CHAPTER 16

Multi-agent Architecture of an AI and Machine Learning System for Mathematics Learning and Instruction

V. I. Slyusar

Institute of Artificial Intelligence Problems, Ukraine,

E-mail: swadim@ukr.net

Abstract

In developing an advanced system for mathematics education, a technology-driven multi-agent environment is proposed as the core foundation. This environment comprises interconnected agents based on large language models (LLMs), collaboratively delivering educational mathematics services and supporting teaching staff. At the heart of this complex hierarchy is a central coordination agent, serving as a digital twin of an educational institution's administrative body. Designated the AI-administrator, it orchestrates various instructional planning and execution activities across all agents, reflecting an innovative rethinking of traditional academic roles tailored specifically to contemporary mathematics education. Within this framework, digital twins of teaching staff perform distinct roles, employing artificial intelligence to replicate institutional responsibilities. An AI-methodologist agent formulates guidelines and regulations for mathematics curricula, offering prompt responses to inquiries from instructors and students regarding academic matters. Complementing this is the 24/7 AI-curator, a distributed set of personal AI-assistants assigned individually to students. Another critical component is the AI-tutor, which continuously assesses students' mathematical understanding, identifies difficulties in real-time, and suggests personalized interventions to enhance comprehension. Inspired by specialized GPT-based tools like OpenAI's Math Solver, the proposed AI-tutor significantly extends existing capabilities by adapting content dynamically to students' cognitive abilities and available time. The AI-examiner contributes by creating tests, evaluating assignments, and generating detailed analytics. Meanwhile, the AI-lecturer delivers interactive educational content, adapting pacing and engagement to student capacities, and fostering discussions to enhance critical thinking. The agent ensemble is completed by the AI-technician and AI-psychologist. Through the implementation of the proposed concept, mathematics education becomes personalized and ascends to a new level of effectiveness, aligning with the principles and objectives of Education 5.0.

Keywords: Artificial Intelligence, Large Language Models (LLMs), Machine Learning, STEM

INTRODUCTION

Mathematics education is undergoing rapid transformation under the influence of modern digital technologies, particularly due to advances in artificial intelligence (AI). The traditional linear model of knowledge transmission is being replaced by an interactive, multi-level support system, in which the learner—ranging from school student to doctoral candidate—interacts with a suite of intelligent agents that assist in analyzing, understanding, revisiting, and creatively reinterpreting the material. In this context, a multi-agent system does not merely serve as an automation tool, but functions as an adaptive learning environment tailored to individual educational trajectories, learning pace, and level of mathematical maturity.

Numerous studies in recent years (e.g., [1–9]) confirm the effectiveness of educational AI systems, particularly in fostering critical thinking, supporting differentiated instruction, and developing self-directed learning skills. For example, article [6] provides a systematic review of research on the application of artificial intelligence in K-12 education over the period 2017–2022. The authors analyze publication trends, research topics, AI methods, technological applications, and the use of AI by students and teachers in the K-12 educational environment.

A review of current literature indicates that the most promising approach is the multi-agent architecture [10], in which agents possess different specializations. In combination with large language models (LLMs) [11 – 21] and RAG approaches [22, 23], such systems can offer a flexible balance of external support and internal learning autonomy.

This section considers how a multi-agent system can adapt to the needs of a learner (or student) at various stages of the educational vertical - from the school algebra course to specialized research in mathematical analysis, discrete mathematics, or topology. Particular attention is paid to the challenge of balancing the consumption of educational services with independent knowledge acquisition in the new context of broad AI-tool accessibility.

Specific features of the multi-agent approach for implementing a next-generation mathematics education system.

In designing an advanced system of mathematics education using AI technologies, it is proposed to base the development on the creation of a multi-agent environment in which a network of interconnected agents, built on LLMs or their simplified versions (SLMs), interact in a coordinated manner to provide mathematics education services and support the relevant teaching staff.

It is well known that the term "agent" came into common use among reinforcement learning (RL) researchers in the late 1980s [24, 25]. In these works, an agent was defined as an autonomous program that learns, through trial and error, to select optimal actions that maximize its received reward. The modern interpretation expands this definition to include language models, which operate and interact with other agents according to user instructions as part of the agent-based approach.

At the institutional level, the key agent in such a complex system hierarchy must be a coordination agent, orchestrating the functioning of all language models at various stages of instructional planning and implementation. In essence, it represents a digital twin - an analogue of the aggregate functions assigned to the administration of an educational institution as the supplier of educational services. Accordingly, this coordination agent may be termed the administrator agent. Its coordinating role in the educational process is complemented by tasks such as identifying redundancy in the functions of other agents, ensuring continuous monitoring of cyber security within the multi-agent environment, and blocking attempts at reprogramming or hacking agents.

Even this example illustrates that, in addition to replicating the functions of traditional teaching and administrative staff, it makes sense in a multi-agent environment to redistribute certain functions, rethinking conventional academic roles in line with the needs of contemporary mathematics education and the specifics of artificial intelligence technologies.

Building on this approach, let us consider the principal functions of the proposed set of digital twins of the teaching staff, which, to some extent, reproduce the institutional duties of various educational personnel based on the capabilities of language models. It should be noted that the agent names given below are largely provisional and may change, as can their functional portfolios.

For instance, the typical task of a methodologist agent is the development of methodological recommendations, guidelines, and internal regulations for mathematics education programs. This agent is expected to provide timely, around-the-clock advice to teachers and students on academic and administrative procedures, clarifying standards for instructional workload and reporting.

This is complemented by a 24/7 curator agent for student groups, which represents a distributed set of personal assistant agents for students. These assistants help to shape educational trajectories, clarify mathematical rules, and offer interactive exercises, adapting them to individual levels of mathematical knowledge and personal preferences.

The implementation of a mathematics curator agent involves a multi-stage process of constructing an interactive intelligent agent, focused on individualized support for mastering mathematical material. The main stages of this process, ordered according to the logic of agent creation, integration, and adaptation to the educational context, are considered below.

The initial stage involves preparing the knowledge base by aggregating educational content relevant to the school or university mathematics curriculum. All available sources are divided into structured (textbooks, methodological manuals, lecture notes, official curricula) and unstructured (video lectures, webinars, presentations, forum discussions) materials. It is essential to systematically organize and convert these materials into a standardized format suitable for LLM processing—for example, in the form of fragments with metadata specifying topic, difficulty level, language, and so on.

The constructed knowledge base is then used to form a RAG (Retrieval-Augmented Generation) repository [22, 23]. The RAG architecture enables the combination of a vector knowledge base (retriever), built on pre-processed educational materials, with an LLM, thus ensuring contextual answer generation based on retrieved fragments. Optimization of the RAG subsystem involves configuring vectorization (e.g., using Sentence Transformers), ranking search results, and filtering out noisy data.

In the subsequent process of training the curator agent, it is critically important to evaluate the quality of its answers to both typical and complex queries from the mathematics curriculum. For this purpose, a set of reference questions with validation answers is created and used for testing. Upon identifying errors or inaccuracies, it is necessary to make adjustments either to the generation parameters (temperature, top-k, top-p) or to the content of the vector database (reformatting fragments, adding clarifications, contextualization).

The curator agent should be implemented as part of larger educational platforms, which include LMS (Learning Management Systems), mobile applications, chatbots, or digital avatars. It is crucial to ensure asynchronous query processing - i.e., enabling students to submit questions to which the system may respond with a certain delay while performing complex generation or making clarifying requests to external sources. In particular, integration with an agent responsible for educational process planning is advisable, so that the curator agent's response takes into account the student's previous query history.

The tone of the curator agent's responses, as with other dialog agents, must correspond to the educational context: it should be friendly, precise, not overly indulgent, but encouraging. These requirements characterize a specific type of stylization (Tone of Voice, ToV) - a set of linguistic and rhetorical features that define how the message content is expressed. In other words, ToV is the style, intonation, and emotional coloring of the response, reflecting the intended audience, communication goals, and context.

In scientific communication, ToV is regulated by strict norms: terminological precision, logical consistency, impersonality, and the absence of emotionally charged words. In other contexts (e.g., marketing, journalism, dialog systems), ToV may be: formal (official, businesslike, protocol-driven); neutral (objectively informative, unemotional); friendly (informal, contact-oriented); ironic (with a degree of humor or sarcasm); motivational (intended to prompt action); or scientific (reasoned, structured, adhering to academic standards). Thus, ToV is the mood with which the text is delivered. It does not change the meaning but strongly influences the perception and credibility of the content.

In the event of a student's mistake, the response style should be correct and constructive, suggesting steps for correction. Dynamic ToV adjustment is possible depending on the category of the inquirer (primary school student, university student, instructor, parent).

Ideally, the curator agent should support a dialogic model of interaction rather than limit itself to one-time responses. This means that after each answer, the system generates clarifications, offers comprehension checks, asks guiding questions, or suggests visualizations. For example, if a student inquires about an integral, after providing an explanation, the system may offer a problem for independent solution or ask whether a geometric interpretation - possibly including a visualization, - should be explained.

Collectively, these components create a functional, adaptive agent system capable of personalizing mathematics education by integrating the strengths of LLMs, RAG, didactic dialog models, and modern pedagogy.

Another critically important component is the tutor agent. It continuously analyzes mathematical abilities and assesses students' level of mathematical proficiency in real time, identifies current problems and difficulties with specific learning topics, provides feedback, and offers targeted personalized measures to improve mastery and understanding of the material. As a prototype for such an agent, one may consider the specialized educational functionality already implemented by OpenAI based on the service for creating custom versions of GPT-4-based chatbots. An example of a corresponding mathematics tutor is the GPTs "Math Solver." Of course, in the proposed approach, the functionality of such GPT-4s should be significantly expanded by dynamically adapting educational content to the quality of knowledge acquisition and by accounting for the cognitive abilities and time resources of individual students.

Another essential agent is the examiner agent. As the name suggests, its function is to create a system of test tasks that enable assessment of student achievement, verification of written work, and oral answers to tests. Essentially, the examiner agent can be regarded as a functionally limited version of the tutor agent, more specifically focused on the formalized assessment of knowledge level. This agent is also responsible for generating and summarizing analytics to inform the instructor in accordance with established assessment indicators.

The lecturer agent delivers educational content by conducting interactive lectures, adapting the pace of new material presentation to the abilities of education service consumers. In the future, this agent will be able to engage in dialogue with students and conduct discussions that stimulate critical thinking and cognitive skills. The previously mentioned response stylization, or ToV, plays an important role in this process. To create a voice-enabled version of any agent capable of responding promptly in real time, it is necessary to consider the features of the entire speech processing pipeline: STT (Speech-to-Text) → GPT (Language Model Inference) → TTS (Text-to-Speech). The primary challenge in this regard is the speed and quality of voice interaction. The total response delay is the sum of the latencies from STT, text generation by the language model (e.g., GPT), and TTS. In typical implementations, this delay reaches 2 – 3 seconds, which is unacceptable for real-time dialogue. The goal is to reduce it to 500 – 800 ms.

Among the key approaches to solving this problem, first and foremost is the use of low-latency models such as Whisper.cpp or Deepgram with real-time streaming mode. Employing chunk-based streaming (in particular, with a chunk length of 200 – 300 ms) makes it possible to obtain partial transcription results before the phrase is finished. Integration of voice activation triggers (VAD, Voice Activity Detection) is also possible for immediate processing initiation.

Moreover, optimization at the LLM level may include: the use of quantized versions (INT4, GGUF); generation with a restricted max_new_tokens parameter; pre-contextual caching; switching to lightweight models (for example, phi-3.5-mini, Gemma-2B) or distillation versions; and the use of early-exit strategies for short queries.

To accelerate TTS, models such as Coqui TTS or Silero TTS are recommended, as they support fast inference. It is important to reduce audio fragment generation latency by employing audio pipelines (streamed audio generation), where the player starts as soon as the first segment (e.g., the initial 500 ms) is generated.

To reduce perceived latency, it is advisable to use short filler phrases such as “Interesting question...”, “One moment, calculating...”, which buy time for generating the full response. These are voiced immediately after the end of a user’s spoken phrase is detected. Before starting the dialogue, the system checks for the presence of a microphone signal, noise level, and channel activity. If no voice is detected, it issues a prompt or diagnostic message (“I cannot hear you. Please check your microphone.”). This phrase can sometimes also be used as a filler, but only rarely so as not to create an impression of poor service.

In the case of particularly heavy computations or when connecting to an external RAG system, each response begins with a template phrase serving as a time buffer. This creates the illusion of continuous dialogue, even if generation takes several seconds.

It is very important for the API interface to support streaming real-time transfer of both textual and audio data. In particular, it is recommended to use a WebSocket API with duplex communication for STT and TTS, with buffered support for partial results (partial transcription/streamed TTS).

To increase engagement in dialogue, TTS should employ emotional modulation (voice, intonation, pauses). Commercial voices (Microsoft Azure Neural Voices, ElevenLabs) or fine-tuned open-voice profiles can be integrated. As previously noted, it is important to support intonational variability depending on the topic (for example, surprise, encouragement, neutral analysis).

To minimize costs, it is preferable to run all components (STT, GPT, TTS) locally on an edge device (for example, Jetson Orin or Raspberry Pi 5 with NPU). This reduces API expenses while maintaining acceptable quality with partial cloud updates (e.g., only online queries to GPT). Alternatively, serverless architectures can be used for highly demanded components.

Thus, a high-speed voice AI agent for mathematics is implemented as an adaptive, asynchronous, multi-component system with modular optimization of each of the three key stages: recognition, text generation, and voicing (see Fig. 1), oriented towards smooth dialogic interaction and effective learning.

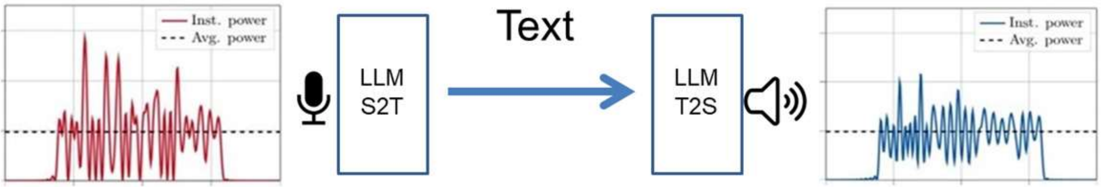

Fig. 1 Sequence of operational stages for a voice AI agent.

The ensemble of agents is completed by the technician agent and the psychologist agent. The former is responsible for overcoming technical and organizational issues, implementing automatic registration of students for courses, lectures, practical classes, and examinations. It must configure virtual classrooms and monitor the availability of educational resources. To support student motivation and psychological well-being, it is also necessary to include a psychologist agent [26] in the system. Like a human psychologist, this agent will monitor emotional states and, if needed, select appropriate psychological practices to create a comfortable learning environment.

Access to all the agents described above can be provided in a decentralized manner, allowing users to independently choose the specific agent that best matches their interests. At the same time, the administrator agent's functionality should include the capability to route queries and select the most relevant agent for the user's request. The administrator agent is therefore tasked with classifying and categorizing queries to ensure clear correspondence with the capabilities of each agent. In the instructions for the other agents, a rule must be established for redirecting queries to the administrator agent if a request is classified as outside the competence of that specific agent.

Overall, the implementation of the proposed concept enables mathematics education to become personalized and transition to a new level of effectiveness in accordance with the requirements of Education 4.0.

In this context, the experience of the University of Artificial Intelligence (https://neural-university.ru) deserves particular attention. This institution has carried out a large-scale transformation of its educational process, integrating its own concept of neural employees. During the project's implementation, the university platform successfully deployed a neural employee system that demonstrates a high level of effectiveness. According to testing data, the quality of the neurocurator's responses reached 4.9 out of 5, and 81% of homework assignments are now checked using neuroverification. The introduction of the neurolecturer significantly reduced the time teachers spend preparing and delivering lectures, as well as increasing student engagement through interactive capabilities and personalized material delivery.

It should be noted, however, that the multi-agent infrastructure is not exhausted by the ensemble of AI agents alone. A critically important component of the architecture is unified API interfaces and inter-agent communication protocols, which ensure agent-to-agent and agent-to-external service interaction. In [10], the author highlighted the need to develop new information exchange protocols for agents, which was subsequently realized in the A2A, MCP, ACP, and other protocols [27–31] developed after [10]. In reality, the new protocols [27–31] have only partially implemented the ideas proposed in [10]. In particular, the issue of real-time high-resolution video stream exchange between agents remains unresolved. Thus, further development and eventual standardization of those approaches most relevant to practical needs is to be expected.

Hierarchical Multi-Agent System for Higher Mathematics Education

The multi-agent system for mathematics education described above represents an example of a relatively simple set of linearly interacting intelligent agents, each specializing in separate aspects of the educational process and implemented on the basis of an LLM, whose functionality is constrained by instructions.

A more advanced implementation of the multi-agent approach consists in applying a modular architecture for each of the described agents, with a complex hierarchy of their functions (see Fig. 2).

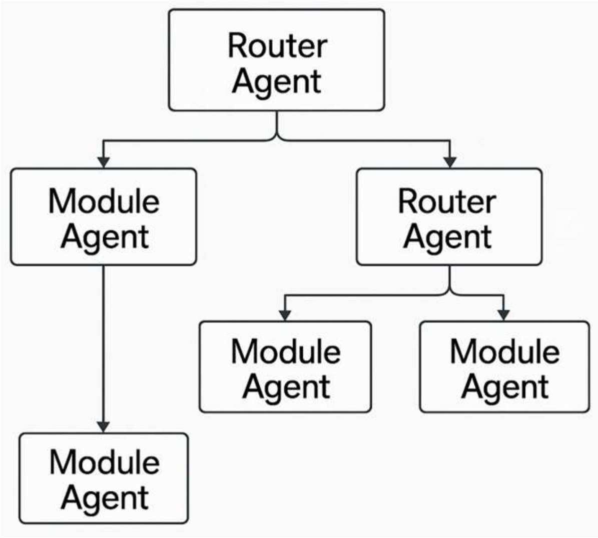

Fig. 2 Modular concept of agent hierarchy in a multi-agent system

This concept allows each branch of mathematics to be represented within the hierarchical multi-agent system by specialized subject agents, each with its own knowledge base, problem-solving algorithms, and explanatory techniques adapted to the user's cognitive style. For example, the algebra agent provides support in studying linear algebra, matrix theory, systems of linear equations, and vector spaces; it interacts with the visualizer agent to generate graphical interpretations of linear transformations and geometric objects. The mathematical analysis agent is responsible for differential and integral calculus, series, limits, and functional analysis; in close cooperation with the tester agent, it generates individualized exercises of increasing complexity and assesses understanding of concepts such as limits, derivatives, and integrals. The analytic geometry agent works with three-dimensional models of surfaces and curves, generates appropriate tasks for interactive simulations, and coordinates with the augmented reality agent, which projects geometric objects into virtual space for real-time manipulation. The differential equations agent specializes in analytical and numerical methods for solving equations and supports interaction with the simulation agent, enabling experiments with models of physical processes described by differential equations. A dedicated probability and statistics agent implements stochastic models, distributions, statistical estimation, and interacts with the hypothesis validation agent, allowing the application of educational material to real-world data analysis. Finally, the complex function theory agent is integrated with the animation agent for dynamic demonstration of conformal mappings and Laurent transformations, among other concepts.

To select a specific data processing pathway, a router agent (see Fig. 2) is employed, which determines the relevant branch for a given task and ensures interdisciplinary interaction. It creates learning trajectories by combining concepts from different branches in accordance with educational goals, such as modeling physical or economic problems. In cases where problem-solving requires interdisciplinary competence (for example, finding the optimum of a function under constraints described by differential equations), a collective session is activated, in which agents for mathematical analysis, optimization theory, and linear algebra work simultaneously. Such interaction not only supports students in solving problems, but also models real scenarios of collaboration in the scientific community, where decomposing complex problems into modules allows effective combination of knowledge and algorithmic approaches.

In addition to subject specialization, the modular architecture must also include agents responsible for various functional domains and stages of interaction with students.

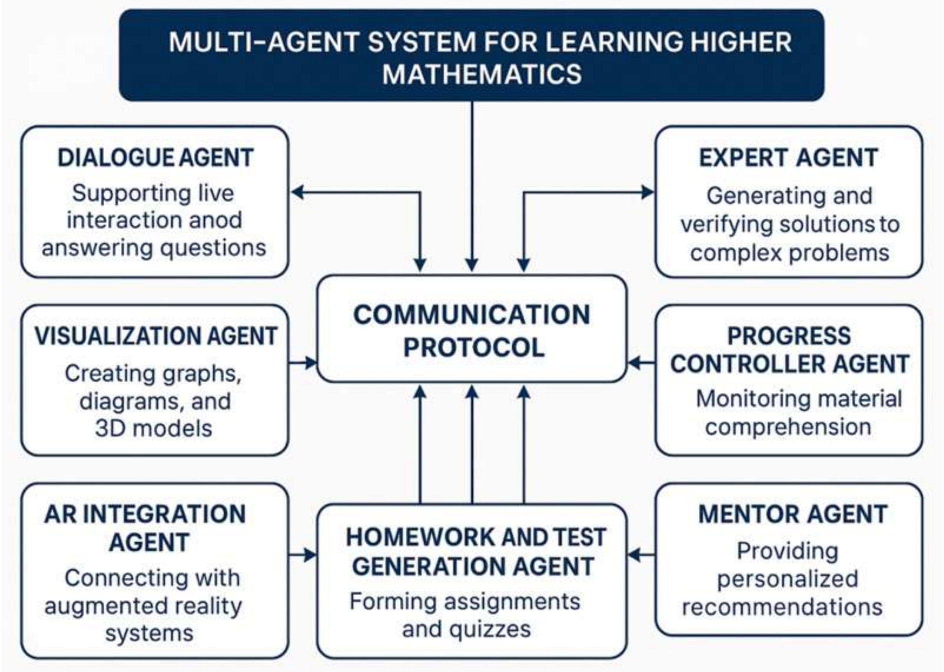
MULTI-AGENT SYSTEM FOR LEARNING HIGHER MATHEMATICS
DIALOGUE AGENT
Supporting live interaction anod answering questions
EXPERT AGENT
Generating and verifying solutions to complex problems
COMMUNICATION PROTOCOL
VISUALIZATION AGENT
Creating graphs, diagrams, and 3D models
PROGRESS CONTROLLER AGENT
Monitoring material comprehension
AR INTEGRATION AGENT
Connecting with augmented reality systems
HOMEWORK AND TEST GENERATION AGENT
Forming assignments and quizzes
MENTOR AGENT
Providing personalized recommendations

Fig. 3 Functional agents responsible for specific stages of interaction.

For example, Fig. 3 presents a dialog agent for supporting live communication and answering students' questions, an expert agent for generating and verifying solutions to complex mathematical problems, a visualizer agent for creating graphs, diagrams, and 3D models, a progress controller agent for monitoring knowledge acquisition, and a mentor agent for individualized selection of learning pathways and recommendations. The actions of these agent modules are coordinated through a local communication protocol, which also enables integration with external educational resources such as digital textbooks, online courses, and interactive simulators.

The system may include specialized agents for homework verification and for generating test tasks of varying complexity. The homework and assessment generation agent in a multi-agent higher mathematics education system is a specialized intelligent module that automatically creates individualized assignments for self-study and knowledge assessment, focusing on the level of content mastery, topics covered, types of errors, the student's thinking style, and course objectives. The main function of the agent is to create substantively relevant, differentiated, and verifiable educational content that can be used both in paper form and in digital or VR environments. Its architecture includes several functional modules:

1. The progress analysis agent interacts with the assessment module and the student's activity tracker to collect statistics on completed tasks, response time, characteristic errors, and topics requiring review. This analysis forms a complexity profile for each student.
2. The task generation agent, based on mathematical templates (for example, for derivatives, equations, integrals, or geometry), generates task variants with parameterization (generation of variables, numbers, conditions). It can also generate non-standard problems with open-ended answers or multi-step logic. For example: "Calculate the area of the figure bounded by the graph of $y=\sin x$ and the lines $x=0$, $x=\pi$."
3. The complexity and coverage control agent uses pedagogical criteria to balance assignments according to complexity, types of thinking (reproductive, analytical, creative-logical), cognitive load, and time constraints. For example, each version of a test includes a basic task, one on application, and one on synthesis of knowledge.
4. The supplementary material generation agent creates versions with hints, teacher-format answers, or automatically generates solutions with explanations, which can be presented in text or visual form. Its functionality overlaps to some extent with the previously described methodologist agent, but it is more specific to its area of specialization.
5. The multimodal support agent can generate tasks in the form of videos or voice messages, as well as support interaction via video camera, where responses are selected through object interaction or figure drawing.

6. The security and uniqueness verification agent checks task variants for plagiarism, minimizes the risk of memorizing answers due to template-based repetition, and generates a large number of unique variants for different students.

By interacting with other agents, such as the mentor or progress agent, it ensures a full learning cycle: from task assignment to explanation, control, and review in a format convenient for the student.

These examples clearly demonstrate that the application of a multi-agent system enables the organization of a personalized educational environment. The differentiated capabilities of agents in their complex hierarchy dynamically adapt to users' educational needs, ensuring prompt feedback. Taken together, all of this will contribute to a deeper understanding of fundamental mathematical concepts by automating and enhancing the possibilities of traditional education.

Given the broad range of language models that can be used to implement a substantial variety of agents, it is necessary to consider metrics that make it possible to select the most effective solutions for deploying multi-agent architectures.

Assessment of the Mathematical Abilities of LLMs at All Educational Levels

The capabilities of LLMs are generally evaluated using specialized benchmark datasets designed to measure their mathematical skills - from basic arithmetic to complex university-level problems. The most widely used metrics for evaluating the mathematical abilities of LLMs encompass various educational levels and skill types, as well as the strengths and weaknesses of the models (including their contribution to the development of structured reasoning and identification of typical errors), performance analysis, and the potential for automated testing. Let's examine these metrics in greater detail.

MATH [33] offers 12,500 challenging problems from school mathematics olympiads (algebra, geometry, number theory, probability, trigonometry) across five difficulty levels. Answers are automatically checked, so even a single computational error yields zero points; however, the dataset effectively identifies gaps in multi-step reasoning and provides detailed reference solutions, which are beneficial for training models.

GSM8K [34] contains 8,500 word problems involving everyday arithmetic reasoning for gifted middle school students. Each test requires 2–8 logical steps and demands an exact numerical answer. Chain-of-thought explanations are popular, but only the final result is evaluated, allowing for correct answers without correct reasoning, and advanced topics (algebra, geometry) are not covered.

MMLU [35] encompasses 57 disciplines with multiple-choice questions ranging from school to university level, including mathematics, logic, and statistics. The result is determined by matching the selected option with the key, enabling quick model comparisons but allowing for guessing, and errors in the keys reduce accuracy.

MathQA [36] provides 37,000 problems from school mathematics and applied fields, each with an "operation program" of steps. The metric is the accuracy of choosing from five options; the presence of explanations aids learning, but the multiple-choice format again allows for guessing, and noise and atypical operations complicate evaluation.

HumanEval [37] includes 164 university-level programming tasks: the model must write a Python function that passes hidden tests. The benchmark effectively assesses algorithmic thinking and precision but focuses more on programming than classical mathematics, and the small number of tasks makes results variable.

Together, these datasets cover elementary school arithmetic, olympiad problems, and university-level algorithmics, providing a multidimensional snapshot of LLMs' mathematical abilities. The absence of partial credit and the possibility of guessing leave room for new metrics that would more accurately reflect models' real understanding and help better select them for educational purposes.

NEW METRICS FOR ASSESSING MATHEMATICAL MULTI-AGENT SYSTEMS

Given the limitations of current metrics for evaluating the mathematical abilities of LLMs, a set of new metrics is proposed, specifically oriented toward mathematics education.

The first proposed metric, **StepScore**, measures the justification of reasoning. It assesses the logical sequence of the solution, awarding points for each correctly derived intermediate step. This is achieved using a specialized LLM or a graph-based logic verifier, which compares the model's steps to the rules of algebra, geometry, etc. The assessment is carried out step-by-step: correct formula → +1; correct transformation → +1; valid logic → +1; final answer → +1. The essence of the metric can be described by the following weighted formula:

$$StepScore = \frac{\sum_{i=1}^{n} w_i c_i}{\sum_{i=1}^{n} w_i},$$

where n is the number of logical steps in the solution, $c_i \in \{0, 1\}$ denotes the correctness of the i-th step (1—correct, 0—incorrect), and $w_i \geq 0$ is the importance weight of the i-th step (by default, $w_i = 1$).

Essentially, this expression yields a normalized weighted average score that reflects the proportion of correctly executed logical transformations. The metric allows for partial credit and pinpoints where exactly the model fails. This indicator can be used in instruction as feedback.

The next metric, **Learnability Index**, measures how effectively the LLM's answer explains the solution process step by step to a student, rather than simply providing the correct answer. The mechanism assumes that each response is analyzed by a mentor agent (e.g., another LLM with a pedagogical prompt). This agent rates clarity of language (Is this understandable for a 7th grader?), justification of steps, and presence of motivational or explanatory phrases ("we see that…", "thus…"). The metric is a linear combination of didactic quality features with weights set by methodologists or empirically:

Learn ability Index = $\lambda_1 \cdot \alpha + \lambda_2 \cdot \beta + \lambda_3 \cdot \gamma \in [0, 1]$,

where $\alpha \in [0, 1]$ is linguistic clarity, $\beta \in [0, 1]$ is didactic alignment, $\gamma \in [0, 1]$ is the presence of inductive or explanatory markers (pedagogical signals), and $\lambda_1, \lambda_2, \lambda_3 \geq 0$ are the weights of these components, with $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

Scoring is performed on a scale from 0 to 1 or by categories (A/B/C). Sample phrasings in responses are compared with didactic patterns. The advantages of this metric include sensitivity to the didactic quality of the response. It encourages the development of models capable of teaching people and can even be applied to incorrect answers.

To more precisely capture how well LLM capabilities match users' educational levels, the **Adaptive Curriculum Error Profiling (ACEP)** metric can be used, which builds an error profile of the model along the curriculum scale. Each problem in the test/evaluation set is labeled with the corresponding educational level (e.g., Algebra I / Geometry / Linear Systems). Based on a series of problems, the metric generates an error map: type of error (arithmetic, logical, conceptual); topic in which the error occurred; frequency of errors in topics. This produces an error profile as a vector $\vec{e} = (e_1, e_2, \ldots, e_m)$, where $e_j \in [0, 1]$, with e_j representing the proportion of problems in topic j solved with an error, or the failure rate on the j-th item.

The educational level conformity function is defined as:

$$ACEP = 1 - \frac{1}{m}\sum_{i=1}^{n} e_i,$$

where m is the number of topics corresponding to the target educational level.

ACEP is the inverse of the mean error share over the curriculum profile and takes values in [0, 1]. Applying it allows one to obtain a map of model weak points, retraining recommendations, and a meta-assessment of suitability for a given level (e.g., “not suitable for 9th grade”). ACEP fully reflects the educational context; it provides explanations for where the model is “undertrained” and can be used for model comparison according to student needs.

A summary comparison of the new and existing metrics is presented in Table 1.

Finally, it is appropriate to use a combined metric, **HybridMathScore**, which integrates all the proposed indicators - StepScore, LearnabilityIndex, and CurriculumFitness, - into a single assessment. In general, such a combined assessment can be presented as a weighted sum:

HybridMathScore = $\mu_1 \cdot S + \mu_2 \cdot L + \mu_3 \cdot C$,

where S is StepScore, L is LearnabilityIndex, C is ACEP, and $\mu_1, \mu_2, \mu_3 \in [0, 1]$ are metric weights reflecting the task objective (for example, for primary education, more emphasis is placed on L; for upper grades, on S), with $\mu_1 + \mu_2 + \mu_3 = 1$.

Table 1 Comparative Characteristics of the Proposed Metrics

Metric	Logic Assessment	Partial Credit	Educational Suitability	Adaptivity
MATH	x	x	X	x
GSM8K	partial	x	X	x
MMLU	x	x	X	x
MathQA	partial	x	X	x
HumanEval	+ (via code)	X	X	x
StepScore	+	+	partial	x
Learnability	partial	partial	+	x
ACEP	+	X	+	+

For upper secondary school, the following distribution of weighting coefficients may be used: $\mu_1 = 0.5$ (stepwise logic is most important), $\mu_2 = 0.2$ (explanations are important but secondary), $\mu_3 = 0.3$ (curricular alignment is moderately important). For primary school, more attention should be given to explanations, selecting, for example, $\mu_1 = 0.2$, $\mu_2 = 0.5$, $\mu_3 = 0.3$.

The proposed metrics fill critical gaps in the current LLM evaluation system. They make it possible not only to measure how "intelligent" a model is, but also how teachable it is, whether it can explain solutions, and where it makes mistakes. These metrics can serve as the foundation for new test sets and benchmark platforms tailored to educational needs - both for automated tutoring and for integrating LLMs into distance mathematics learning systems.

In the context of multi-agent systems, the proposed metric framework enables comparison of different LLMs as educational agents, revealing their strengths and weaknesses not only in terms of outcome, but also in logic, style, and educational alignment. All of this is key to adapting evaluation to the needs of students at different levels through weighted parameters.

BIBLIOGRAPHY

A. Laurent, 'La guerre des intelligences à l'heure de ChatGPT,' Lattes. 2023, 480 p.

A. Shevchenko, V. Panok, A. Shevtsov, V. Slyusar, R. Malyi, T. Yeroshenko, M. Nazar. Development of a virtual psychological assistant with artificial intelligence in the healthcare sector. // Clinical and Preventive Medicine, 2024. N 8. pp. 15 - 27. DOI: 10.31612/2616-4868.8.2024.02.

Amini, A., Gabriel, S., Lin, P., Koncel-Kedziorski, R., Choi, Y., & Hajishirzi, H. (2019). *MathQA: Towards interpretable math word problem solving with operation-based formalisms*. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 2357–2367). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1245.

Anthropic, 'Introducing Claude,' Anthropic News, Mar. 14, 2023. [Online]. Available: https://www.anthropic.com/news/introducing-claude.

Anthropic. (2025, May). *System card: Claude Opus 4 & Claude Sonnet 4*. https://www.anthropic.com/model-card

Barto, A. G., Sutton, R. S., & Watkins, C. J. C. H. (1989, September). *Learning and sequential decision making* (COINS Technical Report 89-95). University of Massachusetts Amherst. https://www.researchgate.net/publication/334786316_Learning_and_Sequential_Decision_Making.

Bhardwaj, D., Beniwal, A., Chaudhari, S., Kalyan, A., Rajpurohit, T., Narasimhan, K. R., Deshpande, A., & Murahari, V. (2025). *Agent context protocols enhance collective inference*. arXiv. https://arxiv.org/abs/2505.14569

Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H. P., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., ... Amodei, D. (2021). *Evaluating large language models trained on code*. arXiv preprint arXiv:2107.03374. https://arxiv.org/abs/2107.03374.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., & Schulman, J. (2021). *Training verifiers to solve math word problems*. arXiv preprint arXiv:2110.14168. https://arxiv.org/abs/2110.14168.

Cox, A. M. (2021). Exploring the impact of artificial intelligence and robots on higher education through literature-based design fictions. *International Journal of Educational Technology in Higher Education, 18*(1), 3. https://doi.org/10.1186/s41239-020-00237-8.

D. Hassabis et al., 'Introducing Gemini: Our Largest and Most Capable AI Model,' Google Blog, Dec. 2023. [Online]. Available: https://blog.google/technology/ai/google-gemini-ai/.

G. Taverniti, C. Lombardo, et al., 'AI Power. Non solo ChatGPT: lavoro, marketing e futuro,' Editore Ulrico Hoepli, Milano, 2023, 224 p.

H. Touvron, T. Lavril, et al., 'LLaMA: Open and Efficient Foundation Language Models,' *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2302.13971.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., & Steinhardt, J. (2020). *Measuring massive multitask language understanding*. arXiv preprint arXiv:2009.03300. https://doi.org/10.48550/arXiv.2009.03300.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., & Steinhardt, J. (2021). *Measuring mathematical problem solving with the MATH dataset*. arXiv preprint arXiv:2103.03874. https://doi.org/10.48550/arXiv.2103.03874.

Hightower, R. (2025, May 28). *Is RAG dead?: Anthropic says no*. Medium. https://medium.com/@richardhightower/is-rag-dead-anthropic-says-no-290acc7bd808

I. Habler, K. Huang, V. S. Narajala, and P. Kulkarni, "Building A Secure Agentic AI Application Leveraging A2A Protocol," arXiv, Apr. 2025. [Online]. Available: https://arxiv.org/abs/2504.16902

Jeong, C. (2025). *A study on the MCP × A2A framework for enhancing interoperability of LLM-based autonomous agents*. arXiv. https://arxiv.org/abs/2506.01804

Kondratenko Y., Shevchenko A., Zhukov Y., Slyusar V., Kondratenko G., Klymenko M., Striuk O., 'Analysis of the Priorities and Perspectives in Artificial Intelligence Implementation' (2023). 2023 13th International Conference on Dependable Systems, Services and Technologies, DESSERT 2023, DOI: 10.1109/DESSERT61349.2023.10416432.

Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., Wu, Y., Neyshabur, B., Gur-Ari, G., & Misra, V. (2022). *Solving quantitative reasoning problems with language models*. arXiv preprint arXiv:2206.14858. https://arxiv.org/abs/2206.14858.

Liu, J., Yu, K., Chen, K., Li, K., Qian, Y., Guo, X., Song, H., & Li, Y. (2025). *ACPs: Agent collaboration protocols for the Internet of Agents*. arXiv. https://arxiv.org/abs/2505.13523

Martin, F., Zhuang, M., & Schaefer, D. (2024). *Systematic review of research on artificial intelligence in K-12 education (2017–2022). Computers and Education: Artificial Intelligence, 6*, 100195. https://doi.org/10.1016/j.caeai.2023.10019.

Martineau, K. (2025, May 28). *The simplest protocol for AI agents to work together*. IBM Research. https://research.ibm.com/blog/agent-communication-protocol-ai

OpenAI, 'GPT-4 technical report,' *arXiv*, 2023. [Online]. Available: https://archive.org/details/gpt-4-technical-paper.

OpenAI, 'Introducing ChatGPT', November 30, 2022. [Online]. Available: https://openai.com/blog/chatgpt/.

OpenAI, 'OpenAI GPT-4.5 System Card,' Feb. 27, 2025. [Online]. Available: https://cdn.openai.com/gpt-4-5-system-card.pdf.

P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al., 'Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks', Advances in Neural Information Processing Systems, vol. 33, pp. 9459–9474, 2020. [Online]. Available: https://arxiv.org/abs/2005.11401.

Qwen Team, 'Qwen2-VL: To See the World More Clearly', Aug. 2024. [Online]. Available: https://qwenlm.github.io/blog/qwen2-vl/.

V.I. Slyusar, Y.P. Kondratenko, A.I. Shevchenko, T.V. Yeroshenko, 'Some Aspects of Artificial Intelligence Development Strategy for Mobile Technologies', Journal of Mobile Multimedia, Vol. 20_3, pp. 525–554, 2024. DOI: 10.13052/jmm1550-4646.2031.

V. Slyusar, 'Distributed Multi-agent Systems Based on the Mixture of Experts Architecture in the Context of 6G Wireless Technologies'. In: Dovgyi, S., et al. (eds) Applied Innovations in Information and Communication Technology. ICAIIT 2024. Lecture Notes in Networks and Systems, vol. 1338. Springer, 2025, pp. 81-110. DOI: 10.1007/978-3-031-89296-7_6.

V. Slyusar. 'Perspectives of the AI Applications for Improving Learning and Teaching', Processes. AI in Education System: Successful Cases and Perspectives. 2025.

Watkins, C. J. C. H. (1989, May). *Learning from delayed rewards* (Doctoral dissertation, University of Cambridge). Retrieved from https://www.cs.rhul.ac.uk/~chrisw/new_thesis.pdf.

X. Bi, et al., 'DeepSeek LLM: Scaling Open-Source Language Models with Longtermism,' *arXiv*, 2024. [Online]. Available: https://arxiv.org/abs/2401.02954.

Y. Kondratenko, G. Kondratenko, A. Shevchenko, V. Slyusar, Y. Zhukov, M. Vakulenko, 'Towards Implementing the Strategy of Artificial Intelligence Development: Ukraine Peculiarities', CEUR Workshop Proceedings, 3513, pp. 106–117, 2023. Available: https://ceur-ws.org/Vol-3513/paper09.pdf.

Y. P. Kondratenko, V. I. Slyusar, M. B. Solesvik, N. Y. Kondratenko, and Z. Gomolka, 'Interrelation and inter-influence of artificial intelligence and higher education systems', *Research Tendencies and Prospect Domains for AI Development and Implementation*, pp. 31–58, 2024.

Zawacki-Richter, O., Marín, V. I., Bond, M., & Gouverneur, F. (2019). Systematic review of research on artificial intelligence applications in higher education—Where are the educators? *International Journal of Educational Technology in Higher Education, 16*(1), 39. https://doi.org/10.1186/s41239-019-0171-0.

'Open Release of Grok-1,' xAI Blog, Mar. 17, 2024. [Online]. Available: https://x.ai/blog/grok-os.