



Behavioral Retrieval-Augmented Tutoring: Grounding a Multi-Agent LLM Math Tutor in Annotated Tutor–Student Dialogue

Shayan Umair¹

Archbishop Mitty High School, San Jose, CA, USA

Kashif Munir²

National University of Computer and Emerging Sciences, Islamabad, Pakistan

¹Corresponding author. Affiliation: Archbishop Mitty High School, San Jose, CA, USA.

²Dr. Kashif Munir is a Professor of Computer Science at National University of Computer and Emerging Sciences (NUCES), Islamabad, Pakistan. Dr. Munir supervised the research process.

Abstract

Intelligent tutoring systems automate the adaptation of mathematics instruction for learners. This paper introduces BR-Tutor, a multi-agent mathematics tutoring architecture designed to ground a Tutor Agent in authentic pedagogical discourse rather than static textbook content. While large language models excel at mathematical problem-solving, their default disposition is to supply answers prematurely, which undermines the Socratic guided-discovery process essential to effective tutoring. To bridge this gap, the proposed system enhances an existing LangGraph-based architecture with a threshold-gated behavioral retrieval module. This module indexes 3,375 talk-move-annotated tutor-student exchanges from the Eedi QATD-2k corpus and injects the closest matching historical tutor response into the agent's prompt to model appropriate pedagogical strategies. Deployed on open-weight language models via the Groq inference platform, BR-Tutor was evaluated on a 50-problem sample from the standard MathDial benchmark. The system achieved a 94% task-success rate (Success@5) alongside a 46% answer disclosure rate (Telling@5), with both metrics rapidly saturating by the second conversational turn. Ultimately, this research highlights a persistent dissociation between raw mathematical correctness and necessary pedagogical restraint in open-weight models, emphasizing the critical role of behavioral grounding in intelligent tutoring systems.

Keywords: Intelligent tutoring systems, LLMs, multi-agent systems, RAG, dialogue tutoring, talk moves

I. Introduction

Mathematics instruction depends on adaptation to individual learners, and intelligent tutoring systems have long sought to automate that adaptation. Large language models (LLMs) make open-ended pedagogical dialogue tractable, but they also sharpen a design problem that predates them. A model optimized to be helpful supplies the answer by default, while guided-discovery tutoring including Socratic questioning, scaffolding, and hinting, requires the opposite disposition. This gap is not closed by scale alone.

The stakes of this design problem are well established. Bloom reported that one-to-one tutoring produced gains of roughly two standard deviations over conventional classroom instruction [1],

and although later meta-analysis places the effect of human tutoring closer to 0.8 standard deviations, it attributes much of that effect to the fine-grained interactivity of tutorial dialogue rather than to the delivery of answers [2]. Scaffolding, in which the tutor controls the elements of a task that exceed the learner's current capacity while leaving the rest to the learner [3], and student self-explanation elicited by tutor questioning [4] are central to that interactivity, and a period of productive struggle before instruction can itself deepen learning [5]. Recent field evidence shows the cost of abandoning these principles: in a randomized study with nearly a thousand high-school mathematics students, unconstrained access to GPT-4 improved practice performance but reduced subsequent unassisted performance, while a tutor designed with safeguards largely avoided this harm [6]. The disposition toward answer disclosure is partly a by-product of instruction tuning, which optimizes models to satisfy user requests directly [7], and surveys of LLM use in education identify it, together with student over-reliance, as a central risk [8].

Benchmarks for dialogue tutoring operationalize this tension through two complementary metrics. The first metric, Success@K, is the proportion of simulated dialogues in which the student reaches the correct answer within K turns, and the second, Telling@K, is the proportion in which the tutor discloses the answer within K turns [9]. An effective guided tutor maximizes the former while minimizing the latter. The metrics are not redundant, since the trivial strategy of stating the answer immediately maximizes both at once.

Chudziak and Kostka [10] address this problem with a multi-agent architecture built on LangGraph, comprising a Tutor Agent orchestrating specialist components through a ReAct-style loop [11], a dual-memory framework separating persistent student modeling from session context, a GraphRAG [12] retrieval layer over textbook material, and a pipeline of agents that assemble directed acyclic graphs (DAGs) for course planning. Their evaluation shows that a pedagogically-informed “Tutor Prompt” substantially outperforms a naive baseline prompt on both metrics, for both GPT-4o and GPT-4o-mini.

The architecture grounds the tutor in *what is true*, the curricular content of a textbook, but carries no information about *how an effective tutor speaks*. The discourse moves that distinguish skilled human tutoring, such as pressing a student for reasoning or revoicing a partially formed idea, are not recoverable from expository prose. This paper extends the architecture with *behavioral retrieval*. The textbook knowledge graph is replaced with an index over real, talk-move-annotated tutor–student exchanges drawn from the Eedi QATD-2k corpus [13], so that retrieval supplies pedagogical form rather than propositional content alone.

To ground this idea concretely, Table I shows an illustrative example of the kind of exchange the retrieval index is built from: a short student utterance paired with a tutor response and the discourse strategy, or “talk move” that response exemplifies. The example is paraphrased for illustration and is not a verbatim excerpt from the corpus, which is subject to its own data-sharing license.

Table I. Illustrative example of a tutor–student exchange in the style of the corpus used for behavioral retrieval (paraphrased for illustration; not a verbatim excerpt).

Speaker	Utterance
Student	I don't get how to solve $3x+5 = 20$. I tried but I keep getting the wrong number.
Tutor	No problem — let's work through it together. What's the first thing you'd want to do to get x by itself on one side?
Student	Um, maybe subtract 5?
Tutor	Exactly right. Why does subtracting 5 from both sides help you get closer to finding x ?
<i>Talk move: Press for Reasoning</i>	

Rather than answering the student's implicit request directly, the tutor turn presses the student to justify a step already taken. This is the kind of discourse move that a textbook-grounded retrieval system has no way to surface, because it exists only in records of actual tutoring interaction. Following are the contributions of the proposed work:

- An extension of the Chudziak–Kostka multi-agent architecture, adapted to open-weight models served through the Groq inference platform, together with a characterization of the structured-output failures this adaptation exposes and the prompt-level remedies that resolve them.
- *Behavioral retrieval*: A FAISS-indexed [14], threshold-gated retrieval module built over sentence embeddings [15] of annotated tutor–student dialogue, which injects real tutor responses and their talk-move labels as few-shot context into the Tutor Agent.
- An evaluation on a standard dialogue-tutoring benchmark, showing a Success@5 of 94% alongside a Telling@5 of 46% — a profile compared directly against the original architecture and against other reported baselines on the same benchmark.

The remainder of this paper is organized as follows: Section II reviews related work on intelligent tutoring systems, LLMs in education, multi-agent architectures, RAG, and discourse analysis of tutoring dialogue. Section III presents the proposed mechanism, BR-Tutor, including its architecture, its behavioral-retrieval extension, datasets, and the metrics used to evaluate it. Section IV reports the experimental setup, results, a comparison against existing mechanisms, and a discussion of the findings. Section V concludes the paper and outlines directions for future work.

II. Related Work

In this section, we review the bodies of work that motivate the proposed mechanism, namely foundational and human-modeled intelligent tutoring systems, LLMs applied to education, multi-agent architectures, RAG, mathematical reasoning in LLMs, exemplar retrieval for in-context learning, and the use of dialogue and discourse data to directly study tutoring behavior.

A. Foundational and Human-Modeled Intelligent Tutoring Systems

AutoTutor [16] simulates a human tutor through scripted natural-language dialogue built around a curriculum of deep-reasoning questions, and has produced measurable learning gains across more than a decade of controlled trials.

ASSISTments [17] takes a different approach, pairing an online homework and formative-assessment platform with a testbed for randomized controlled experiments run at the scale of thousands of students. Both systems demonstrate the value of structured tutorial interaction, but neither incorporates a large language model: their pedagogical content is authored in advance rather than generated or retrieved dynamically.

Motivating the broader case for adaptive pedagogy, van Hoeve et al. [18] study how a growth-mindset framing in secondary mathematics classrooms affects student engagement, underscoring that the manner of instructional feedback, not only its correctness, shapes learning outcomes. CLASS [19] moves closer to the LLM era, fine-tuning a language model on curated scaffolding and conversational datasets to produce SPOCK, a proof-of-concept biology tutor evaluated through expert qualitative review of its responses.

This line of work also established the student-modeling machinery that the architecture extended here inherits. Cognitive Tutors combined production-rule models of domain skill with model tracing [20] and demonstrated measurable gains in urban classrooms [21]; knowledge tracing formalized the estimation of skill mastery from observed responses [22], and was later recast with recurrent neural networks [23]. On the dialogue side, analyses of naturalistic one-to-one tutoring identified a recurring five-step frame of tutor question, student answer, tutor feedback, collaborative improvement of the answer, and a check of understanding [24], which AutoTutor then operationalized across nearly two decades of development [25]. The long-term memory component used in this paper plays the role that these student models played in classical ITS, although it stores learner attributes as natural-language records rather than as probabilistic skill estimates.

B. Large Language Models (LLMs) for Education

Several surveys chart how LLMs have been adopted across educational settings. Kumar [26] reviews LLM architectures and deployment frameworks with an eye toward emerging application domains, and Kasneci et al. [8] survey applications of LLMs specifically in education, cataloging their use for content generation, assessment, and tutoring.

Wang et al. [27] complement this with a thematic analysis of undergraduates' own perceptions of AI in personalized learning environments, and Mishra et al. [28] document adoption patterns among global clinical researchers, illustrating how quickly LLMs have entered research and academic workflows beyond the classroom.

Closer to tutoring specifically, Chen et al. [29] chain multiple LLMs together to deliver private tutoring support, and Jill Watson [30] deploys a ChatGPT-backed virtual teaching assistant at

scale, using retrieval over course materials and a skill-classifier architecture to answer real student questions in production classrooms.

Park et al. [31] present a conversation-based tutoring system that builds an explicit model of the student across a session, and Liu et al. [32] extend tutoring beyond mathematics with a multi-modal scaffolding system for language learning, both reinforcing that session-level personalization is a recurring design target across subject domains.

Several studies probe the pedagogical quality of LLM tutors directly. Tack and Piech [33] compare model-generated and human teacher replies in educational dialogue and find that the models trail human teachers in helpfulness to the student, and Wang et al. [34] show that eliciting expert decision-making, namely identifying the error, choosing a remediation strategy, and stating an intention, improves LLM responses to student mathematics mistakes. Benchmark evaluation reaches a similar conclusion at scale: across fourteen leading models, EduGuardBench finds that pedagogical incompetence is the dominant failure mode of LLMs acting as teachers [35]. Deployment studies extend this picture beyond the laboratory: Jill Watson has been scaled across online classrooms [36], and the randomized study of Bastani et al. [6] shows that a guardrailed tutor preserves learning that an unconstrained one erodes. Training-based approaches pursue the same goal from another direction: multi-horizon preference optimization refines tutor agents in a simulated teacher–student environment, rewarding both turn-level pedagogical quality and trajectory-level effectiveness [37]. Together, these results motivate treating answer restraint as a primary design objective rather than a secondary matter of style.

C. Multi-Agent Architectures

Multi-agent decomposition has emerged as a way to assign specialized reasoning to distinct components rather than overloading a single prompt. Chudziak and Kostka [10], the architecture extended in this paper, coordinate tutoring, memory, task creation, and course planning as distinct agents in a LangGraph state machine. The same coordination pattern recurs outside education: ElliottAgents [38] applies a multi-agent, natural-language-driven pipeline to stock market analysis, and a related agile-software-engineering system [39] distributes planning and coding responsibilities across LLM-augmented agents. Kostka and Chudziak [40] examine how logical reasoning, long-term memory, and inter-agent collaboration interact within multi-agent LLM systems more generally, work that directly informs the dual-memory design reused here.

Viswanathan et al. [41] apply a multi-agent approach to online education specifically, showing that agent decomposition is a recurring, cross-domain strategy for managing the complexity of adaptive, multi-component systems rather than a design choice specific to tutoring.

Beyond education, generative agents equipped with memory, reflection, and planning produce believable long-horizon behavior [42], and role-specialized agent teams structure software development through standardized operating procedures [43] and chat-chain communication between agents [44]. These systems share with the tutoring architecture a reliance on structured inter-agent messages, which makes them sensitive to the reliability of structured output discussed in Section III-B.

D. Retrieval-Augmented Generation (RAG) and Knowledge Grounding

GraphRAG [12] represents a text corpus as a knowledge graph, pre-summarizing communities of related entities so that global, corpus-level questions can be answered without exhaustively searching every passage; the original tutoring architecture adopts this approach for textbook grounding.

Dense retrieval over learned embeddings [45] and retrieval-augmented generation [46] established the now-standard pattern of conditioning a generator on passages selected by vector similarity; GraphRAG extends this pattern with graph structure aimed at global, corpus-level questions rather than narrow fact lookup. Li et al. [47] apply retrieval-augmented generation to education directly, using textbook content as context to link diagnostic assessments to curriculum goals. Across this body of work, retrieval supplies factual or curricular content; none of it addresses the separate question of which discourse strategy a tutor should use once the relevant content has been retrieved, which is the gap the present extension targets.

E. Mathematical Reasoning in Large Language Models

A tutor's ability to guide a student presupposes that the underlying model can solve the problem itself. Guan et al. [48] show that small language models can achieve strong mathematical reasoning through self-evolved deep thinking, and Trinh et al. [49] demonstrate olympiad-level geometry problem solving without human demonstrations. Neither system is a tutor, but both establish that open-weight and smaller models are increasingly viable for the mathematical reasoning a tutoring system depends on, which motivates evaluating open-weight models in the tutoring role directly rather than assuming only the largest closed models are competent problem solvers.

F. Exemplar Retrieval for In-Context Learning

Behavioral retrieval is, mechanically, a form of demonstration selection for in-context learning. Large language models can adapt to a task from a handful of demonstrations supplied in the prompt [50], and demonstrations of intermediate reasoning can further shape how a model approaches a problem [51]. Which demonstrations are supplied matters: retrievers that select demonstrations resembling the current input improve over random selection, and a retriever trained for this purpose improves further [52]. At the same time, analyses of in-context learning indicate that models draw heavily on the format, input distribution, and label space of demonstrations rather than on their exact input-output mapping [53]. The latter finding is directly relevant here, because the purpose of an injected tutor exemplar is to convey the form of a pedagogical move rather than a specific answer. It also implies a risk that the paper returns to in Section IV: a single demonstration may be too weak a signal to override a model's default response style.

G. Tutor Discourse, Talk Moves, and Dialogue Benchmarks

MathDial [9] grounds tutoring dialogues in math word problems and supplies the Success@K and Telling@K protocol used throughout this paper; Macina et al. [54] separately survey the

broader opportunities and challenges facing neural dialog tutoring. A parallel line of work studies classroom and tutoring discourse directly.

Suresh et al. [55] introduce the TalkMoves dataset, annotating whole-classroom lesson transcripts for ten discursive moves based on accountable-talk theory; because that corpus captures one-to-many classroom lectures rather than one-to-one tutoring, it is better suited to studying teacher discourse at the class level than to grounding a one-on-one tutoring agent. The Eedi QATD-2k corpus [13] instead annotates real, one-to-one, chat-based tutor–student exchanges anchored to individual mathematics problems with talk-move labels, which is the closer match to the one-on-one dialogue setting a tutoring agent operates in and is the corpus this paper builds on.

The talk-move framework itself originates in accountable-talk theory, which characterizes productive classroom discourse by its accountability to the learning community, to accurate knowledge, and to accepted standards of reasoning [56]. Computational work has since operationalized related constructs at scale, for example by measuring conversational uptake, the extent to which a teacher builds on a student's contribution, in mathematics classroom transcripts [57]. These measures motivate using talk-move labels as a retrieval key, because they describe what a tutor turn does for the student rather than what it asserts.

Table II summarizes the grounding source and principal limitation of representative systems from the categories above, relative to the mechanism proposed in Section III.

Table II. Critical analysis of representative related work relative to the proposed mechanism.

Work	Core Approach	Grounding Source	Limitation Relative to Proposed Work
AutoTutor [16]	Scripted natural-language dialogue around a curriculum of deep-reasoning questions	Hand-authored curriculum scripts	High per-domain authoring cost; strategy is fixed by the script, not retrieved from observed tutor behavior
ASSISTment s [17]	Web-based formative assessment and hint platform	Pre-authored hints and skill builders	Static, teacher-authored content; no LLM-driven dialogue generation
CLASS / SPOCK [19]	LLM fine-tuned on scaffolding and conversational datasets	Training data baked into model weights	Pedagogical strategy fixed at training time; cannot incorporate new exemplars without retraining

Jill Watson [30]	RAG over course materials with a skill-classifier architecture	Course syllabi and documents	Retrieval grounds factual course content, not pedagogical discourse strategy
GraphRAG [12]	Graph-structured community summarization for global queries	Generic text corpora	Optimized for content summarization, not turn-level pedagogical response selection
Chudziak & Kostka [10]	Multi-agent architecture with a GraphRAG-grounded Tutor Agent and dual memory	Textbook knowledge graph	Retrieval supplies curricular content but no information about effective discourse moves
BR-Tutor (proposed)	Same multi-agent architecture, extended with threshold-gated behavioral retrieval	Talk-move-annotated real tutor–student dialogue (QATD-2k)	Directly targets pedagogical form; evaluated as a single configuration without an isolating ablation (see Section IV-E)

The systems reviewed above are evaluated in markedly different ways, ranging from controlled learning-gain experiments to simulated-dialogue benchmarks. Table III summarizes these strategies and the metrics each work reports, providing the basis for the comparison carried out in Section IV-C.

Table III. Evaluation strategies and performance metrics used in related work.

Work	Evaluation Strategy	Reported Metrics
AutoTutor [16]	Controlled experiments against reading/no-intervention controls, with pre/post learning measures	Learning-gain effect size ($\approx 0.8\sigma$)
ASSISTments [17]	Multi-year, school-level randomized controlled trials	State-test score gains (≈ 10 points)
CLASS / SPOCK [19]	Expert (subject-matter) qualitative review	Ratings of factual accuracy and relevance
Jill Watson [30]	Controlled classroom A/B deployment	Response accuracy (75–97%) and teaching-presence survey scores
GraphRAG [12]	LLM-judged pairwise comparison against vector RAG	Comprehensiveness and diversity win rate

MathDial [9]	Simulated-student interactive dialogue	Success@K, Telling@K
Chudziak & Kostka [10]	Simulated-student interactive dialogue (MathDial protocol)	Success@K, Telling@K
BR-Tutor (this work)	Simulated-student interactive dialogue (MathDial protocol), reported with Wilson score intervals	Success@K, Telling@K

III. Proposed Mechanism, BR-Tutor

This section presents BR-Tutor, the mechanism proposed in this paper. It builds directly on the multi-agent architecture of Chudziak and Kostka [10], extends it with a behavioral-retrieval module, and adapts it to run on open-weight models. This section covers the architecture, the infrastructure adaptation, the behavioral-retrieval extension, the overall methodology as a flowchart, and the datasets and metrics used to evaluate the resulting system.

A. Architecture Overview

BR-Tutor is a LangGraph state graph over five functional components. The **TutorAgent** is the student-facing conversational agent and the entry point for each turn. The **MemoryAgent** is implemented as a single parameterized class serving two roles: long-term memory (LTM), holding persistent learner attributes such as prior knowledge, recurring misconceptions, and stated preferences, and working memory (WM), holding session-local state such as the active topic and current problem.

The **MemoryDispatcherAgent** determines, per turn, which memory store to read from and which to update. The **TaskCreationAgent** produces personalized practice problems from a topic and difficulty specification. A pipeline of **Course Creation agents** (Research → Planning → Step Handling → Coding) assembles DAG-structured learning paths. This architecture, and the dual-memory design in particular, follows the multi-agent coordination pattern examined more generally by Kostka and Chudziak [40].

B. Infrastructure Adaptation

Development proceeded under administratively restricted macOS privileges, precluding system-level package management. Two consequences shaped the implementation. First, Python 3.14 introduced a change to str-subclassed Enum formatting: implicit string conversion no longer yields the member value, which silently corrupted role labels in downstream code that relied on the legacy behavior until explicit `__str__` overrides were added.

Second, the GraphRAG dependency chain proved incompatible with the available interpreter and could not be installed. Rather than block on this, GraphRAG was replaced by a stub preserving the call interface, allowing the remaining agents to be developed and evaluated independently; this substitution is later made permanent by the behavioral-retrieval extension

described in Section III-C, but should be read as a constraint on the comparison this paper can offer rather than a designed ablation.

The original implementation targets OpenAI models. This work adapts the full codebase to the Groq inference platform via langchain-groq, motivated by cost and latency considerations. This adaptation was not a substitution of endpoints. Four Groq-hosted models, llama-3.3-70b-versatile, gpt-oss-120b, gpt-oss-20b, and qwen3.6-27b, were tested against LangGraph's structured tool-calling interface, and all four failed in the same characteristic way: rather than emitting a registered tool call, the model wrapped its intended JSON payload inside an invented tool name that the graph could not route. The remedy was to remove the affected agent from the tool-calling path entirely: the TaskCreationAgent now issues a plain `llm.invoke()` call carrying explicit JSON formatting instructions in the prompt, and parses the reply with `json.loads()` after stripping markdown code-fence artifacts. A related constraint applies to the MemoryDispatcherAgent: Groq's JSON mode requires the literal token `json` to appear in the prompt, and dispatch behavior was unreliable until this was satisfied explicitly.

The general principle drawn from this adaptation is narrow but practical. In multi-agent systems, the reliability of structured output is a property of the model–framework pair, not of the orchestration logic. Architectures validated against a single provider should not be assumed portable without revalidation of every structured interface.

This observation is consistent with the broader tool-use literature, in which reliable API invocation is treated as a capability that must be learned rather than assumed: Toolformer teaches a model when and how to call tools through self-supervised annotation [58], and ToolLLM constructs large instruction corpora over real-world APIs to obtain dependable function calling [59]. Open-weight models served through a third-party endpoint inherit whatever tool-calling behavior their training produced, and the failures reported here indicate that this behavior should be verified per model before an agent graph depends on it.

C. Behavioral Retrieval Extension

The GraphRAG component of the original architecture grounds responses in textbook content. BR-Tutor grounds them in tutoring behavior instead. The unit of retrieval is not a passage but an exchange: given what the student just said, what did a real tutor say next, and what pedagogical move were they making? This shifts retrieval from a content service to a behavioral prior.

The retrieval index is built from the Eedi QATD-2k corpus [13], in which individual messages carry TalkMovePrediction labels including *Press for Accuracy*, *Press for Reasoning*, *Revoicing*, *Keep Together*, and *Getting Student to Relate*. Approximately 55,000 raw message rows are grouped by InterventionId, and each tutor turn is paired with the student turn immediately preceding it. Because a large fraction of student turns are acknowledgements or otherwise non-substantive, a content filter based on mathematical and reasoning keyword detection is applied, retaining only exchanges in which the student utterance carries interpretable mathematical content. This pipeline yields approximately 3,375 usable (student utterance, tutor response, talk move) triples.

Student-side utterances are embedded with all-MiniLM-L6-v2, a sentence-transformer model built on the Sentence-BERT framework [15] and distilled with MiniLM [60], and indexed in FAISS [14] using IndexFlatIP over L_2 -normalized vectors, which makes inner-product search equivalent to cosine similarity. The index and its metadata, the paired tutor response and talk-move label for each entry, are persisted to disk so that inference requires no re-embedding of the corpus.

At each turn, the incoming student utterance is embedded and matched against the index. Retrieval is *threshold-gated*: the nearest exemplar is injected into the Tutor Agent's prompt only when cosine similarity exceeds 0.45. This threshold was set empirically by contrasting scores for deliberately off-topic control queries, which clustered near 0.38, against on-topic queries, which ranged from roughly 0.59 to 0.84. Gating matters because an ungated nearest neighbor is always returned regardless of relevance; on genuinely novel student utterances, injecting a poorly matched exemplar would supply misleading pedagogical framing rather than none.

Retrieval is invoked as a direct Python call inside the TutorAgent's response path, not as an LLM-selected tool. This is a deliberate consequence of the tool-calling instability documented in Section III-B: making injection a deterministic application-level decision removes the model's opportunity to fail at deciding when to retrieve. The cost is that the tutor cannot elect to retrieve on reasoning grounds unavailable to a similarity score; the benefit is that retrieval behavior is reproducible and independently auditable.

Figure 1 summarizes the proposed methodology end-to-end, from the student's utterance through memory dispatch, behavioral retrieval, and response generation. The Task Creation and Course Creation agents described in Section III-A operate alongside this loop, drawing on the same memory stores to generate practice problems and structured learning paths, but are omitted from the figure for clarity since they are not in the critical path of the behavioral-retrieval extension being evaluated.

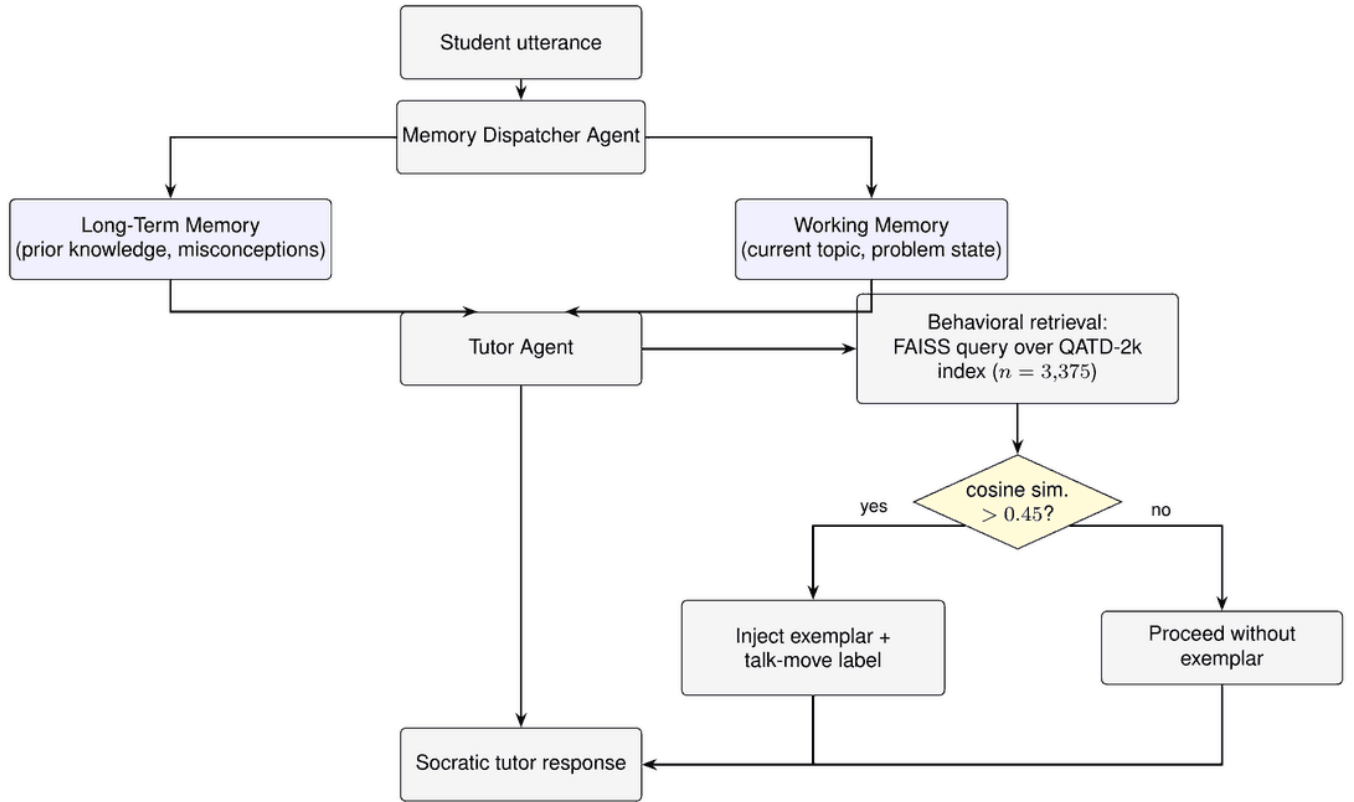


Fig. 1. Proposed methodology of BR-Tutor: the student utterance is dispatched against long-term and working memory, the Tutor Agent queries the behavioral-retrieval index, and a real tutor exemplar is injected into the prompt only when the similarity gate is cleared.

D. Design Rationale and Trade-offs

Two design decisions in BR-Tutor are deliberate trade-offs rather than incidental workarounds, and both constrain what the evaluation can show. First, retrieval is executed deterministically on every student turn and gated by similarity, instead of being exposed as a tool that the Tutor Agent may choose to call. Agentic tool selection would, in principle, let the tutor retrieve only when it judges an exemplar useful; in practice, the tool-calling failures documented in Section III-B made that decision unreliable on the evaluated models, and a retrieval step that silently fails to fire would be indistinguishable in the metrics from one that fires and is ignored. Deterministic invocation makes every retrieval event observable and loggable, which is the property that the planned ablations in Section IV-F require. Second, BR-Tutor substitutes behavioral grounding for content grounding rather than adding it alongside GraphRAG. The evaluated system therefore has no textbook grounding at all, and its mathematical correctness rests on the base model's own reasoning. This choice is appropriate for MathDial, whose word problems are self-contained and require no curricular lookup, but it means that the comparison with the original architecture is between different grounding sources, not between the presence and absence of one. A factorial design that crosses content grounding with behavioral grounding is the appropriate test of whether the two are complementary; it is specified in Table VII.

E. Datasets and Evaluation Metrics

Two datasets are used, for two distinct purposes, and each is chosen for its fit to that purpose. **MathDial** [9] grounds tutoring dialogues in multi-step math word problems and is used here as the evaluation benchmark: it supplies a simulated-student protocol and the Success@K / Telling@K metrics adopted throughout Section IV, and it is the same benchmark used to evaluate the architecture this paper extends [10], which keeps the comparison in Section IV-C on common ground.

Eedi QATD-2k [13] is used as the source corpus for behavioral retrieval, for a reason specific to the tutoring setting this system operates in: unlike the TalkMoves dataset [55], which annotates whole-classroom lecture transcripts, QATD-2k records one-to-one, chat-based tutoring exchanges anchored to individual problems, which matches the one-on-one interaction BR-Tutor's Tutor Agent conducts with a single student far more closely than a classroom transcript would.

The evaluation metrics, Success@K and Telling@K, are used elsewhere beyond their originating benchmark: Chudziak and Kostka [10] apply the same pair of metrics to evaluate their GPT-4o and GPT-4o-mini Tutor-Prompt conditions, which is what makes a direct numerical comparison in Section IV-C possible. Reporting both metrics jointly, rather than either alone, is what prevents a trivial always-answer strategy from appearing to succeed, since such a strategy would maximize Success@K and Telling@K simultaneously.

IV. Performance Evaluation and Discussion of Results

The evaluation is organized around two research questions whose scope is deliberately limited to what a single-configuration experiment can answer:

- **RQ1.** What task-success and answer-disclosure profile does BR-Tutor exhibit on MathDial when deployed on an open-weight model?
- **RQ2.** How does that profile evolve across conversational turns, and at which turn is it determined?

A third question, whether behavioral retrieval causes any difference in that profile, requires the controlled ablation described in Section IV-F and is not answered by the present data.

A. Experimental Setup

Evaluation uses the MathDial repository's simulated-interaction protocol, adapted to Groq. Adaptation required rewriting `utils.py` against the Groq SDK, replacing deprecated text-completion API calls in `gpt_baseline.py`, correcting the Python 3.14 Roles enum formatting defect described in Section III-B, and adding incremental per-problem output so that partial runs remain recoverable.

Table IV. Evaluation configuration.

Parameter	Value
-----------	-------

Benchmark	MathDial test set
Sample size	50 problems (random, seed = 42, deduplicated)
Tutor model	openai/gpt-oss-20b (Groq)
Simulated student model	openai/gpt-oss-20b (Groq)
Maximum turns per problem	5
Metrics	Success@K, Telling@K for $K = 1 \dots 5$

Sampling was verified to contain 50 distinct problems with no duplicates. Both conversational roles are played by the same model, a configuration returned to in Section IV-E.

B. Results

Table V. Success@K and Telling@K on a 50-problem MathDial sample. Counts are out of 50; intervals are 95% Wilson score intervals [61].

K	Success@K	95% CI	Telling@K	95% CI
1	90% (45/50)	[78.6, 95.7]	42% (21/50)	[29.4, 55.8]
2	94% (47/50)	[83.8, 97.9]	46% (23/50)	[33.0, 59.6]
3	94% (47/50)	[83.8, 97.9]	46% (23/50)	[33.0, 59.6]
4	94% (47/50)	[83.8, 97.9]	46% (23/50)	[33.0, 59.6]
5	94% (47/50)	[83.8, 97.9]	46% (23/50)	[33.0, 59.6]

Two features of Table V are worth isolating. First, both curves are flat from $K = 2$ onward: forty-five of the fifty dialogues resolve correctly on the first turn and only two more resolve thereafter; likewise, twenty-one dialogues contain answer revelation at the first turn and only two more acquire it subsequently. No dialogue changes state after the second turn, so the five-turn budget is not binding — the system's behavior is effectively determined by its opening moves. Second, the system achieves high correctness and substantial answer revelation simultaneously: a Telling@5 of 46% means that in nearly half of all dialogues the tutor supplied or confirmed the answer rather than eliciting it, and because 42 of those 46 percentage points are already present at $K = 1$, disclosure is predominantly an opening-turn phenomenon rather than a gradual concession made after the simulated student fails to progress.

C. Comparison with Existing Mechanisms

A frequent and fair criticism of single-system evaluations is the absence of a direct comparison against the most relevant existing mechanisms. Table VI addresses this directly, situating BR-Tutor against the baselines reported in the original MathDial paper [9] and against the architecture this work extends [10].

Table VI. Comparison against reported baselines on the MathDial benchmark. Rows above the midrule are as reported in the cited source; the bottom row is this work.

System	Success@5	Telling@5	Success@10	Telling@10
NextStep baseline [9]	25%	0%	25%	0%
ChatGPT baseline [9]	29%	16%	53%	32%
Flan-T5 _{780M} , fine-tuned [9]	30%	<1%	39%	4%
Human ground truth [9]	59%	<1%	82%	4%
Chudziak & Kostka, GPT-4o Tutor-Prompt* [10]	—	—	≈90–98%	≈5–12%
Chudziak & Kostka, GPT-4o-mini Tutor-Prompt* [10]	—	—	≈90–98%	≈1–4%
BR-Tutor (this work), gpt-oss-20b	94%	46%	N/A [†]	N/A [†]

*Values are visual estimates read from Figure 4 of Chudziak and Kostka [10], which reports these results only as a plotted curve and does not give numeric values in the text. Reading a percentage off a small printed figure carries inherent imprecision on the order of several percentage points, so these bands should be read as an approximate range, not an exact reproduction of the original authors' results; per- K values at $K = 5$ are not separately distinguishable in the published figure and are therefore omitted rather than estimated. [†]This evaluation is capped at $K = 5$ turns; see Section IV-E.

Three observations follow from this comparison. First, BR-Tutor's Success@5 of 94% exceeds every baseline in the table at $K = 5$, including the human ground truth (59% at the same K), and is within range of the ground truth's own Success@10 of 82%, reached five turns earlier. Second, BR-Tutor's Telling@5 of 46% is far above the ChatGPT baseline's Telling@5 of 16% and substantially above the Chudziak–Kostka GPT-4o and GPT-4o-mini conditions, whose Telling@10 values remain concentrated in the single-to-low-double-digit range even under the widened estimate. Third, this comparison is indicative rather than controlled: the rows differ in base model, sample size and composition, maximum turn count, and simulated-student configuration, and the confidence intervals in Table V are wide enough to accommodate several percentage points of variation on top of the graph-reading uncertainty already noted in the table. The comparison is offered to situate the result against the most relevant existing mechanisms, as is appropriate practice, not to claim a controlled head-to-head ranking.

D. Discussion

The central observation is that correctness and pedagogical restraint dissociate. On raw task success, a 20B open-weight model deployed on commodity inference infrastructure is competitive with the closed models reported in the original study, and converges faster. On

answer revelation it is not close. This is a useful result precisely because it is not a uniform improvement or regression: it localizes the difference between model families to discourse style rather than mathematical capability.

Several mechanisms plausibly contribute, and the present data distinguish them only partially. The strongest candidate is the front-loading pattern visible in the $K = 1$ figures. A tutor that disclosed answers as a fallback — after the student stalled — would produce a Telling curve that rose across turns; this system's curve does not rise, so disclosure is an opening-move disposition. This is consistent with instruction-tuned helpfulness overriding the prompt's Socratic framing on the first response, before any evidence of student difficulty exists to scaffold against.

The behavioral-retrieval module is designed to counteract exactly this disposition by supplying a concrete, real tutor response as few-shot context. The present evaluation cannot establish whether it does so, because only the extended configuration was run; no ablation isolating retrieval's contribution is available. This is stated plainly rather than attributing the observed profile to the extension: the result characterizes the system as built, and does not decompose it.

A secondary consideration is the interaction between threshold gating and the saturation pattern. Because the system's behavior is fixed by turn two, retrieval has at most two opportunities to influence a dialogue, and on the opening turn the student utterance is typically a restatement of the problem rather than a diagnostic expression of difficulty — the condition under which the QATD-2k exemplars are most informative. Retrieval may therefore be arriving after the turn at which it would matter most.

This interpretation is consistent with prior evidence. Instruction tuning rewards direct satisfaction of the user's request [7], and demonstration studies suggest that in-context examples mostly transmit format rather than overriding such dispositions [53]; a single retrieved exemplar is therefore a weak counterweight to a strongly trained default. The practical implication, in line with the guardrail findings of Bastani et al. [6], is that answer restraint is better enforced by explicit first-turn constraints on what the tutor may reveal, or by optimizing the tutor directly for pedagogical behavior [37], with retrieval supplying the form of the pedagogical move rather than being relied on to suppress disclosure by itself.

E. Limitations

- **Single condition.** Only the extended system was evaluated. Without a matched run of the unextended architecture on identical infrastructure, no claim about the effect of behavioral retrieval on Success@K or Telling@K is supported by these data.
- **Sample size.** Fifty problems yield Wilson intervals spanning roughly 14 points for Success and 27 points for Telling. Comparisons at the level of a few percentage points are not resolvable at this n .
- **Self-play.** The same model plays tutor and student. Language models are known to recognize and favor their own generations [62], and model-based evaluation more generally carries systematic biases [63]; a simulated student that shares the tutor's representations may therefore resolve terse or ambiguous tutor turns that a human learner or a different student model would not, inflating Success@K. A recent systematic evaluation of student simulation on real mathematics tutoring dialogues finds that

prompting-based simulators, the approach used here, perform poorly relative to fine-tuned alternatives [64], reinforcing this caveat.

- **Turn budget.** A five-turn cap prevents direct comparison with results reported at $K = 10$, although the observed saturation at $K = 2$ suggests the truncation is not materially constraining for this system.
- **No GraphRAG comparison.** GraphRAG could not be installed in the available environment. Behavioral retrieval therefore substitutes for a stubbed component rather than competing against a functioning one.
- **Heuristic filtering.** Corpus construction relies on keyword-based detection of content-rich student utterances, which may exclude valid pedagogical exchanges and admit noise. The 0.45 similarity threshold was calibrated on a small set of control comparisons and is not established as transferable.
- **Simulated learners.** As in the original study, no evaluation with human students was conducted. Metric performance in simulation is not evidence of learning gains.

F. Planned Ablation and Evaluation Protocol

The limitations above are addressable with experiments that the present infrastructure already supports. Table VII specifies them so that the contribution of each component can be isolated on the same 50-problem sample, inference endpoint, and seed, with Wilson intervals reported for every arm. The protocol is stated here as a commitment rather than as a result: none of its arms other than A3 has been run for this paper.

Table VII. Planned experimental protocol addressing the evaluation gaps identified in Section IV-E. All arms use the MathDial sample, seed, and metrics of Table IV.

Experiment	Configuration	Question addressed
A1: Unextended baseline	Multi-agent tutor with Tutor Prompt only; retrieval disabled	Does behavioral retrieval change Success@K or Telling@K at all?
A2: Ungated retrieval	Nearest exemplar always injected ($\tau = 0$)	Does the similarity gate help, hurt, or make no difference?
A3: Gated BR-Tutor	Current system ($\tau = 0.45$)	Reference condition, reported in Table V
S1: Threshold sweep	$\tau \in \{0.20, 0.30, \dots, 0.80\}$, with retrieval rate logged per turn	How sensitive are the metrics to τ , and how often does retrieval fire?
S2: Heterogeneous students	Same tutor against student simulators from other model families	How much of Success@K is attributable to self-play?

S3: Extended horizon	Turn limit raised to $K = 10$	Does saturation at $K = 2$ persist, and how does the system compare at $K = 10$?
S4: First-turn restraint	Explicit first-turn prohibition on confirming final answers; retrieval keyed on the problem statement	Can opening-turn disclosure be reduced without lowering Success@K?
F1: Content $\times$ behavior	2×2 factorial: GraphRAG on/off $\times$ behavioral retrieval on/off	Are content and behavioral grounding complementary or competing?
H1: Human review	Expert educators rate a transcript sample for accuracy, Socratic quality, and tone	Do the automated metrics agree with expert judgment?

V. Conclusion and Future Work

This paper has extended a multi-agent mathematics tutoring architecture with behavioral retrieval over talk-move-annotated tutor–student dialogue, and has adapted the resulting system, BR-Tutor, to run on open-weight infrastructure. On a 50-problem MathDial sample, the system has reached a Success@5 of 94% and a Telling@5 of 46%, with both metrics saturating by the second turn, and this profile has been compared directly against the architecture's original closed-model results and against other reported MathDial baselines.

The immediate priority for future work is the ablation this study lacks: a matched run with retrieval disabled, and a third condition with gating removed, would separate the contribution of the exemplars from that of the threshold. Second, given that disclosure is concentrated in the opening turn, an intervention targeting the first response specifically, for instance, retrieving on the problem statement rather than the student's initial utterance, is a more direct remedy than turn-agnostic injection. Third, restoring a functional GraphRAG component would permit a head-to-head comparison between content grounding and behavioral grounding, and would allow testing whether the two are complementary. Finally, evaluation against a distinct simulated-student model, and ultimately with human learners, is necessary before any of these metrics can be read as evidence about instruction.

Beyond its specific numbers, the study offers a concrete observation for designers of LLM tutors: on this benchmark, correctness and restraint are separable properties, and restraint is decided in the opening turn. Retrieval of authentic tutor behavior is a plausible mechanism for shaping that turn, and the protocol in Table VII provides the means to establish whether it does.

References

- [1] Bloom, B. S.: The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher* **13**(6), 4–16 (1984).
<https://doi.org/10.3102/0013189X013006004>

- [2] VanLehn, K.: The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist* **46**(4), 197–221 (2011).
<https://doi.org/10.1080/00461520.2011.611369>
- [3] Wood, D., Bruner, J. S., Ross, G.: The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry* **17**(2), 89–100 (1976).
<https://doi.org/10.1111/j.1469-7610.1976.tb00381.x>
- [4] Chi, M. T. H., Siler, S. A., Jeong, H., Yamauchi, T., Hausmann, R. G.: Learning from human tutoring. *Cognitive Science* **25**(4), 471–533 (2001).
https://doi.org/10.1207/s15516709cog2504_1
- [5] Kapur, M.: Productive failure. *Cognition and Instruction* **26**(3), 379–424 (2008).
<https://doi.org/10.1080/07370000802212669>
- [6] Bastani, H., Bastani, O., Sungu, A., Ge, H., Kabakcı, Ö., Mariman, R.: Generative AI without guardrails can harm learning: Evidence from high school mathematics. *Proceedings of the National Academy of Sciences* **122**(26), e2422633122 (2025).
<https://doi.org/10.1073/pnas.2422633122>
- [7] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., et al.: Training language models to follow instructions with human feedback. In: *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, pp. 27730–27744 (2022).
- [8] Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., et al.: ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences* **103**, 102274 (2023).
<https://doi.org/10.1016/j.lindif.2023.102274>
- [9] Macina, J., Daheim, N., Chowdhury, S. P., Sinha, T., Kapur, M., Gurevych, I., Sachan, M.: MathDial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems. In: *Findings of the ACL: EMNLP 2023*, pp. 5602–5621 (2023).
<https://doi.org/10.18653/v1/2023.findings-emnlp.372>
- [10] Chudziak, J. A., Kostka, A.: AI-Powered Math Tutoring: Platform for Personalized and Adaptive Education. *arXiv preprint arXiv:2507.12484* (2025). <https://arxiv.org/abs/2507.12484>
- [11] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., Cao, Y.: ReAct: Synergizing reasoning and acting in language models. In: *Proc. 11th Int. Conf. Learning Representations (ICLR 2023)* (2023).
- [12] Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Ness, R. O., Larson, J.: From local to global: A graph RAG approach to query-focused summarization. *Microsoft Research technical report, arXiv:2404.16130* (2024).
<https://arxiv.org/abs/2404.16130>

- [13] Zent, M., Smith, D., Woodhead, S.: PIIvot: A lightweight NLP anonymization framework for question-anchored tutoring dialogues. In: Proc. 2025 Conf. Empirical Methods in Natural Language Processing (EMNLP 2025), pp. 27479–27488. ACL, Suzhou, China (2025).
- [14] Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* **7**(3), 535–547 (2021). <https://doi.org/10.1109/TBDATA.2019.2921572>
- [15] Reimers, N., Gurevych, I.: Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In: Proc. 2019 Conf. Empirical Methods in Natural Language Processing and 9th Int. Joint Conf. Natural Language Processing (EMNLP-IJCNLP), pp. 3982–3992 (2019). <https://doi.org/10.18653/v1/D19-1410>
- [16] Graesser, A. C., Wiemer-Hastings, K., Wiemer-Hastings, P., Kreuz, R.: AutoTutor: A simulation of a human tutor. *Cognitive Systems Research* **1**(1), 35–51 (1999). [https://doi.org/10.1016/S1389-0417\(99\)00005-4](https://doi.org/10.1016/S1389-0417(99)00005-4)
- [17] Heffernan, N., Heffernan, C.: The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education* **24**, 470–497 (2014). <https://doi.org/10.1007/s40593-014-0024-x>
- [18] van Hoeve, M., Doorman, M., Veldhuis, M.: Fostering a growth mindset in secondary mathematics classrooms in the Netherlands. *Research in Mathematics Education*, 1–22 (2023). <https://doi.org/10.1080/14794802.2023.2241433>
- [19] Sonkar, S., Liu, N., Mallick, D. B., Baraniuk, R. G.: CLASS: A design framework for building intelligent tutoring systems based on learning science principles. In: Findings of the ACL: EMNLP 2023, pp. 1941–1961 (2023). <https://doi.org/10.18653/v1/2023.findings-emnlp.130>
- [20] Anderson, J. R., Corbett, A. T., Koedinger, K. R., Pelletier, R.: Cognitive tutors: Lessons learned. *Journal of the Learning Sciences* **4**(2), 167–207 (1995). https://doi.org/10.1207/s15327809jls0402_2
- [21] Koedinger, K. R., Anderson, J. R., Hadley, W. H., Mark, M. A.: Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education* **8**, 30–43 (1997).
- [22] Corbett, A. T., Anderson, J. R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction* **4**(4), 253–278 (1994). <https://doi.org/10.1007/BF01099821>
- [23] Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., Sohl-Dickstein, J.: Deep knowledge tracing. In: *Advances in Neural Information Processing Systems* 28 (NeurIPS 2015) (2015).
- [24] Graesser, A. C., Person, N. K., Magliano, J. P.: Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology* **9**(6), 495–522 (1995). <https://doi.org/10.1002/acp.2350090604>

- [25] Nye, B. D., Graesser, A. C., Hu, X.: AutoTutor and family: A review of 17 years of natural language tutoring. *International Journal of Artificial Intelligence in Education* **24**(4), 427–469 (2014).
- [26] Kumar, P.: Large language models (LLMs): Survey, technical frameworks, and future challenges. *Artificial Intelligence Review* **57** (2024). <https://doi.org/10.1007/s10462-024-10888-y>
- [27] Wang, X., Xu, X., Zhang, Y., Hao, S., Jie, W.: Exploring the impact of artificial intelligence application in personalized learning environments: Thematic analysis of undergraduates' perceptions in China. *Humanities and Social Sciences Communications* **11**(1) (2024). <https://doi.org/10.1057/s41599-024-04168-x>
- [28] Mishra, T., Sutanto, E., Rossanti, R., Pant, N., Ashraf, A., Raut, A., Uwabareze, G., Oluwatomiwa, A., Zeeshan, B.: Use of large language models as artificial intelligence tools in academic research and publishing among global clinical researchers. *Scientific Reports* **14**(1) (2024). <https://doi.org/10.1038/s41598-024-81370-6>
- [29] Chen, Y., Ding, N., Zheng, H. T., Liu, Z., Sun, M., Zhou, B.: Empowering private tutoring by chaining large language models. In: *Proc. 33rd ACM Int. Conf. Information and Knowledge Management (CIKM '24)*, pp. 354–364 (2024). <https://doi.org/10.1145/3627673.3679665>
- [30] Taneja, K., Maiti, P., Kakar, S., Guruprasad, P., Rao, S., Goel, A. K.: Jill Watson: A virtual teaching assistant powered by ChatGPT. In: *Artificial Intelligence in Education (AIED 2024)*, Part I, pp. 324–337. Springer (2024).
- [31] Park, M., Kim, S., Lee, S., Kwon, S., Kim, K.: Empowering personalized learning through a conversation-based tutoring system with student modeling. In: *Ext. Abstracts CHI Conf. Human Factors in Computing Systems (CHI EA '24)*, pp. 1–10 (2024). <https://doi.org/10.1145/3613905.3651122>
- [32] Liu, Z., Yin, S. X., Lee, C., Chen, N. F.: Scaffolding language learning via multi-modal tutoring systems with pedagogical instructions. In: *2024 IEEE Conf. Artificial Intelligence (CAI)*, pp. 1258–1265 (2024).
- [33] Tack, A., Piech, C.: The AI teacher test: Measuring the pedagogical ability of Blender and GPT-3 in educational dialogues. In: *Proc. 15th Int. Conf. Educational Data Mining (EDM 2022)* (2022).
- [34] Wang, R. E., Zhang, Q., Robinson, C., Loeb, S., Demszky, D.: Bridging the novice-expert gap via models of decision-making: A case study on remediating math mistakes. In: *Proc. 2024 Conf. North American Chapter of the ACL (NAACL 2024)* (2024).
- [35] Jiang, Y., Zhang, M., Yin, X., Jin, S., Lu, S., Ying, Z., Yu, Z., Kong, X.: EduGuardBench: A holistic benchmark for evaluating the pedagogical fidelity and adversarial safety of LLMs as simulated teachers. *Proceedings of the AAAI Conference on Artificial Intelligence* **40**(37), 31356–31364 (2026). <https://doi.org/10.1609/aaai.v40i37.40399>

- [36] Kakar, S., Maiti, P., Taneja, K., Nandula, A., Nguyen, G., Zhao, A., Nandan, V., Goel, A. K.: Jill Watson: Scaling and deploying an AI conversational agent in online classrooms. In: Intelligent Tutoring Systems (ITS 2024), Part I, pp. 78–90. Springer (2024).
- [37] Shi, X., Zhang, C., Zhu, Y., Zhang, X., Luo, Y.: Beyond pedagogical principles: Multi-horizon preference optimization for efficient Socratic tutoring. In: Proc. 64th Annual Meeting of the ACL (ACL 2026), Long Papers, pp. 11289–11306. San Diego, CA, USA (2026).
<https://aclanthology.org/2026.acl-long.518/>
- [38] Chudziak, J. A., Wawer, M.: ElliottAgents: A natural language-driven multi-agent system for stock market analysis and prediction. In: Proc. 38th Pacific Asia Conf. Language, Information and Computation (PACLIC 38), pp. 961–970. Tokyo, Japan (2024).
- [39] Chudziak, J. A., Cinkusz, K.: Towards LLM-augmented multiagent systems for agile software engineering. In: Proc. 39th IEEE/ACM Int. Conf. Automated Software Engineering (ASE '24), pp. 2476–2477 (2024).
- [40] Kostka, A., Chudziak, J. A.: Synergizing logical reasoning, long-term memory, and collaborative intelligence in multi-agent LLM systems. In: Proc. 38th Pacific Asia Conf. Language, Information and Computation (PACLIC 38). Tokyo, Japan (2024).
- [41] Viswanathan, N., Meacham, S., Adedoyin, F. F.: Enhancement of online education system by using a multi-agent approach. Computers and Education: Artificial Intelligence **3**, 100057 (2022). <https://doi.org/10.1016/j.caeai.2022.100057>
- [42] Park, J. S., O'Brien, J., Cai, C. J., Morris, M. R., Liang, P., Bernstein, M. S.: Generative agents: Interactive simulacra of human behavior. In: Proc. 36th Annual ACM Symp. User Interface Software and Technology (UIST '23), pp. 1–22 (2023).
<https://doi.org/10.1145/3586183.3606763>
- [43] Hong, S., Zhuge, M., Chen, J., Zheng, X., Cheng, Y., Zhang, C., et al.: MetaGPT: Meta programming for a multi-agent collaborative framework. In: Proc. 12th Int. Conf. Learning Representations (ICLR 2024) (2024).
- [44] Qian, C., Liu, W., Liu, H., Chen, N., Dang, Y., Li, J., et al.: ChatDev: Communicative agents for software development. In: Proc. 62nd Annual Meeting of the ACL (ACL 2024), Long Papers (2024).
- [45] Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.: Dense passage retrieval for open-domain question answering. In: Proc. 2020 Conf. Empirical Methods in Natural Language Processing (EMNLP 2020), pp. 6769–6781 (2020).
<https://doi.org/10.18653/v1/2020.emnlp-main.550>
- [46] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., et al.: Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Advances in Neural Information Processing Systems 33 (NeurIPS 2020), pp. 9459–9474 (2020).

- [47] Li, X., Henriksson, A., Duneld, M., Nouri, J., Wu, Y.: Supporting teaching-to-the-curriculum by linking diagnostic tests to curriculum goals: Using textbook content as context for retrieval-augmented generation with large language models. In: Artificial Intelligence in Education (AIED 2024), pp. 118–132. Springer (2024).
https://doi.org/10.1007/978-3-031-64302-6_9
- [48] Guan, X., Zhang, L. L., Liu, Y., Shang, N., Sun, Y., Zhu, Y., Yang, F., Yang, M.: rStar-Math: Small LLMs can master math reasoning with self-evolved deep thinking. In: Proc. 42nd Int. Conf. Machine Learning (ICML 2025), PMLR, vol. 267, pp. 20640–20661 (2025).
- [49] Trinh, T. H., Wu, Y., Le, Q. V., He, H., Luong, T.: Solving olympiad geometry without human demonstrations. *Nature* **625**(7995), 476–482 (2024).
<https://doi.org/10.1038/s41586-023-06747-5>
- [50] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., et al.: Language models are few-shot learners. In: Advances in Neural Information Processing Systems 33 (NeurIPS 2020), pp. 1877–1901 (2020).
- [51] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q. V., Zhou, D.: Chain-of-thought prompting elicits reasoning in large language models. In: Advances in Neural Information Processing Systems 35 (NeurIPS 2022), pp. 24824–24837 (2022).
- [52] Rubin, O., Herzig, J., Berant, J.: Learning to retrieve prompts for in-context learning. In: Proc. 2022 Conf. North American Chapter of the ACL (NAACL 2022) (2022).
- [53] Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., Zettlemoyer, L.: Rethinking the role of demonstrations: What makes in-context learning work? In: Proc. 2022 Conf. Empirical Methods in Natural Language Processing (EMNLP 2022) (2022).
- [54] Macina, J., Daheim, N., Wang, L., Sinha, T., Kapur, M., Gurevych, I., Sachan, M.: Opportunities and challenges in neural dialog tutoring. In: Proc. 17th Conf. European Chapter of the ACL (EACL 2023), pp. 2357–2372 (2023). <https://doi.org/10.18653/v1/2023.eacl-main.173>
- [55] Suresh, A., Jacobs, J., Harty, C., Perkoff, M., Martin, J. H., Sumner, T.: The TalkMoves dataset: K-12 mathematics lesson transcripts annotated for teacher and student discursive moves. In: Proc. 13th Language Resources and Evaluation Conf. (LREC 2022), pp. 4654–4662 (2022).
- [56] Michaels, S., O'Connor, C., Resnick, L. B.: Deliberative discourse idealized and realized: Accountable talk in the classroom and in civic life. *Studies in Philosophy and Education* **27**(4), 283–297 (2008).
- [57] Demszky, D., Liu, J., Mancenido, Z., Cohen, J., Hill, H., Jurafsky, D., Hashimoto, T.: Measuring conversational uptake: A case study on U.S. math education. In: Proc. 59th Annual Meeting of the ACL and 11th Int. Joint Conf. Natural Language Processing (ACL-IJCNLP 2021) (2021).

- [58] Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., Scialom, T.: Toolformer: Language models can teach themselves to use tools. In: Advances in Neural Information Processing Systems 36 (NeurIPS 2023) (2023).
- [59] Qin, Y., Liang, S., Ye, Y., Zhu, K., Yan, L., Lu, Y., et al.: ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In: Proc. 12th Int. Conf. Learning Representations (ICLR 2024) (2024).
- [60] Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., Zhou, M.: MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In: Advances in Neural Information Processing Systems 33 (NeurIPS 2020), pp. 5776–5788 (2020).
- [61] Wilson, E. B.: Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association* **22**(158), 209–212 (1927).
<https://doi.org/10.1080/01621459.1927.10502953>
- [62] Panickssery, A., Bowman, S. R., Feng, S.: LLM evaluators recognize and favor their own generations. In: Advances in Neural Information Processing Systems 37 (NeurIPS 2024) (2024).
- [63] Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., et al.: Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In: Advances in Neural Information Processing Systems 36 (NeurIPS 2023), Datasets and Benchmarks Track (2023).
- [64] Scarlatos, A., Lee, J., Woodhead, S., Lan, A.: Simulated students in tutoring dialogues: Substance or illusion? In: Proc. 64th Annual Meeting of the ACL (ACL 2026), Long Papers, pp. 42349–42385. San Diego, CA, USA (2026). <https://doi.org/10.18653/v1/2026.acl-long.1960>