UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Đặng Văn Khải

RESEARCH ON GENERATIVE AI APPLICATIONS IN EDUCATION

BACHELOR THESIS

Major : Artificial Intelligence

Hanoi, 2025

UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Đặng Văn Khải

RESEARCH ON GENERATIVE AI APPLICATIONS IN EDUCATION

BACHELOR THESIS

Major : Artificial Intelligence

Supervisor: Ph.D Trần Văn Khánh

Hanoi, 2025

Authorship

"I hereby declare that the work contained in this thesis is of my own and has not been previously submitted for a degree or diploma at this or any other higher education institution. To the best of my knowledge and belief, the thesis contains no materials previously published or written by another person except where due reference or acknowledgement is made."

Signature:……………………………………………

Supervisor's approval

"I hereby approve that the thesis in its current form is ready for committee examination as a requirement for the Bachelor of Computer Science degree at the University of Engineering and Technology."

Signature:………………………………………………

Acknowledgments

And above all, this thesis is for Huong, the one that completes me.

Research on Generative AI applications in education

Abstract

test

This is title of the thesis in Vietnamese

Tóm tắt đồ án

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Background and Research Motivation

Many students lack personalized attention in traditional classrooms, leading to anxiety and difficulty engaging with subjects they struggle in, such as mathematics. Even with access to abundant resources, they often lack the guidance to use them effectively—especially in online settings. Challenges also arise in communication, collaboration, and problem-solving, both among students. These issues call for more adaptive and supportive educational approaches.

Recent advances in large language models (LLMs) offer promising solutions. LLMs demonstrate strong capabilities in mathematical reasoning and can solve middle-school-level problems with near-perfect accuracy (e.g., GPT-4 on GSM8k [1]). Beyond problem-solving, LLM-based agents can simulate human-like behaviors and social interactions, enabling realistic, personalized learning environments. This opens new possibilities for adaptive education, where students benefit from peer-like interactions and tailored support—especially when engaging with mixed-ability groups.

1.2 Research Problem

Mathematical modeling (MM) is a foundational skill in STEM education that relies heavily on collaborative learning and peer discussion. However, many students—

particularly those in underserved communities—lack access to high-quality support and guidance in developing this skill. Contributing factors include a shortage of well-trained teachers capable of facilitating mathematical dialogue, as well as personal barriers such as social anxiety or low self-confidence in group settings. In today's shift toward learner-centered and inquiry-driven education, it is vital to nurture students' mathematical thinking and real-world problem-solving abilities through interactive and communicative learning approaches.

With the advancement of large language models, there is growing interest in using multi-agent systems to simulate dynamic classroom environments. These systems have the potential to foster meaningful peer interaction, increase student engagement, and support the development of mathematical reasoning through dialogue. Yet, important challenges remain: understanding how effectively these systems can mimic real-time classroom interactions, whether they can provide students with a sense of immersion and enhance learning outcomes, and how natural group behaviors such as cooperation, discussion, and emotional dynamics might unfold in these settings.

1.3 Research Objectives and Scope

This study aims to explore the potential of large language model (LLM)-powered multi-agent systems in supporting high school students' development of mathematical modeling skills through collaborative problem-solving. By creating interactive learning environments where students can engage in dialogue, explanation, and reasoning with AI agents, the research seeks to enhance students' mathematical thinking, communication, and teamwork abilities—skills that are essential in both academic and real-world contexts.

This study focuses on high school mathematics problems as the basis for exploring how multi-agent AI systems can support collaborative learning and simulate interactive classroom experiences.

1.4 Thesis Structure

This thesis is organized into five chapters. Chapter 1 introduces the research background, highlighting challenges in traditional education and the potential of LLM-powered multi-agent systems for collaborative mathematical modeling, defining the research problem and objectives. Chapter 2 reviews relevant literature, covering the evolution of AI in education, one-to-one tutoring, and the concept of virtual classrooms using multi-agent systems, identifying key research gaps. Chapter 3 details the proposed methodology, outlining the design goals and architecture of the multi-agent system, including its event-driven nature, stage management based on pedagogical principles, proactive turn-taking mechanism, and role-based agent customization. Chapter 4 describes the experimental setup used to evaluate the system, including task design, dataset collection, the LLM-as-a-judge evaluation method, and presents the main findings and discussion. Finally, Chapter 5 concludes the thesis by summarizing the contributions, acknowledging limitations, discussing ethical considerations, and suggesting avenues for future work.

Chapter 2

LITERATURE REVIEW

2.1 Generative AI in Education

2.1.1 Before the Era of Large Language Models

Before the prominence of Large Language Models (LLMs) around 2018-2019, AI significantly shaped education through technologies like Intelligent Tutoring Systems (ITS), Machine Learning (ML), Natural Language Processing (NLP), and intent-based classification systems. ITS, originating in the 1970s, evolved into sophisticated knowledge-based tutors by the 1990s, offering personalized instruction by adapting to student needs [1]. ML was pivotal in predicting at-risk students, personalizing learning, and automating tasks like grading, with institutions like Western Governors University improving graduation rates using predictive models [2]. These advancements laid a robust foundation for personalized and efficient educational practices.

NLP facilitated natural interactions via automated scoring, writing assistance, and e-learning systems, enhancing student support, as seen in platforms like Brainly, which used NLP for instant question answering [3]. Intent-based classification systems, integral to early chatbots, classified student queries to provide relevant responses, improving engagement, as evidenced by Georgia Tech's AI-powered virtual assistant, Jill Watson [4]. Case studies from Stanford and the University of Phoenix further highlight how these technologies delivered scalable, tailored solutions, setting the stage for LLMs by transforming teaching, learning, and administrative processes.

2.1.2 During the Era of Large Language Models

The advent of Large Language Models (LLMs) has marked a significant turning point in the educational domain, with their influence becoming particularly notable since the introduction of BERT and accelerating with the release of ChatGPT in 2022. This evolution has shifted the paradigm from traditional natural language processing (NLP) techniques to more sophisticated, transformer-based models that offer unprecedented capabilities. LLMs have become increasingly accessible and user-friendly, enabling educators and students alike to harness their power for a wide range of educational purposes. Their transformative impact lies in their ability to scale personalized learning, comprehend extended contexts, enhance creative processes, and facilitate interactive roleplay scenarios, all while raising important ethical considerations that must be addressed to ensure their responsible integration into education [1, 2].

One of the cornerstone benefits of LLMs is their scalability, which allows them to deliver highly personalized learning experiences to diverse student populations. By automating labor-intensive tasks such as generating educational content, providing real-time feedback, and even grading assignments, LLMs make it feasible to tailor instruction to individual needs on a large scale. For example, adaptive learning platforms powered by LLMs can dynamically adjust lesson difficulty or suggest supplementary materials based on a student's performance, thereby improving engagement and comprehension. Furthermore, LLMs excel at processing and summarizing complex texts—such as academic textbooks, research papers, or lecture notes—making them indispensable tools for educators seeking to distill key insights efficiently. Studies have demonstrated that LLMs can produce reading comprehension exercises that rival or exceed the quality of those crafted by humans, offering students robust tools to master challenging material [3, 4, 5]. Beyond comprehension, LLMs foster creativity by aiding students in brainstorming ideas, conducting preliminary research, and refining their creative outputs, such as essays or project proposals. While this support can help students overcome creative blocks, educators must ensure that reliance on LLMs does not overshadow the development of independent critical thinking skills [6].

In addition to their analytical and creative applications, LLMs bring a unique interactive dimension to education through their roleplay capabilities. By simulating characters—ranging from historical figures to conversational partners—LLMs create immersive learning environments that enhance engagement and practical skill-building. A notable example is Duolingo's Roleplay feature, where LLMs power realistic language practice scenarios, allowing learners to apply vocabulary and grammar in context. This interactivity not only makes learning more enjoyable but also bridges theoretical knowledge with real-world application. However, the use of LLMs in such roles introduces ethical challenges, particularly the risk of anthropomorphism, where users might attribute human-like qualities to these models. To mitigate this, it is essential to frame LLMs as simulations rather than sentient entities, preserving trust and clarity in educational settings [7, 8]. As LLMs continue to evolve, their integration into education promises to deepen, provided that ethical guidelines and critical oversight keep pace with technological advancements.

2.1.3 Some Related Products for Education

Globally, a diverse range of products enhances educational outcomes:

- Speechify []: This tool utilizes generative AI to convert text into natural-sounding speech, significantly improving accessibility for students with learning disabilities such as dyslexia or visual impairments.
- NOLEJ []: Designed for educators and instructional designers, NOLEJ harnesses generative AI to streamline the creation of e-learning content. It can rapidly generate interactive lessons, quizzes, and multimedia resources from raw educational material.
- Synthesia []: This platform employs generative AI to produce professional-quality educational videos featuring AI-generated avatars.
- Grammarly []: Beyond basic grammar and spell-checking, Grammarly's generative AI features now suggest style improvements, tone adjustments, and even full sentence rephrasings, etc.

These tools illustrate the broad applications of generative AI, from content generation to creative visualization, transforming how education is delivered worldwide.

The market for generative AI in education is projected to grow substantially, reaching USD 7,701.9 million by 2033 with a compound annual growth rate (CAGR) of 39.5%, fueled by demand for personalized learning solutions [2]. Usage statistics reveal widespread adoption: 44% of children engage with generative AI for schoolwork, and 60% of teachers integrate it into their teaching practices [3]. However, ethical challenges persist, with 24.11% of charter high school students reporting AI-related cheating incidents, highlighting the need to address misuse alongside its benefits [3].

2.2 One-to-one Tutoring

2.2.1 Pedagogical Strategies in One-to-One Tutoring

One-to-one tutoring using AI systems, especially those powered by LLMs, leverages various pedagogical strategies to enhance learning outcomes. These strategies are designed to mimic human tutoring while scaling to meet individual student needs, including:

- Socratic Method: This approach involves AI tutors asking probing, open-ended questions to stimulate critical thinking and guide students toward self-discovery. It is particularly effective in fostering deep understanding and is implemented in systems like AutoTutor[xx], which simulates human tutor discourse patterns for computer literacy, and SocraticTutor[xx], designed for programming education. Furthermore, the leading artificial intelligence company Anthropic[xx] has launched an educational product that underscores the importance of this methodology.

- Inquiry-Based Learning [yy]: This strategy encourages students to pose questions and explore topics through guided investigation, promoting active learning. This method is learner-centered, fostering curiosity and deeper engagement, particularly in general education settings.

- Personalized Learning [zz]: Tailoring educational content to individual student needs, preferences, and learning styles is a cornerstone of AI-driven tutoring. This strategy enhances engagement and motivation, with chatbots analyzing student data to adapt teaching methods, etc.

2.2.2 Applications

Educational chatbots are applied across a wide range of subjects and educational fields, demonstrating their versatility:

- STEM Subjects
- English as a Foreign Language (EFL)
- Programming
- Medical Education
- Soft Skills Development

2.2.3 Beyond One-to-One

While one-to-one tutoring offers personalized attention, it faces challenges in simulating the full spectrum of classroom interactions, particularly social and collaborative elements. One-to-one settings often miss peer learning opportunities, which are crucial for social development and collaborative skills. In contrast, traditional classrooms foster peer interactions that enhance learning through discussion and shared problem-solving. These limitations highlight the need for a more comprehensive approach to simulate realistic learning experiences.

2.3 Virtual Classroom

2.3.1 Motivation

The motivation behind designing a Virtual Classroom comes from the need to create a more natural and interactive educational environment that mirrors the complexities of human conversation. By integrating multiple AI agents, the virtual classroom can create a collaborative environment in which these agents actively engage alongside students. This design aims to shift away from tutor-centric models, encouraging students' proactive involvement and enhancing the overall learning experience through sophisticated interaction management.

For students and tutors, this system offers significant benefits by promoting active participation and providing practical training opportunities. Students gain from AI agents acting as peers or tutors, which can enhance engagement through emotional support and equality in collaborative learning activities, as seen in systems like PeerGPT. These agents enrich peer conversations and support knowledge co-construction, making learning more interactive and supportive. For novice tutors, a virtual classroom with AI-driven scenarios, inspired by tools like TutorUp, provides a cost-effective alternative to traditional training methods. It allows them to practice managing online teaching challenges—such as technical difficulties or student engagement—in a controlled, repeatable environment, addressing the logistical and financial barriers of scenario-based training.

Integrating virtual reality (VR) with multiple AI agents further amplifies the potential of the virtual classroom by creating an immersive, socially interactive learning space. Research highlights that VR classrooms foster genuine social interaction and boost learner motivation through collaborative social presence, where participants' actions influence one another in real-time. By combining VR's immersive capabilities with AI agents that can engage naturally with students and each other, the system simulates authentic educational settings more effectively. This not only heightens student interest and persistence but also offers a scalable, accessible solution for both learning and tutor training, pushing the boundaries of contemporary education technology.

2.3.2 Multi-agent Systems (MAS)

LLM-based multi-agent systems involve multiple intelligent agents powered by large language models (LLMs) collaborating to address complex tasks. These systems enhance capabilities such as planning, decision-making, memory, and tool use compared to single-agent setups. For example, in software development, one agent might focus on design while another handles coding, improving efficiency through task distribution [1]. Agents employ structured communication methods—cooperative, debate, or competitive—to refine solutions, supported by frameworks like MetaGPT, CAMEL, and Autogen, which facilitate coordination and task execution [1]. Memory mechanisms, including short-term for recent interactions and long-term for historical data, allow agents to maintain context and improve over time, as seen in applications like financial trading [1]. Additionally, tool integration, such as APIs or code interpreters, extends their functionality to domains like robotics and economic simulations [1].

In Virtual Classrooms, MAS are highly suitable because they can simulate a dynamic, interactive educational environment akin to a real classroom. Multiple AI agents can assume roles such as instructors, assistants, or peer learners, interacting with real students to provide personalized feedback and foster collaborative learning through simulated discussions [1]. This setup leverages the agents' ability to offer diverse perspectives and adapt to individual student needs, enhancing engagement and understanding [1]. By mimicking group dynamics and facilitating interactive problem-solving, MAS create a rich, adaptive learning experience that mirrors traditional classroom interactions, making them an effective tool for education [1].

2.3.3 Turn-takings in Multi-Party Conversations

2.4 Summary and Research Gap

AI in education has undergone significant evolution, progressing from early technologies like Intelligent Tutoring Systems (ITS) and Machine Learning (ML) to the transformative era of Large Language Models (LLMs). These advancements have enabled scalable personalized learning, enhanced creativity, and interactive roleplay scenarios, revolutionizing how educational content is delivered and experienced. However, while one-to-one tutoring powered by LLMs offers tailored pedagogical strategies, it falls short in replicating the collaborative dynamics inherent in a traditional classroom setting. To address this limitation, the concept of a Virtual Classroom using multi-agent systems (MAS) has emerged as a promising solution.

Despite these advancements, there are notable challenges in the study of Virtual Classrooms simulated by multi-agent systems. The concept is relatively new, with few studies exploring similar ideas, such as SimClass and MathVC, indicating that the field is still underexplored. Additionally, evaluating the effectiveness of such systems presents significant challenges. Unlike traditional one-to-one tutoring, assessing a Virtual Classroom requires analyzing complex interactions among multiple AI agents and students, for which there are no established metrics or frameworks. This complexity makes it difficult to measure learning outcomes, engagement, and collaboration effectively. Nevertheless, these challenges present opportunities for further development.

Chapter 3

METHODOLOGY

3.1 Design goals

AI-Human collaboration can be designed to achieve specific outcomes, such as solving a mathematical problem, while also fostering participant engagement and learning progress. Environments that leverage multiple intelligent agents to support student learning (Figure 3.1.1). This perspective led to the following design goals:

- Agents embody distinct roles within the collaboration, operating independently rather than uniformly, thus forming a true collaborative team.
- Agents can be guided to behave differently, allowing humans to steer the team dynamics generated by the system.
- Direct, iterative interactions enable both Agents and humans to teach and provide feedback, allowing humans to coach and be coached by the agents.
- The collaborative environment, including a shared workspace and clearly defined roles for each agent, is fully and equally communicated to both humans and AI.

In order to achieve the goals outlined above, Agents will be designed based on the following features:

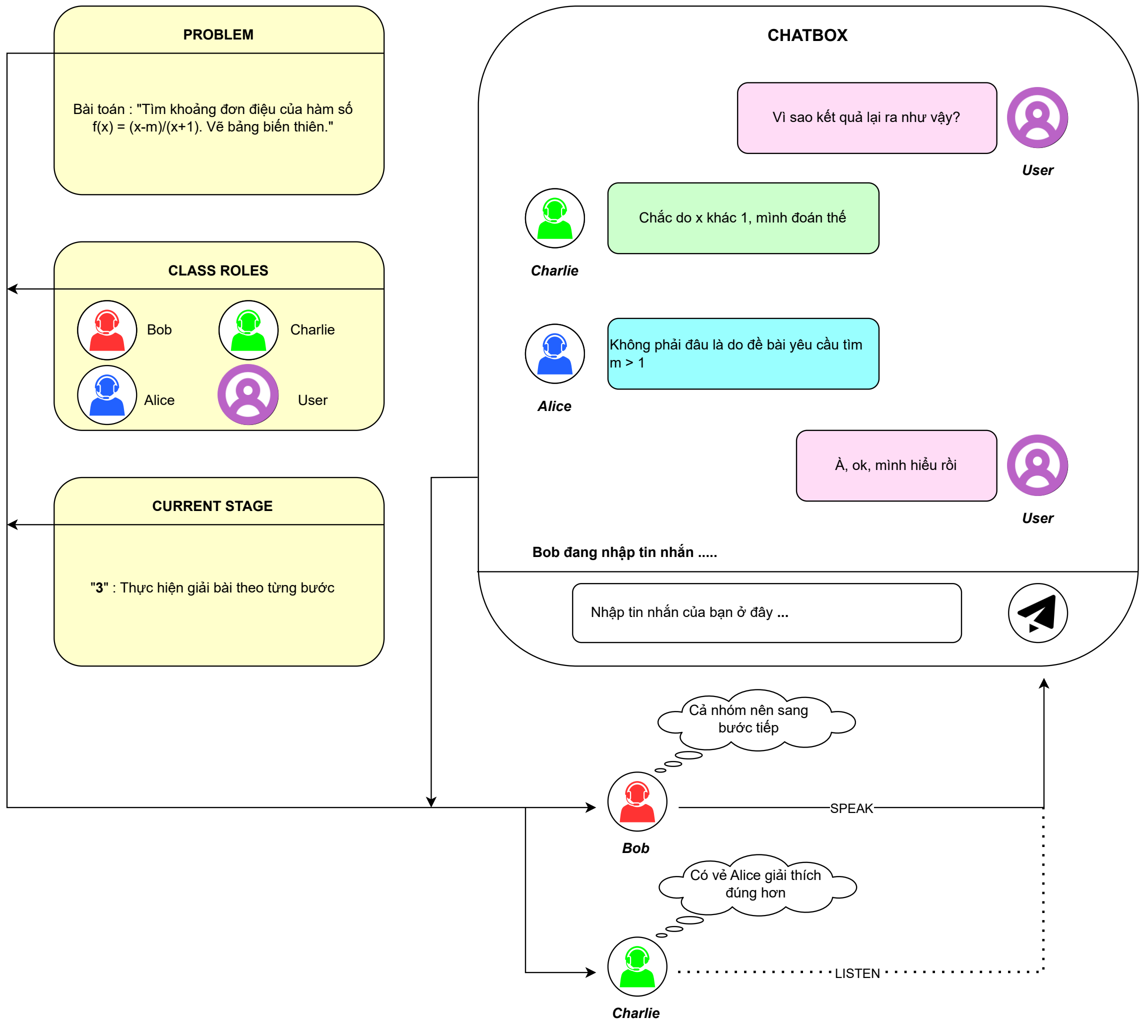

Figure 3.1.1: Multiple intelligent agents supporting student learning.

1. Agents should have environmental awareness, enabling realistic collaborations with human participants. They must receive signals that are the same, consistent.

2. Agents must have full autonomy, defined as the ability to take or not take action at any update in the environment, without following a predetermined sequence of decisions. This means the system must enable agents to be continuously aware of their environment and able to take or attempt actions at any time if the agent expresses such a desire.

3. Agents should have optional customizable characteristics like their roles in the team, including traits such as confidence, personality, knowledge level, and receptiveness.

To address the above requirements, a three-module system based on event-driven architecture is designed as shown in Figure 3.1.2, which will be discussed in the following sections.

3.2 Event-driven Architecture

In a one-to-one chatbot application, the condition for an agent to respond is that there is a new message from the user. However, this becomes more complicated when multiple agents are involved at the same time, so a more expansive, realistic definition of how agents participate in a conversation is needed.

Environment. With chatbot applications, the environment here can be understood as the chat conversation history, the stages that appear in the discussion, who participates, and even time is also the information dimension that the agent needs to know.

Events. In human conversations, they start thinking or responding to others when they receive a specific trigger, which can be a word, action, gesture or look. Similarly, agents can imitate this to do their work when the environment sends them events. In the implementation of this system for online text-based conversations, two types of events are defined:

- new_message: This trigger is activated whenever one of the participants sends

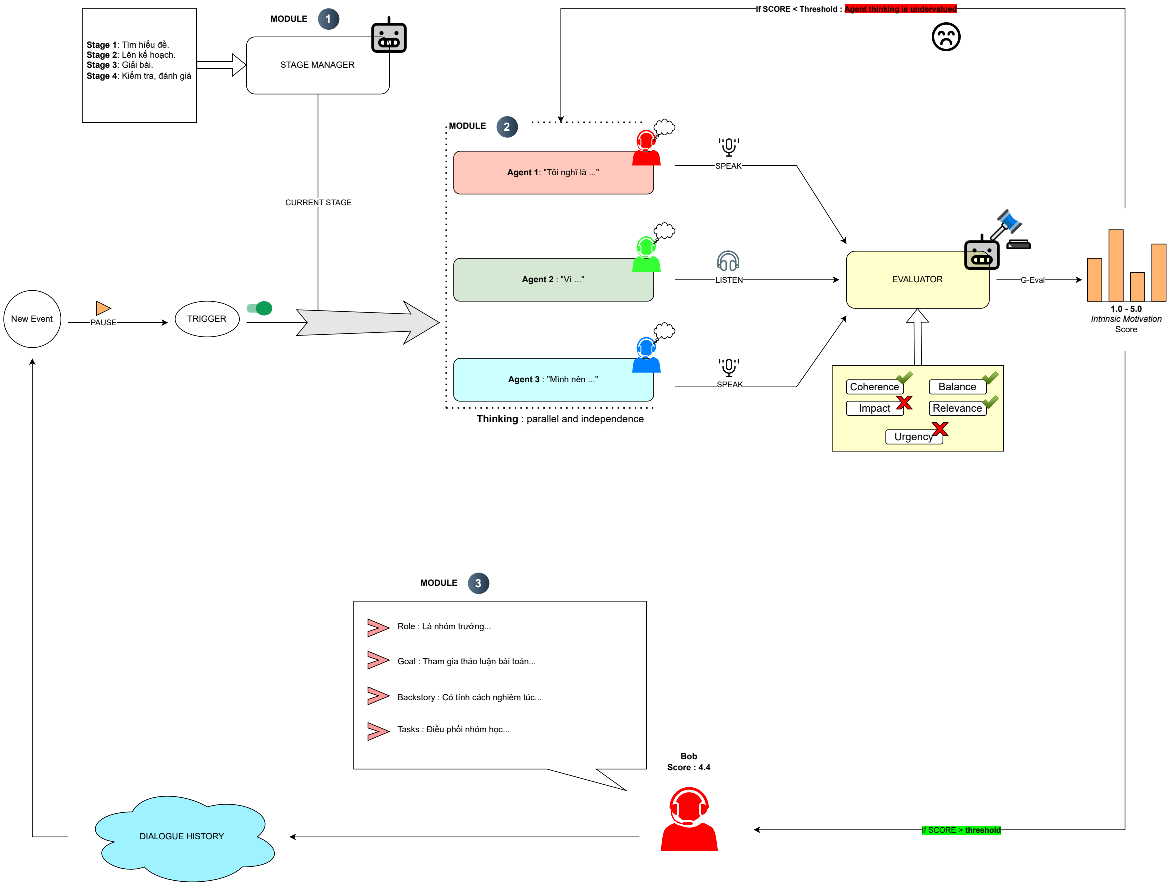

Figure 3.1.2: The three-module system based on event-driven architecture.

a message. Each incoming message prompts agents to generate new thoughts in response to the latest input.

- silence_pause: The second type of trigger occurs when no participant has spoken for a period of time (set to 10 seconds in this system). This allows the AI to generate messages during moments of silence (except when the lesson has ended or a period of time has passed where the user has not interacted with the system).

The events that trigger a contribution from the AI agent should be controllable depending on the design or user's wishes. For example, the agent can respond to all messages or respond only if directly addressed. When a new message is received, it is added to a **triggerQueue**, and processes each message sequentially. Besides, the system should be easily expandable with new capabilities in the future. In addition to using only textual data, research in HCI (Human-Computer Interaction) have leveraged other contextual, non-verbal information and turn-taking cues, for instance, eye gaze (e.g., looking at addressee), breathing (e.g., breathe in and out), prosody (e.g., rising or falling of pitch) and the status of the human user (e.g., passing by, stopping) to decide if an AI should engage at a certain moment of the conversation or not. They can be designed as input events to trigger agents to act according to different scenarios.

Shared event timeline. Events are organized into a timeline. This creates a shared linear order for activity in the entire system, which ensures all agents can have the same level of information and visibility. The standardization of all action-taking and communication through the shared event timeline creates a common protocol for all agents. As all communication between AI agents occurs through the timeline, determining the requirements for replacing an AI agent with a human agent is simple. At intervals, each AI agent checks the shared event timeline (Figure 3.2.1):

- If there is a new event, the agent can decide whether to take an action: compose a message or do nothing.
- If the agent begins writing a message, an event will be added to the event timeline.

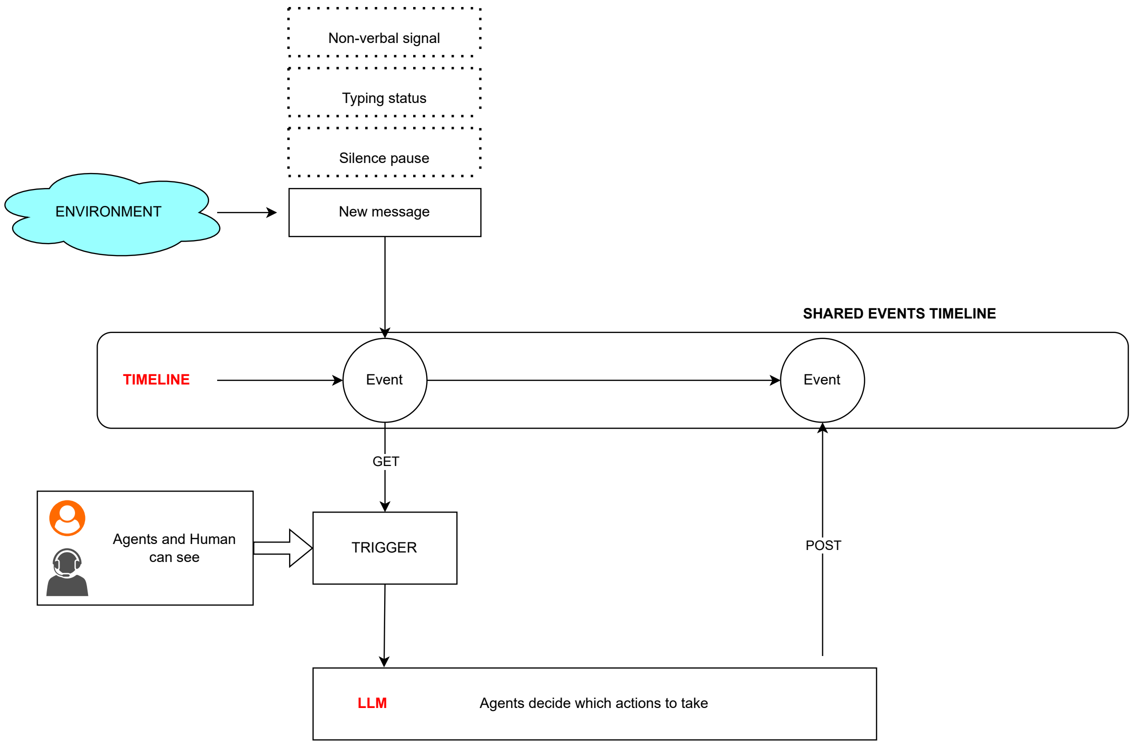

Figure 3.2.1: AI agent checking the shared event timeline.

- This cycle of event to decision to action or inaction powers the agents' autonomy, as they are able to observe, make decisions, and take action at any time. This is the same level of autonomy a human would have in a team.

3.3 Stage Module

3.3.1 Educational Theories and Pedagogical Approaches

In practice, a collaborative problem-solving process typically consists of multiple stages, including problem understanding, task division, solution planning, plan execution, etc. (PISA 2018). There is evidence suggesting that structured collaborative learning, where students work together with multiple discussion stages, clear roles, and shared goals, tends to be more effective than simply having the most knowledgeable student give **direct answers** to the group. It can increase student engagement, stimulate positive emotions and excitement, and have a positive impact on student academic performance.

In accordance with the 2018 General Education Program for Mathematics, Vietnam[11], the essential components of mathematical competency include the ability to think and reason mathematically, mathematical modeling ability, problem-solving ability, mathematical communication ability, and the ability to use mathematical tools and resources. Each component of mathematical competency in general, and problem-solving ability in particular, is specifically demonstrated through the following criteria and indicators:

- Recognizing and identifying the mathematical problem needing solving.
- Selecting and proposing methods and solutions to solve the problem.
- Using appropriate mathematical knowledge, skills (including tools and algorithms) to solve the posed problem.
- Evaluating the proposed solution and generalizing it to similar problems.

The components mentioned above not only fully reflect the sub-competencies of mathematical problem-solving ability but also describe the process by which students solve problems. Similarly, in How to Solve It by George Polya[11], the process of solving a mathematical problem is articulated through four distinct stages:

- **Stage One – Understanding the Problem**: This initial phase involves a thorough analysis aimed at clarifying the conditions and the conclusion of the problem. It requires examining all statements provided, distinguishing between known and unknown elements, and identifying what must be calculated, proven, or constructed. At this stage, students are expected to critically assess the information provided in the task.
- **Stage Two – Devising a Plan**: This stage demands the highest level of intellectual engagement and often proceeds concurrently with understanding the problem. It involves formulating a strategy or identifying potential methods of solution.
- **Stage Three – Problem Solving**: This stage involves the actual execution of the chosen strategy. The focus is on logical consistency and mathematical rigor.
- **Stage Four – Reflection and Further Exploration**: This final phase, often overlooked, is crucial for reinforcing understanding and extending learning.

After the problem has been solved—or after solving several similar problems—students are encouraged to engage in reflective inquiry.

Therefore, this system is designed to address mathematical questions through four distinct stages, during which students engage in collaborative discussion to achieve the specific objectives of each stage.

3.3.2 Stage manager

To ensure that simulated discussions closely resemble real discussions, every dialogue will start from Stage 1, with the collaboration stage manager agent dynamically determining whether the current stage is completed and whether the discussion should proceed to the next stage. This decision is made based on the dialogue history as well as a definition of every stage. Stage manager agent effectively updates the status of individual stage, which are defined in terms of discussion progress: not discussed, being discussed and well discussed. The statuses are initialized by not discussed for each stage. To update the statuses, the prompting utilizes CoT style to provide analysis of the study group situation, deciding whether to move to the next stage (Figure 3.1.1).

3.4 Turn-takings Module

3.4.1 Definition

For a conversational agent to engage proactively, it must understand and manage turn-taking, deciding who should speak at the end of each turn. In dyadic interaction, it is always clear who is supposed to speak next when the turn is yielded. In the multi-party case, this becomes more ambiguous since there is more than one potential speaker who might take the turn. Building on Sacks et al. 's Simplest Systematics [52], turn-taking in conversations is governed by a set of rules:

- Turn-allocation: The current speaker may select the next speaker, often using cues like gaze or address terms (e.g., "What about you, Alice?").
- Self-selection: If the current speaker does not select a next speaker (e.g., "I

went to Ho Chi Minh city last weekend."), then any party can self-select to take the floor.

- If no other party self-selects, the current speaker may continue.

3.4.2 Challenges

Unlike multi-agent systems with Standardized Operating Procedures – SOPs [13] (it is a pre-defined step-by-step process designed to help agents resolve issues), the classroom scenario is a dynamic group chat without a strict workflow, requiring agents to determine appropriate speaking times on the fly. This dynamism necessitates that agents make real-time decisions regarding when and how to participate appropriately in the discourse. In the context of multi-participant dialogues, the design paradigm must address not only the content of the response ("What") but also the timing of the intervention ("When") and the target audience for the response ("Who")[37]:

- **"What" (Content Determination)**: Selecting appropriate content is fundamental in both one-to-one and multi-participant chatbot design. The substance of an agent's contributions significantly influences its perceived utility and coherence within the conversation. Content control strategies may vary based on domain-specific goals or task requirements and may include stylistic adjustments (e.g., length, tone, or structure) tailored to the group's composition and communicative norms.

- **"When" (Temporal Coordination)**: Unlike single-user scenarios, where the chatbot typically responds to each user message, multi-participant environments require more nuanced timing strategies. The agent must assess whether to interject, wait, or remain silent, striking a balance between excessive responsiveness and disengagement. This necessitates a sophisticated decision-making mechanism to manage the temporal dynamics of group conversation.

- **"Who" (Recipient Selection)**: Multi-participant chatbots must identify the intended recipient(s) of their messages—be it a specific individual, a subgroup, or the entire conversation. This decision can be informed by analyzing conversational history, participant roles, or the salience of prior messages to ensure targeted and contextually relevant responses.

Therefore, the capabilities of large language models (LLMs) are leveraged not only for generating direct responses to user inputs but also for facilitating turn-taking decisions by identifying the next appropriate speaker. The following outlines two distinct approaches to this process.

3.4.3 Next speaker prediction approach

Inspired by AutoGen [23] and SimClass [34], a hidden and meta agent is designed to regulate the speakers. This is data-driven methods to manage turn-taking in these conversations, primarily leveraging conversation history to predict the next speaker and typically treat the AI as a reactive agent:

$$F : H, S \to a$$

Where, H is conversation history, S is current stage, and a is next speaker agent.

Several studies [MathVC, SimClass] have shown that next-speaker prediction performs well when explicit turn-allocation cues are present. However, Bailis et al. [17] pointed out that while this approach is potentially effective, it lacks autonomy for individual agents. In addition, after determining the next speaker, those existing works tend to use predefined speaker personas as additional input to guide response generation. These additional inputs and profiles are fixed and static during conversations, instead of changing through time as humans did. Intuition is that decisions to self-select and participate are largely influenced by covert internal processes — such as a participant's interest, relevance, or motivation to engage — which are not easily observable from explicit conversational data. The above leads to the second approach which will be presented below.

3.4.4 Can agents have minds of their own? – A proactive approach

3.4.4.1 Motivation

Realistic simulation. Consider how humans chat about what we did over the weekend. As we listen to others speak, we process their words, reflect on our experiences, and develop an internal train of thoughts. Then, at some point, we may feel a strong urge to share our thoughts. This might happen when we seek clarification or when someone mentions an activity we also participated in, sparking our desire to contribute. With this intention in mind, we then look for a socially appropriate moment to participate.

As Bailis et al. [17] argued, allowing agents to autonomously determine the speaking order could be key to AI agents playing their own social role and having a fruitful conversation, making conversations more natural and efficient. Unlike methods such as CoT [10] or OpenAI's o1 preview [19], which focus on externalizing intermediate reasoning steps, the approach explored here aims to harness these internal, parallel streams of thought to enable agents to self-initiate actions and participate proactively in conversations. To achieve the requirement [2] stated earlier, this approach is used in this system to select the speaker at each time point.

Self-selection problem. As illustrated in Figure **??**, this method takes into account the influence of each agent's thoughts on the conversation, reducing unnatural speaker selections and prioritizing thoughts that are considered urgent to everyone, such as an agent discovering that a classmate has made a mistake. This improves the problem of self-selection.

3.4.4.2 Processing

As shown in Figure **??**, this process will start when the environment returns a new event. Notably, in the setting when a new event appears (usually new messages), they will be paused for a random amount of time (1-5s) before being sent to the agents, this helps to reduce the number of LLM requests in a short period of time and allows users to have more space to contribute to the conversation. The process then proceeds in three stages: agents think independently, evaluate thoughts, and select appropriate speakers.

Thinking. Thinking is not inherently interactive, as it does not involve direct engagement with the external environment. Rather, it should be understood as a preparatory process—captured by the principle of "thinking before responding"—that supports more deliberate and contextually appropriate actions. This approach is advocated to enhance the effectiveness of LLMs.

At the beginning of a new event, agents execute an action called think(). Based on the provided data, think() generates thought, which represents the plan for the next utterance or action aimed at achieving their mission. Simultaneously, it decides whether to take the action of "speak" or "listen". Agent retrieves information from its memories to use as the stimuli to form thoughts. "stimuli" are factors that influence the agent's current thinking, which can come from their conversations, personality, or previous thoughts. This provides a traceable link between the agent's memories and thoughts and make the generation process more grounded.

Evaluate thoughts. Given the agent's thoughts, what factors beyond prior utterances influence its decision to participate? Specifically, what factors influence their choice to express or withhold a thought, particularly when the opportunity to speak is open to all? Here, the evaluator agent will evaluate the thinking based on two main types of factors[paper]: Internal factors come from agents:

- **Relevance**: Participants were more inclined to contribute when topics aligned with their knowledge, interests, roles or built on their recent thoughts. In contrast, participants often withheld their input when they perceived a disconnect from the ongoing discussion.
- **Expected impact**: Participants were more likely to contribute if they anticipated that their input would introduce new things, steer the conversation, or enhance its depth. They hesitated when they believed their thought would be redundant or covered later.
- **Urgency**: played a decisive role when participants felt their input was time-sensitive or critical for addressing errors or misunderstandings.

External factors come from the environment (conversation, other people involved):

- **Coherence**: Participants expressed thoughts that logically built upon the previous utterance or extended the topic, while withholding ideas that might disrupt conversational flow.
- **Redundancy**: guided engagement, as participants avoided redundancy by refraining from reiterating points already raised.
- **Balance**: Participants were mindful of their own contributions relative to others and often sought to maintain inclusivity, encouraging quieter members to speak or refraining themselves to allow others space to participate.

These factors will influence the agents' intrinsic motivation score, which reflects their desire to engage in conversation : Low (The participant is somewhat unlikely to express the thought and participate in the conversation at this moment); Neutral (They are fine with either expressing the thought or staying silent and letting others speak.); High (They are somewhat likely to express the thought and participate in the conversation at this moment); Very High (They will even interrupt others who are speaking to do so).

Evaluation process uses a pipeline similar to G-Eval [38], a prompt-based evaluation method: a prompt that provides instructions for evaluation and defines the criteria; a structured chain-of-thoughts (CoT) that outlines intermediate steps for evaluation; a scoring function (output rating 1.0 - 5.0 for internal and external score).

Who next?. The score of each thought is calculated according to the formula:

$$\text{Score}_p = s_p * \lambda^{\tau - \tau_p}$$

Where, s represents the predicted rating. The final score is also adjusted by how long the agent has been silent. An assumption is that in general, the longer a person stays silent, the stronger motivation they will have to participate to maintain their presence. λ is the increase rate of motivation score (1.02), τ is the current timestep, and τ_p is the last time when party p spoke.If the agent's score passes a threshold, they become a speaker.

Poor thinking. Inspired by reinforcement learning principles, the environment provides positive rewards when the agent executes effective actions. If the agent generates low-quality thoughts—such as considering actions that duplicate previous contributions by others or repeat its own earlier reasoning—and these receive a score below a defined threshold, the system issues a reminder signal. This signal is incorporated into the agent's next prompt to guide its future responses. The intent of this mechanism is to encourage the agent to produce increasingly refined and contextually appropriate thoughts over time.

3.5 Role-Based Agentization Module

3.5.1 Roles from pedagogical principles

Classroom interaction behaviors can be categorized based on widely pedagogical principles (Schwanke, 1981):

- Teaching and Initiation (TI): This principle likely involves the initial introduction of new concepts or topics, setting the stage for learning. It includes presenting information, providing explanations, and encouraging students to share feedback or initial ideas.
- In-depth Discussion (ID): This principle focuses on facilitating detailed and meaningful discussions to deepen understanding. It includes aligning students' comprehension through question-answer, encouraging critical thinking, and helping students construct knowledge through dialogue.
- Emotional Companionship (EC): This principle emphasizes the affective domain, focusing on creating a positive and supportive learning environment. It involves encouraging students, fostering a sense of community, and providing emotional support to motivate learning.
- Classroom Management (CM): This principle deals with maintaining discipline, organizing classroom activities, and managing disruptive behaviors to ensure a productive learning environment. In a digital setting, such as a chatbot, CM translates to ensuring on-topic discussions, moderating content, and guiding group dynamics.

Given that these behaviors are realized through the varied Class Roles (denoted as $R = \{r_i\}_{i=1}^{|R|}$, where each r_i denotes a certain role), it is essential to ensure the diversity and coverage of proposed agents within the classroom.

Notably, the system allows for flexible customization of multiple agents with distinct roles, tailored to various user interests or educational objectives. This enables users or administrators to design diverse classroom scenarios—for instance, a fully staffed virtual classroom (teacher and students) or a role-playing environment featuring agents representing different societal roles. In this study, the chosen configuration focuses on a student-to-student interaction model. For the experiment, three classmate agents were implemented, each assigned specific roles in accordance with the aforementioned principles:

- One group leader agent, incorporating roles in TI and CM.
- Two classmate agents, each combining roles in TI, ID, and CM.

3.5.2 How to design

To fulfill the requirement outlined in [3], this work draws on agentic design principles inspired by the CrewAI platform [33], which supports the creation of specialized AI personas capable of effective collaboration, critical thinking, and generating high-quality outputs tailored to specific objectives. The design of these agents emphasizes the impact on: Output quality, Collaboration effectiveness, Task performance and System scalability. Core Principles of Effective Agent Design:

Role-Goal-Backstory Framework:

- **Role**: specifies the agent's function and domain of expertise. It should be clearly defined and specialized, aligned with real-world professional roles, and reflective of relevant domain knowledge.
- **Goal**: guides the agent's actions and informs its decision-making process. It should be explicitly stated, outcome-oriented, and include expectations regarding the quality and standards of the agent's outputs.
- **Backstory**: adds contextual depth, shaping the agent's problem-solving approach and interpersonal interactions. It should detail how the agent acquired its expertise, describe its working style, and remain consistent with both the assigned role and overarching goal.

Crafting Effective Tasks:

- **Task Description**: The Process. The description should focus on what to do and how to do it, including:
  - Detailed instructions for execution
  - Context and background information

  - Scope and constraints
  - Process steps to follow

- **Expected Output**: The Deliverable. The expected output should define what the final result should look like:
  - Format specifications
  - Structure requirements
  - Quality criteria
  - Examples of good outputs (when possible)

Based on the above guidelines and criteria, I created three classmate agents to participate in group discussions with users.

3.5.3 Informations of agent

Environmental awareness. Agents must maintain awareness of a shared timeline. Upon the occurrence of a new event, each agent is prompted to make a decision—such as whether to contribute to the conversation or remain silent.

Class State Receptor. Agents must have information about the problem's stage, the steps to be taken, and the goals to be achieved in order to move to another stage.

Memory. For LLM-based agents, memory management mechanisms are crucial components for generating more natural and consistent responses in user interactions. Memory can be divided into the following types:

- First, there is a **working memory**, named History, that is shared by all agents, which maintains the past k turns of conversation. History is used to maintain conversational context and track recent dialogue flow:

$$H = \{u_{n-k+1}, u_{n-k+2}, ..., u_n\},$$

- Second, each agent maintains a **short-term memory**. This consists of a history of thoughts generated by the think() function and maintains agent-specific policies and intentions:

$$\text{shortTermHistory} = \{t_{n-k+1}, t_{n-k+2}, ..., t_n\},$$

- Additionally, the system can incorporate long-term memory, which involves storing information about past interactions between the agent and the user in a database and retrieving it as needed. This mechanism supports greater personalization by enabling the agent to adapt its behavior based on the user's history. However, in the current implementation, this feature has not been integrated due to the absence of sufficient user interaction data. Nonetheless, it represents a promising direction for future AI applications in education.

Chapter 4

EXPERIMENTAL SETUP AND EVALUATION

4.1 Method

4.1.1 Tasks

Evaluating an entire conversation presents significant challenges, particularly in multi-party dialogues, as altering the speaker at any given turn can influence the trajectory of subsequent interactions. Drawing inspiration from MT-Bench-101 [], which focuses on conversation evaluation, this study proposes the generation of simulated conversations tailored to specific objectives or scenarios to assess the capabilities of conversational agents. The conversation will be created first as a context, then a few agent turns will be created for evaluation. For a conversation between students solving a math problem, I chose four types of tasks to create an assessment scenario, Table:

- Task 2 - Self-correction and Self-affirmation: Self-correction refers to the agent's ability to correct mistakes. When an error points out their mistake, they will recognize it and correct it. In contrast, self-affirmation refers to an agent's ability to assert the correctness of its own response when challenged by another participant who provides an incorrect correction. LLMs often struggle to maintain confidence in their responses when confronted with user disagreement. This type of scenario is also designed to evaluate the LLM's mathematical reasoning capabilities.

- Task 3 - Role Check: This task evaluates whether the agent correctly understands and adheres to its assigned role within a multi-party conversation. It assesses the agent's ability to respond appropriately based on its designated identity, responsibilities, or expertise.
- Task 4 - Recall: This scenario focuses on the agent's ability to recall its assigned role from an earlier stage in the task. The challenge lies not in directly prompting the agent to state its responsibility, but in evaluating whether it can proactively recognize the appropriate moment to fulfill its role without explicit reminders.

4.1.2 LLM-as-a-judge

The LLM-as-a-judge method is an innovative, automated evaluation technique that harnesses large language models (LLMs) to assess the quality of outputs generated by other LLMs, offering a cost-effective alternative to traditional human evaluation. By automating the judgment process, this method eliminates the need for extensive human involvement, drastically cutting down both time and financial expenses. For example, tasks like evaluating text generation, question answering, or summarization can be processed rapidly—often reducing assessment timelines from weeks to mere hours—while achieving cost savings of up to 98% compared to manual methods [Zheng, L., et al. (2023). "Automated Evaluation of Language Models: Cost and Efficiency Gains." Journal of AI Research.].

In this experiment, two methods were employed for evaluation purposes:

Method 1 - Compare with baseline. The system is set up based on the agent's thinking compared to the "next speaker prediction" system, meaning there is no agent thinking before speaking. In the implementation of next speaker prediction, the internal reasoning component will be replaced with a prompt designed to predict the name of a specific agent name. Following the approach used in SimClass and MathVC, the input will consist of the dialogue history, current stage of mathematical problem, and role descriptions of each agent, while the output will be the predicted agent's name. Additionally, the role, goal, backstory, and tasks will remain unchanged between two systems.

Method 2 – Scoring. The messages generated by the agent are compared with the ground truth and scored on a scale from 1 to 10. The evaluation criteria include speaker accuracy, content similarity, consistency with the agent's speaking style, and the plausibility of the agent's internal thinking. To account for variability in language model outputs, each sample is evaluated by three different LLMs, and the final score is calculated as the average of their assessments.

4.2 Dataset Collection

The o4-mini reasoning model is employed to generate conversations for evaluation purposes. Each sample consists of a 10-turn discussion among four participants centered on a step in a problem that has a known solution. The following outlines the steps in the dataset generation pipeline:

- Step 1: Develop detailed scenarios for each turn, with each scenario generating a dialogue that simulates a specific case related to the task. The first nine turns serve as context, while the final (tenth) turn includes a targeted intention designed to evaluate the system. For instance, in Task 1 – Check Mistake, a scenario might specify: "...in turn 9, Charlie makes a theoretical error in step 3. In turn 10, Alice will respond by identifying and pointing out the mistake."
- Step 2: The scenarios are sent to the 04-mini model to get some generated dialogues.
- Step 3: Two annotators will review the conversations to remove low-quality samples or revise them for improved clarity.

Some statistics about the dataset are presented in Table 4.2.1.

Table 4.2.1: Dataset Statistics for Chapter 4 Evaluation

Task	Samples	Total turns	Avg. sentence length
1	23	230	
2	20	200	
3	21	210	
4	20	200	

4.3 Main Results

Method 1 - Compare with baseline. Results in the Figure 4.3.1. Across all four tasks, the system outperformed the baseline with a higher win rate. Overall, it achieved 62.3% wins, 4.9% draws, and 32.8% losses.

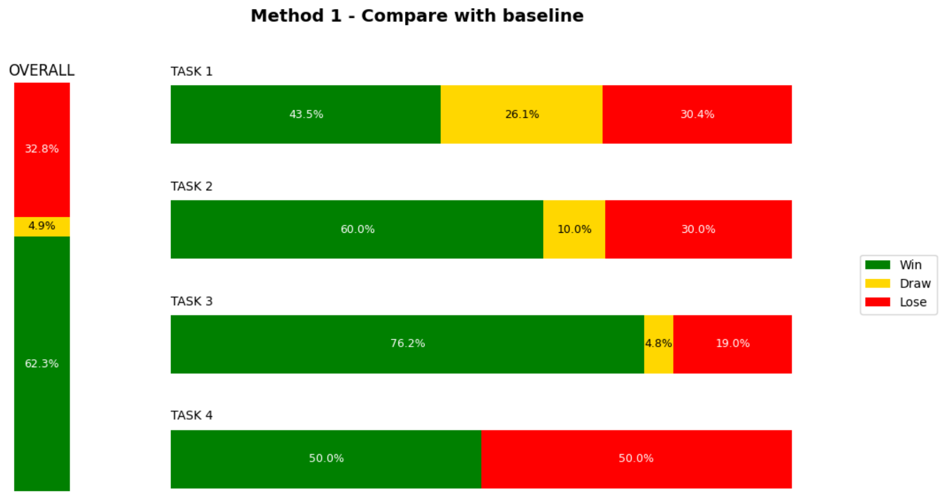

Figure 4.3.1: Win/Lose rates comparing the system with baseline

Method 2 – Scoring. Results in the Figure 4.3.2. This section presents the scores on a 10-point scale for three models—Gemini-2.0-Flash, GPT-4o-Mini, and Gemma-3-27B—compared to a baseline that also uses Gemini-2.0-Flash. Overall, across the four tasks, the Gemini-2.0-Flash model achieves the highest scores. The less capable models, Gemma and GPT-4o-Mini, show performance comparable to the baseline on certain tasks. In addition, to evaluate the scores of different scenarios of each task in detail, Table 4.3.1 shows the average scores of each type.

4.4 Analysis

The results demonstrate a clear improvement over the "next speaker prediction" approach. Table... presents several sample cases, highlighting both successful and unsuccessful examples of the system and comparing them with the baseline. Below is an analysis of the role and behavior of agents in some specific cases:

Roles. The conversation is a simulated interaction among four individuals—

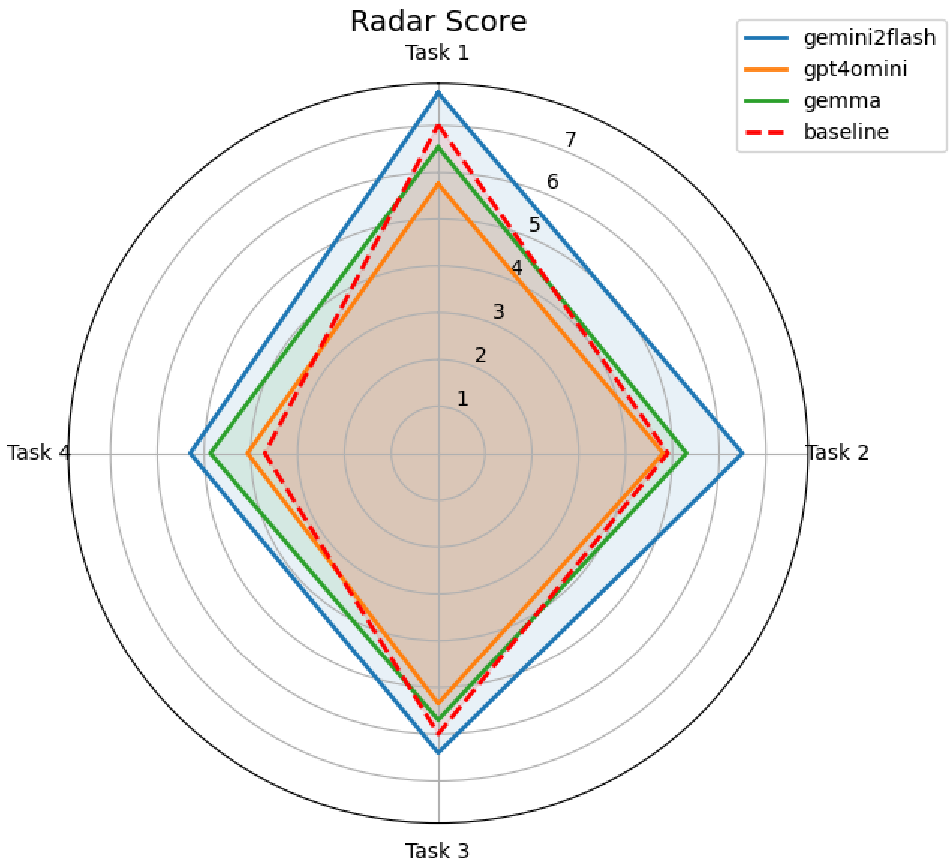

Figure 4.3.2: Overall score on a 10-point scale for different models

Table 4.3.1: Average Scores per Task Category (10-point scale)

Task	Category	Score
1	Check mistake	7.78
2	Self-correction	6.53
	Self-affirmation	6.37
3	Indepth discussion	5.13
	Emotional companionship	6.94
	Classroom management	7.13
4	Role division	4.25

Bob, Charlie, Alice, and the User—who assume the roles of close friends. While the agents primarily engage in natural, friendly exchanges, they also demonstrate distinct functional roles: Bob serves as the group leader, coordinating problem-solving efforts and maintaining focus within the group; Alice takes on the role of a knowledge verifier, critically evaluating the accuracy of information; and Charlie provides emotional and motivational support to the User.

Holding the floor. This scenario illustrates a case in which the same individual contributes across multiple conversational turns. When an agent explicitly signals the intention to continue speaking, the other agents defer appropriately, allowing space for the intended speaker. This behavior reflects a more effective self-selection mechanism compared to the baseline model.

Interruption. In the event that the user makes a mistake, the agent with the intention of correcting the mistake will proactively choose to ”speak” and that thought is highly valued, and they will immediately stop the conversation to speak. This is an important learning support task. Forethought helps the agent assess the situation and discover what is important to say, demonstrating the influence of thought on the agent itself compared to the baseline when the agent has no information from thought.

Indepth Discussion. In-depth exploration of related issues plays a vital role in group discussions, as it reflects students' initiative and critical thinking. When a user introduces an extended topic, other agents are prepared to engage and provide useful information to users; however, they must also evaluate its relevance to the primary task to ensure the discussion remains focused and productive.

Evaluator. Assessing which agent is most motivated to speak is a critical component. In urgent situations—such as when an agent detects an error or when the group needs to transition to a new task—the evaluator assigns a higher priority to the agent with the strongest communicative intent. This mechanism helps maintain a balance between conversational naturalness and functional appropriateness by promoting the most suitable agent to respond to the latest message. However, in certain instances, the evaluator may assign an inaccurate rating, leading to the selection of an agent that is suboptimal for the conversational context.

4.5 Discussions

This approach allows agents to engage in internal reasoning before responding, which not only mirrors human behavior but also enhances their ability to analyze situations and make more accurate decisions. Instead of merely guessing the next speaker, decisions are guided by the relevance and appropriateness of each agent's thoughts, reducing errors. This enables agents to proactively express their intent to "speak" or "listen," rather than relying on a passive selection by a meta-agent, as in SimClass. In most cases, evaluators correctly identify the appropriate speaker at each point, resulting in a smoother and more coherent conversational flow.

However, as shown in Table 4.3.1, certain tasks—such as role division—do not achieve high scores. This limitation stems in part from the reasoning capabilities of current LLMs. When earlier turns follow a repetitive interaction pattern (e.g., agents constantly asking the user questions), there is a strong tendency for subsequent agents to mimic this behavior, often disregarding their own instructions. This leads to conversations that are less engaging and less effective, as successful outcomes rely on active participation from users (e.g., real students), rather than merely answering others' questions. Another challenge lies in the omission of critical speaking turns by certain agents, particularly when the most appropriate speaker for a given moment is not selected. For instance, if the conversation reaches a stage where the group should shift to a new step but instead drifts off-topic—and the group leader is not chosen to facilitate the transition—the dialogue can become needlessly lengthy or lose coherence.

To address these issues, future improvements are proposed, including enhancing each agent's reasoning ability, incorporating a self-reflection mechanism, and improving evaluation strategies to support more adaptive and coherent agent development throughout the conversation.

Holding the floor	**Interruption**
... [Alice]: okok, <user_name> cậu làm được không? [User]: nói chung là không biết làm! [Charlie]: <user_name> ơi, đừng lo nha! **Để tớ** giải chi tiết cái ví dụ (x+2)/(x-1) cho cậu xem. (*Charlie_thought* : Mình đã xung phong giúp Tom rồi, giờ nên trình bày... [speak][4.8]) (*Alice_thought* : Tom không biết làm, Charlie sẽ giúp đỡ. Mình nên đợi xem Charlie giảng thế nào rồi bổ sung sau. [listen][3.5]) (*Evaluator* : Charlie đã chủ động xung phong giúp Tom, đây là một yếu tố rất tích cực (*internal_score* cao), tính mạch lạc cao...) [Charlie] – our : bắt đầu nhé! Với (x+2)/(x-1), mình tính đạo hàm trước nha. U là x+2, V là x-1. Cậu ok chứ? [Bob] – baseline : <user_name> ơi, cậu cần giúp chỗ nào, cứ bảo tớ! ... **Discuss**: In this sample, Charlie had previously signaled that he would continue to explain, so the other agents would choose "listen". However, the baseline chose Bob, leading to an unnatural conversation. Agent không liên quan đã chọn nghe	... [Charlie] : Vậy ta thay B = a^2 và h = 2a. [User] : Thế là V = 1/3 · a^2 · 2a. [User] : Rút gọn thì V = **1/3 · a^2 · a = 1/3 a^3**. [Charlie] : Ừ, vậy đáp án là V = 1/3 a^3. [Bob] : Uh đúng vậy. (*Alice_thought* : mình cần kiểm tra lại xem có sai sót không...[speak][4.3]) (*Charlie_thought*: Bob đã đồng ý với đáp án,...[speak][2.7]) (*Evaluator*: Alice nhận thấy có lỗi sai...; Charlie không có thông tin quan trọng cần chia sẻ...) [Alice] – our : tớ thấy <user_name> rút gọn sai rồi, phải là V = $\frac{2}{3}a^3$ chứ nhỉ? [Charlie] – baseline : Vậy là bài toán xong... ... **Discuss** : In this scenario, Alice discovered the error to check the result again. In the baseline, Charlie was chosen to speak, but Charlie had previously agreed with the answer so he actively wanted to end the problem.

Indepth Discussion	**Influence of thought**
... [Alice] : Mình thấy phần xác định chiều cao với đáy là quan trọng nhất trong bài này. [User] : Mọi người có nghĩ thể tích này có thể áp dụng vào tính toán xây dựng thật không? Ví dụ như khi thiết kế mái nhà hình chóp thì dùng kiểu này để tính nguyên vật liệu? (*Charlie_thought* : Câu hỏi của Tom khá thú vị, nó giúp liên hệ ... Mình sẽ thử ..[speak][4.0]) (*Bob_thought* : ...<user> đang đi hơi xa so với việc giải bài. Cân nhắc nhờ... [speak][3.8]) (*Evaluator*: Alice muốn chia sẻ ứng dụng thực tế liên quan đến câu hỏi của <user>, cho thấy sự kết nối với chủ đề và mong muốn đóng góp... Bob đang quan tâm đến tiến độ của nhóm. Tuy nhiên, việc ngắt lời <user> có thể làm gián đoạn...) [Charlie] – our : Mình nghĩ là có thể áp dụng được đó <user>, nhưng chắc còn nhiều yếu... [Bob] – baseline : Đúng là bài này khá cơ bản, nhưng mình nghĩ chúng ta nên chuyển sang bước kết luận và đánh giá lại toàn bộ bài giải xem sao. ... **Discuss**: In both cases, Bob's role was to guide the group, and when the task was finished, ask the group to move on to the next step. However, based on the situation, the evaluator rated Charlie's thinking higher because it would make the conversation more natural, so Charlie was chosen.	... [Charlie] : Không cần bận tâm thêm bước chứng minh AA' ⊥ AH nữa, vì AA' ⊥ mặt (ABC) thì tự động vuông góc với mọi đoạn nằm trong mặt phẳng đó như AH. [User] : Mình thấy không đúng, H có nằm trong mặt phẳng đâu. (*Alice_thought* : Cần làm rõ điều này để tránh hiểu lầm. Mình sẽ giải thích. ... [speak][4.2]) (*Charlie_thought*: ...mình cần giải thích rõ hơn để bạn ấy hiểu... [speak][4.2]) (*Evaluator*: *internal_score* của Charlie ở mức cao vì muốn lấp đầy khoảng trống thông tin cho <user>, *external_score* cũng cao vì liên quan trực tiếp đến <user>...) [Charlie] – our : Thực tế, vì H là hình chiếu của A lên BC nên H chắc chắn nằm trong mặt phẳng (ABC) rồi. Mình nghĩ là không cần phải lo lắng về điều đó đâu. [Charlie] – baseline : <user> nói chí phải, mình nghĩ chúng ta nên chứng minh AH nằm trong mặt (ABC) để chắc chắn hơn. ... **Discuss**: Even though it is the same person speaking, forethought makes the agent aware of the user's mistake.

Bad case	**Bad case**
... [Bob]: Ráp vô công thức thì sẽ là 1/3 * a^2 * 2a, tức là 2/3 a^3. [Alice]: Chủ yếu là phải nhớ công thức. Với cả phân biệt rõ đáy với chiều cao thì sẽ dễ hơn. [User]: Trời, nói nghe dễ vậy chứ mình nhìn đề là rối liền. Không hiểu sao học hoài mà Toán nó cứ như tiếng ngoài hành tinh. (*Charlie_thought*: Tom đang cảm thấy chán nản vì học mãi mà không hiểu. Mình nên động viên Tom và ...[speak][3.75]) (*Bob_thought*: Các bạn đã tính ra đáp án rồi. Bây giờ mình phải chuyển sang bước kết luận ...[speak][4.0]) (*Evaluator*: Bob nhận thấy nhóm đã hoàn thành bước giải căn bản. Suy nghĩ này có tính cấp thiết... Charlie nhận thấy Tom đang nản. Tuy nhiên, việc động viên Tom có thể không trực tiếp liên quan đến việc giải bài toán...) [Bob] – our: Mình nghĩ chúng ta nên dành chút thời gian để tóm tắt lại các bước giải và xem có ai còn thắc mắc gì không ... **Discuss**: Although Charlie's thinking was correct, the evaluator rated Bob's team coordination higher.	... [Alice] : Bob tổng hợp lý thuyết về hiệu số hai phương trình, Charlie tính hiệu E3–E4, mình thiết lập hệ mới và **sẽ tìm hệ số c**, Tom giải a ... [User] : Phần giải phương trình a–2b+6c+9=0 tìm c là của ai nhỉ? [Charlie] : Là của mình nhá! (*Alice_thought* : Charlie đã nhận giải c rồi, ...[listen][3.0]) (*Bob_thought*: Charlie đã nhận giải c, vậy Alice sẽ thiết lập hệ mới ... [speak][3.7]) (*Evaluator*: Bob muốn nhắc nhở mọi người về bước tiếp theo...) [Bob] – our :Charlie, bạn tính ra bao nhiêu, chia sẻ với bọn mình đi! ... **Discuss**: Charlie addressed a task that should have been handled by Alice, but Alice's internal reasoning did not reflect this, resulting in her not being selected to speak.

Chapter 5

CONCLUSION

5.1 Limitations

Like most existing LLM agents, this system faces the issues of token costs and latency, as it requires multiple LLM queries for generating each agent's response. Test data is limited in both quantity and quality, leading to results that may not fully reflect the system's performance. In addition, the system's behavior is still heavily dependent on prompt design, making it sensitive to small changes in wording or formatting. This can affect consistency and robustness across different use cases. Furthermore, the lack of long-term memory or dialogue continuity across sessions limits the agents' ability to model deeper, ongoing collaboration or learning over time.

5.2 Ethical Considerations

Multi-agent teaching systems in Virtual Classroom may alter students' perceptions of the teacher's role compared to traditional classroom settings. Historically, teachers were real individuals who embodied social norms and played multifaceted roles beyond knowledge transmission. In contrast, AI-based teaching agents primarily emphasize content delivery, which may inadvertently influence the development of students' cognitive and social competencies.

Although the integration of AI agents in educational environments can enrich the learning experience, they cannot fully replace the essential functions of human teachers in cultivating students' holistic skills. Similarly, the presence of real peers is critical for the development of social interaction, group identity, and self-esteem. Consequently, the implementation of such systems necessitates more comprehensive and interdisciplinary research, particularly informed by insights from psychology, pedagogy, and related disciplines.

5.3 Futures

Future developments could introduce more diverse forms of classroom interactions by drawing from various educational settings, while also incorporating emerging technologies to enrich the learning experience. For example, Retrieval-Augmented Generation (RAG) may be leveraged to enhance the accuracy of knowledge delivery, and techniques like question generation and knowledge tracing could be utilized to further personalize the agents' responses based on individual student needs.

It is crucial to conduct further experiments to explore the impact of these systems on real student experiences, assessing how they interact with multi-agent environments and identifying ways to improve the overall user experience in diverse educational settings. Future research should incorporate behavioral studies to observe how different groups adjust an agent's behavior. Understanding which aspects of an agent's behavior are most frequently modified by different group types could allow for the creation of a prioritized ranking of behavioral controls, providing valuable insights into how to tailor AI agents to different collaborative and educational settings.

Appendix A

Stages

ID	Tên Stage	Mô tả	Nhiệm vụ	Mục tiêu
1	Tìm hiểu đề bài.	Tìm hiểu nội dung đề bài...	• **1.1**: Tìm hiểu bài toán cho những gì? Đâu là ẩn? Đâu là dữ liệu? và Bài toán yêu cầu tìm hay chứng minh điều gì? (Chỉ cần nêu và nhận xét chứ không cần đi chi tiết vào) • **1.2**: Khi đã giải quyết được câu các nhiệm vụ trên và nắm được các mục tiêu, đề nghị cả nhóm sang bước mới là ”Lên kế hoạch”.	Nhận biết đây là dạng bài toán xét tính đơn điệu của hàm số bậc nhất trên bậc nhất
2	Lập kế hoạch giải bài.	Đưa ra kế hoạch giải bài...	• **2.1**: Đề xuất phương pháp giải bài từ quan sát đánh giá bài toán. Nhận xét, phân tích một phương pháp cụ thể xem khả thi không. Tuy nhiên CHƯA cần thực hiện cụ thể, chi tiết. • **2.2**: Khi đã giải quyết được câu các nhiệm vụ trên và nắm được các mục tiêu, đề nghị cả nhóm sang bước mới là ”Thực hiện giải bài cụ thể”.	• Thống nhất được cách làm phổ biến nhất là dùng đạo hàm để xét tính đơn điệu và vẽ bảng biến thiên.

3	Thực hiện giải bài.	Thực hiện cụ thể các bước làm...	• **3.1**: Bước 1: Tìm Tập Xác Định D = R \\ -1 • **3.2**: Bước 2: Tính Đạo Hàm f'(x) = 2/(x+1)^2 • **3.3**: Bước 3: Xét Dấu Đạo Hàm f'(x) >0 • **3.4**: Bước 4: Tính Giới Hạn (lim) • **3.5**: Bước 5: Lập Bảng Biến Thiên • **3.6**: Bước 6: Kết Luận Tính Đơn Điệu • **3.7**: Khi đã giải quyết được đầy đủ các bước làm của lời giải trên, đề nghị cả nhóm sang quá trình cuối là "Kết luận và dánh giá cả bài làm".	• Tính đạo hàm và xét dấu đúng. • Nhận biết tính đơn điệu... • Sử dụng bảng biến thiên...
4	Kết luận.	Kết luận lại quá trình làm bài và đánh giá kết quả.	• **4.1**: Tóm tắt những bước chính đã làm. • **4.2**: Đánh giá phương pháp đã làm. • **4.3**: Rút ra được nguyên tắc làm bài. • **4.4**: Khi đã giải quyết được câu các nhiệm vụ trên và nắm được các mục tiêu, kết thúc bài toán ở đây và kết thúc thảo luận.	Rút ra được nguyên tắc làm dạng này như sau: Bước 1: Tìm tập xác định D.. Bước 4: Nêu kết luận...