, P. S.; and Wen, Q. 2024. Large language models for education: A survey and outlook. *arXiv preprint arXiv:2403.18105*.

Wang, Y.; Jiang, N.; and Joty, S. 2023. Improving fine-tuned question answering models with self-generated query embedding. *ACM Transactions on Information Systems*, 41(4): 1–28.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Chi, E. H.; Xia, F.; Le, Q.; and Zhou, D. 2022a. Chain of Thought Prompting Elicits Reasoning in Large Language Models. *ArXiv*, abs/2201.11903.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q. V.; and Zhou, D. 2022b. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, volume 35, 24824–24837.

Yasunaga, M.; Zettlemoyer, L. S.; and Ghazvininejad, M. 2025. Multimodal RewardBench: Holistic Evaluation of Reward Models for Vision Language Models. *ArXiv*, abs/2502.14191.

Zamfirescu-Pereira, J.; Qi, L.; Hartmann, B.; DeNero, J.; and Norouzi, N. 2024. 61A-Bot: AI homework assistance in CS1 is fast and cheap–but is it helpful? *arXiv preprint arXiv:2406.05600*.

Zhang, L.; Zheng, J.; Agarwal, N.; Kumar, P.; Cheng, Y.; Bubeck, S.; and Sun, P. 2024. Generative Verifiers: Reward Modeling as Next-Token Prediction. *arXiv preprint arXiv:2408.15240*.

Zhao, F. N. I.; et al. 2023. Towards Explainability in Large Language Models: Approaches and Challenges. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.