2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #52561

Educational AI Agent for Student Team Interactions and Analysis

Yuxin Ren, New York University Tandon School of Engineering
Mvano Michael Cabangoh, New York University Tandon School of Engineering
Prof. Rui Li, New York University

Dr. Li earned his master's degree in Chemical Engineering in 2009 from the Imperial College of London and his doctoral degree in 2020 from the University of Georgia, College of Engineering.

Educational AI Agent for Student Team Interactions and Analysis

Abstract

This evidence-based paper presents the development of a multi-conversational AI agent designed to support student teams in managing collaborative projects more effectively. Student groups often face challenges such as communication breakdowns, lack of continuous support, and early-stage team conflicts. Current approaches, such as instructor interventions and predictive modeling, provide partial solutions but lack real-time, personalized engagement.

Our proposed system integrates an AI agent within a Discord-based environment to monitor conversations, identify potential conflicts, and generate feedback for both students and instructors. Using Natural Language Processing (NLP) techniques, the AI agent summarizes discussions, recognizes individual contributions, and maintains contextual memory across conversations.

All the group conversations will be logged into our PostgreSQL database. Then we will conduct an NLP sentimental analysis to check for the team member's satisfaction. Every team with a score less than or equal to zero will be noted. Their logged conversations will be checked for team inefficiencies using the CATME five team work dimensions (Contributing to the Team's work, Interacting with Teammates, Having Relevant KSAs, Expecting Quality, Keeping the Team on Track) using NLP text classification. We'll note any low scores in any dimension. Their logged information will further be summarized using NLP abstractive summarization. Then, the professor will then be notified via email about the team inefficiency with the logged team conversations along with the highlighted low scores in those five dimensions.

We use a database in a HPC for easier retrieval and uploading of the logs that the students make in the team chat. We plan to use a vector database to better capture the context when a regular database reaches its limit especially in retrieving information from a long time.

The study follows a phased methodology, starting with two-member test groups, extending to larger teams, and evaluating the AI agent's performance in understanding, recall, and contextual feedback generation.

Expected outcomes include improved communication flow, better conflict resolution, and enhanced team productivity through automated feedback and instructor insights. This work contributes to the field of AI-assisted educational collaboration by bridging social interaction analysis and intelligent facilitation in group learning environments.

Introduction

Collaborative team-based learning has become a defining feature of modern engineering education, particularly as institutions transition toward project-based curricula, multi-semester design studios, and interdisciplinary experiences. Industry and accreditation organizations alike increasingly emphasize the need to develop communication, coordination, conflict management, and shared decision-making skills alongside technical expertise [1]. The rapid expansion of project-centered learning through vertically integrated projects, makerspaces, and undergraduate research has intensified the need for instructional frameworks that not only evaluate teamwork but actively support student collaboration [2], [3]. The challenges that students encounter in group environments persist across contexts and disciplines [4].

Studies show that uneven contribution, unclear delegation, conflict avoidance, and unequal work distribution routinely undermine learning outcomes [5]. Communication breakdowns represent one of the most common failure modes in undergraduate teams, often surfacing only after deadlines have passed or projects have stalled [6]. Moreover, instructors frequently lack visibility into team functioning because collaboration now occurs in digital spaces—Slack, Discord, Teams, GroupMe, WhatsApp—rather than in classrooms or laboratories [7]. This shift has created a significant information gap: while students generate rich conversational data documenting their coordination and conflict, instructors rarely have the time or capacity to review hundreds of asynchronous messages [8].

To address this need, the engineering education community has adopted structured tools such as the Comprehensive Assessment for Team-Member Effectiveness (CATME). CATME enables peer evaluation grounded in five validated teamwork dimensions: contribution, interaction, keeping the team on track, expecting quality, and possessing necessary knowledge, skills, and abilities [9]. Numerous studies confirm that CATME can detect early interpersonal breakdowns and support adjustment of team processes [10], [11]. However, CATME assessments are periodic and retrospective—they identify problems only when prompted by survey administration, not at the moment communication breaks down [12]. This creates a persistent need for continuous monitoring tools capable of surfacing issues before they escalate.

Recent progress in artificial intelligence (AI), natural language processing (NLP), and large language models (LLMs) offers a transformative opportunity to analyze team interactions as they unfold. Advances in transformer-based models allow automatic analysis of sentiment, contribution patterns, and emotional tone in chat environments [13]. Educational research has demonstrated the feasibility of using machine learning to classify group roles [14], summarize asynchronous group discussions [15], detect early warning signals of team dysfunction [16], and scaffold student reflection [17]. Emerging systems show that LLM-powered tools can coach students through collaboration challenges, help resolve misunderstandings, and generate real-time support aligned with teamwork frameworks [18], [19].

Within engineering education specifically, AI-driven teaching assistants have been deployed to facilitate design team discussions [20], summarize communication logs [21], and provide formative feedback at scale [22], with encouraging evidence that students perceive these agents as accessible, supportive, and nonjudgmental [23]. This aligns with broader findings showing that students often feel more comfortable communicating limitations, confusion, or interpersonal concerns to automated tools than to instructors [24]. Such systems hold potential not only to augment instructional bandwidth but also to democratize support by reaching students who might otherwise remain silent [25].

Despite this promise, existing applications of AI-supported collaboration typically face three key limitations. First, many tools lack persistent conversational memory and therefore cannot maintain context across lengthy, multi-week project threads [26]. Second, systems rarely integrate validated teamwork frameworks such as CATME into automated classification pipelines [27]. Third, most interventions remain student-facing rather than instructor-facing, meaning that educators still lack timely visibility into which teams are struggling [28].

The present work addresses this gap by designing and deploying a multi-conversational AI agent integrated into a Discord collaboration environment to assist student teams across a project-based engineering course. The agent uses LLM-based inference, sentiment analysis, and CATME-aligned text classification to identify emerging breakdowns in group communication. Chat messages are continuously logged to a structured PostgreSQL environment and analyzed to identify team stress indicators across the five CATME teamwork dimensions. When any dimension falls below threshold, the agent summarizes relevant conversation segments, flags signals of dysfunction, and notifies the course instructor with actionable insight [29].

By analyzing student collaboration data at conversational speed rather than at survey intervals, this approach shifts team support from reactive to proactive. Instead of relying solely on self-reporting or late-stage instructor intervention, AI-enabled monitoring provides just-in-time awareness of communication climate, contribution equity, and emotional tone. This research contributes a new model for feedback triangulation that blends continuous chat-based analytics, LLM summarization, and validated teamwork frameworks to enhance student success in high-collaboration learning environments [30].

Methods

The study followed a two-stage deployment process. In the initial pilot phase, the system was tested with two student teams comprising four students total to verify logging stability, response generation, and workflow reliability within the Discord environment. After confirming system functionality, deployment expanded to the full course, involving five teams and a total of 15 students. All teams used the Discord platform for project communication and had access to the AI agent during the first half of the semester. Survey responses reported in the Results section reflect participation from this full deployment group, although the number of valid responses varies by survey item.

System Architecture

We implemented a multi-conversational Discord AI agent that generates responses using a large language model (LLM) and maintains persistent conversational context across channels and threads. The system is event-driven and is activated only when explicitly mentioned by a user, ensuring controlled and predictable interactions.

The novelty of this system lies in its integration of persistent conversational memory, CATME-aligned teamwork analysis, and instructor-facing alerts within a single workflow. While individual components such as sentiment analysis or LLM-based response generation exist independently, their combination enables longitudinal monitoring of team dynamics and earlier detection of collaboration challenges than traditional survey-based approaches.

The agent is composed of three core components: (i) a Discord message handler responsible for ingesting and logging messages, (ii) a memory subsystem that stores conversation history and rolling summaries in a PostgreSQL database, and (iii) a response generation module that assembles contextual information and queries the LLM. Persistent storage enables the agent to maintain continuity across long-running conversations without exceeding prompt length constraints.

Agent Workflow

The agent follows a structured workflow consisting of three main stages:

1. Message Ingestion and Triggering
   Incoming Discord messages are captured and stored along with metadata such as channel, thread, and reply relationships. The agent remains inactive unless explicitly mentioned, at which point the message is marked for response generation.

2. Context Construction and Memory Augmentation
   Upon activation, the agent constructs a contextual prompt by combining recent messages, thread- or channel-level history, and previously stored summaries. When conversations exceed a predefined size threshold, older content is summarized and persisted, allowing the agent to preserve semantic continuity while controlling context length.

3. Response Generation and Delivery
   The assembled prompt is sent to the LLM for inference. The generated response is

post-processed to comply with Discord message limits and returned to the originating channel or thread.

Implementation Details

The system is implemented in Python and uses Google Gemini (gemini-2.5-flash) for language generation. PostgreSQL is used for persistent storage, with tables automatically initialized at runtime. This design enables scalable multi-threaded interaction while maintaining coherence across independent conversations.

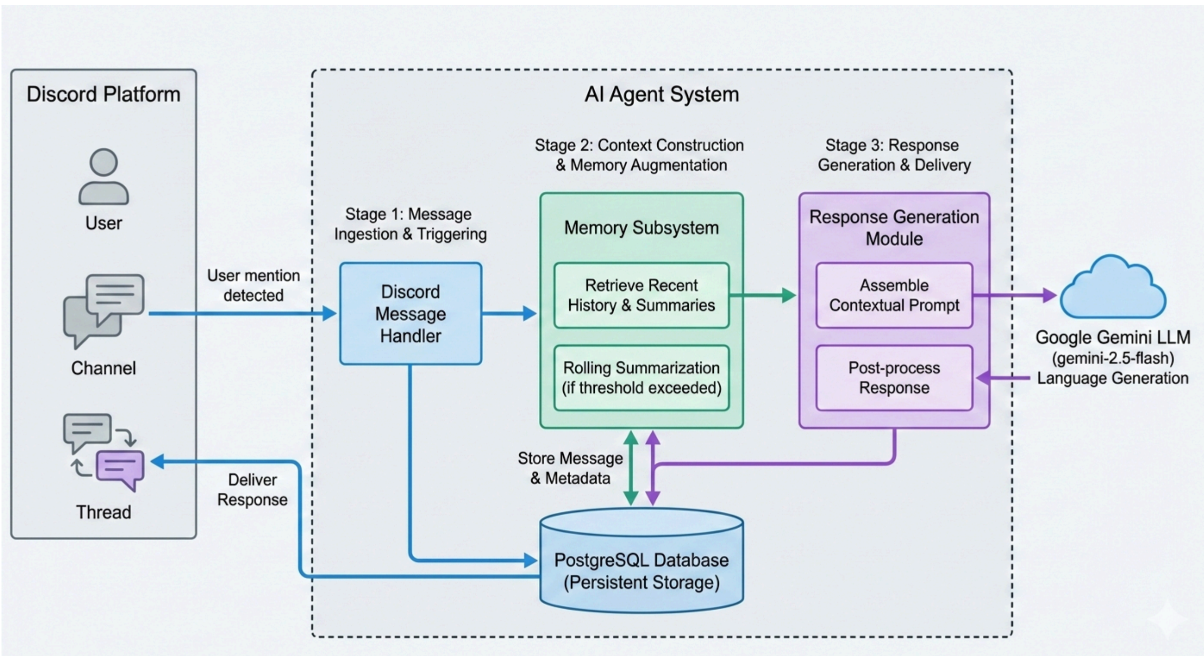


Figure 1 : System architecture and workflow of the Agent

Results

Students reported using the AI assistant primarily during early project planning and coordination tasks. Among respondents who completed usage questions (**N = 10**), **6 students (60%)** indicated that the assistant helped them organize project ideas and structure tasks, while **5 students (50%)** reported using it to clarify technical concepts. Fewer students relied on the assistant for programming support (**3 students, 30%**) or debugging (**2 students, 20%**), suggesting that students perceived the tool primarily as a planning and reasoning aid rather than a replacement for hands-on technical work.

Students also reported generally positive experiences with the assistant. Of the respondents who rated overall satisfaction (**N = 10**), **2 students (20%)** rated the assistant at the highest level (5), **6**

students (60%) rated it as 4, and **2 students (20%)** rated it as 3. No respondents rated the assistant negatively.

Qualitative responses further indicated that students valued the assistant's ability to summarize discussions, suggest next steps, and recall previous project context. Several respondents noted that the assistant was particularly useful for brainstorming and structuring project workflows, reinforcing its role as a collaboration support tool rather than a direct problem-solving replacement.

Here are the survey results from students who have been interacting with the AI agent. The results show that students most often used AI tools to support the early stages of their projects. Around 60% reported relying on AI for project planning, breaking tasks into manageable steps, and brainstorming ideas. About half of the students used AI to help understand technical concepts. In comparison, fewer students turned to AI for direct programming assistance, with roughly 30% using it for coding help and only 20% for debugging. Overall, these patterns suggest that students viewed AI primarily as a tool for organizing ideas and shaping their thinking, rather than as a replacement for hands-on coding or troubleshooting.

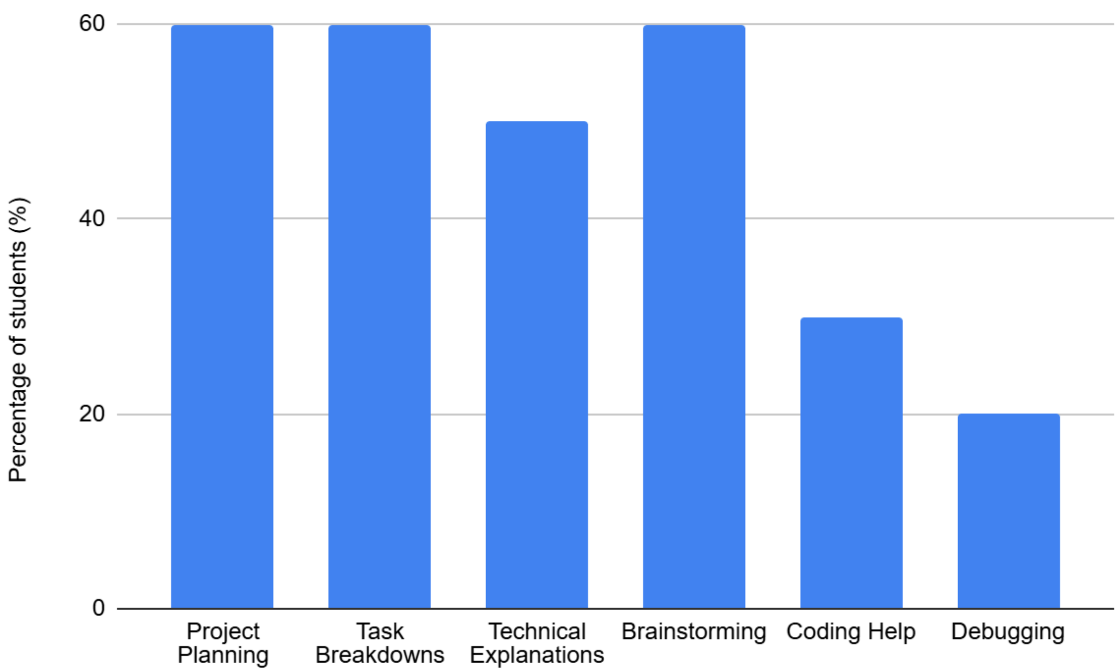
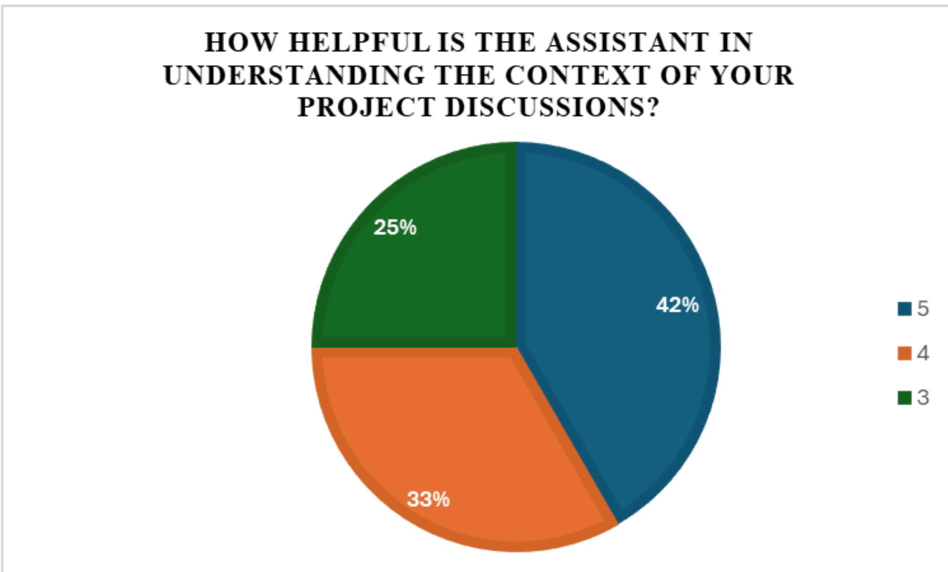


Figure 2: Survey results showing how students were using the agentic AI

Figure 3 shows the students' user experience. The largest portion of respondents (42%) rated the assistant at the highest level of helpfulness (5), indicating strong perceived support. Another 33% of students selected a rating of 4, suggesting that the assistant was generally helpful. The remaining 25% rated the assistant at level 3, reflecting moderate usefulness. Overall, the distribution shows that a majority of students viewed the assistant as helpful or very helpful in supporting their understanding of project-related conversations.

(a) User experience on project context discussion (1-5 Scale, 5 means very satisfactory)



(b) User experience on project information recall

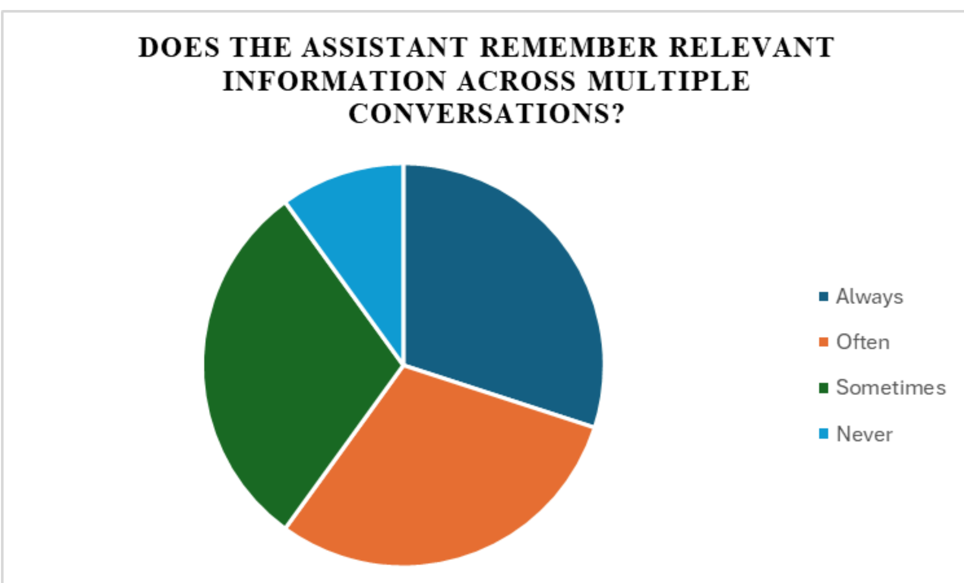


Figure 3. **User experience survey after the students trying out the AI agent**

Figure 4 shows the dialog between the students and the AI agent. The interaction starts with a student asking what kinds of technical questions the assistant can help with. In its response, the assistant explains that it can support a range of topics, including research methods, project planning, development best practices, and specialized engineering areas such as soft robotics. The student then asks for guidance on how to evaluate and compare different mechanical design iterations in a robotics project. The Project_Assistant replies with a clear, step-by-step explanation, beginning with the importance of establishing measurable performance criteria, such as payload capacity, strength-to-weight ratio, energy efficiency, accuracy, manufacturability, and cost. The screenshot highlights the natural back-and-forth of the discussion, including timestamps, user mentions, and the assistant's detailed technical guidance within a collaborative team communication setting.

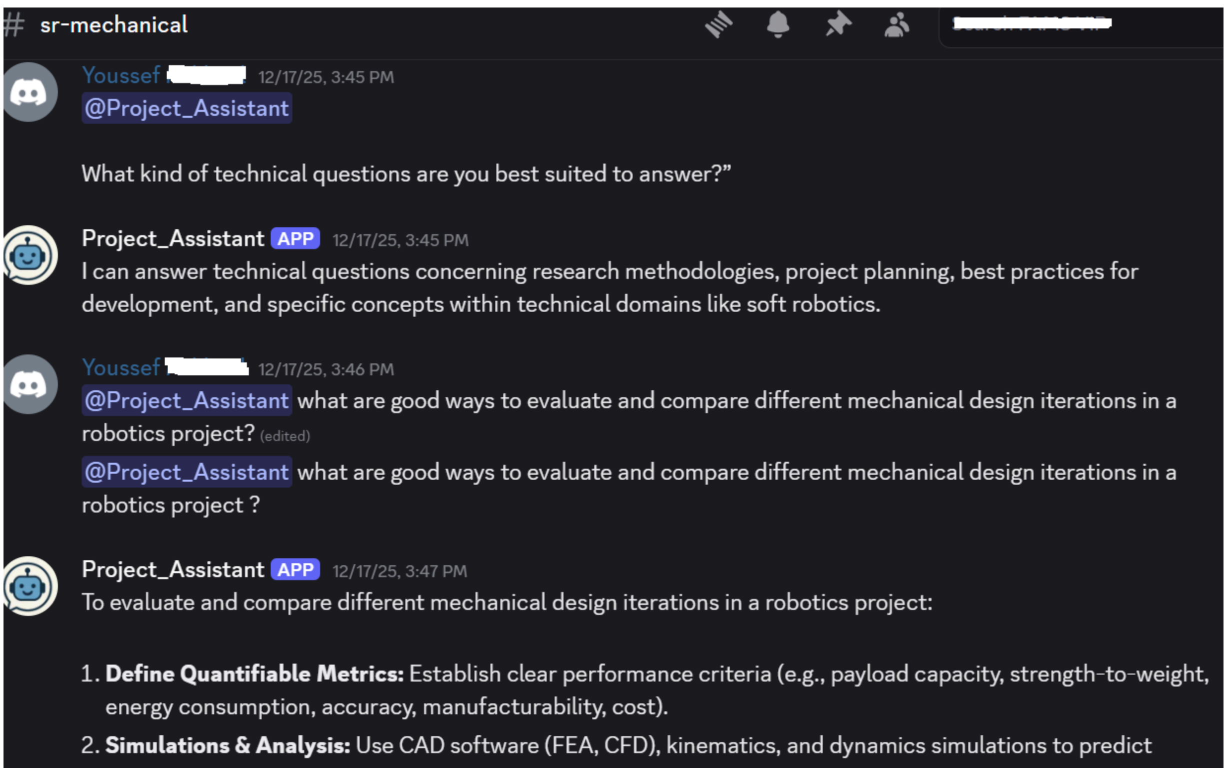


Figure 4: The AI interface

Discussion

The results of this study suggest that an AI-supported, conversation-aware agent can meaningfully enhance visibility into student team dynamics within project-based engineering courses. Rather than replacing existing teamwork assessment tools, the proposed system complements them by providing continuous, real-time insight into how teams interact as projects unfold. This addresses a persistent limitation of traditional teamwork assessment approaches, which often rely on retrospective surveys administered only once or twice during a semester.

One of the most significant contributions of this work is the integration of validated CATME teamwork dimensions into an automated NLP pipeline. By aligning sentiment analysis and text classification with established constructs such as contribution, interaction quality, and keeping the team on track—the system translates unstructured chat data into interpretable indicators that instructors already recognize. This alignment is critical for instructional adoption, as it bridges emerging AI analytics with familiar pedagogical frameworks rather than introducing opaque or unfamiliar metrics.

The Discord-based deployment further reflects the realities of modern student collaboration. As much of students' teamwork now occurs outside formal learning management systems, instructors often lack awareness of emerging problems until deadlines are missed or peer evaluations reveal conflict. By operating within the same platforms students naturally use, the AI agent reduces this visibility gap and enables earlier identification of issues such as disengagement, frustration, or uneven participation.

Another important finding is the role of contextual memory in supporting long-term analysis. Many prior educational chatbots and analytics tools operate on short conversational windows, limiting their ability to recognize persistent patterns of behavior. The use of a persistent database and rolling summarization allows the agent to maintain continuity across weeks of interaction. This design enables the system to distinguish isolated disagreements from recurring dysfunction, an essential distinction for meaningful instructor intervention.

From an instructional perspective, the instructor-facing notification mechanism represents a shift from reactive to proactive support. Rather than requiring instructors to manually inspect conversations or rely solely on student self-reporting, the system synthesizes key signals and provides concise summaries tied to specific teamwork dimensions. This approach respects instructor time constraints while preserving human decision-making authority; the AI surfaces concerns, but instructors remain responsible for interpretation and response.

Despite these promising outcomes, several limitations should be acknowledged. The current implementation focuses primarily on text-based communication and does not capture nonverbal collaboration cues such as meeting attendance, shared document edits, or task management behavior. Sentiment analysis and NLP classification may also misinterpret sarcasm, humor, or culturally specific language. Additionally, the system has not yet been evaluated at large scale across multiple courses or institutions, limiting the generalizability of findings. A prototype is also developed with advanced listening and speech capabilities (multimodal perception) so that also an embodied version of this AI agent will have a physical footprint and interactive medium with student groups being an active member and giving out valuable feedback as a team member.

Ethical considerations are equally important. Continuous monitoring of student communication raises valid concerns regarding privacy, transparency, and consent. While the system is designed to operate within course-sanctioned channels and emphasizes instructor awareness rather than surveillance, careful policy design and student communication remain essential. Future implementations must continue to prioritize ethical AI use, clear disclosure, and opt-in participation.

Overall, the findings indicate that AI-supported analysis of team communication can serve as a useful instructional aid when grounded in an educational framework.

Conclusion

This project presented the design and development of a multi-conversational AI agent intended to support student teamwork in project-based engineering education. By integrating natural language processing, sentiment analysis, persistent conversational memory, and CATME-aligned classification, the system offers a structured approach for monitoring collaboration challenges as they emerge within student team interactions.

The proposed framework addresses a longstanding challenge in engineering education: the lack of timely visibility into how student teams function outside the classroom. Rather than relying exclusively on retrospective peer evaluations or manual instructor intervention, the system enables continuous monitoring into contribution patterns and team dynamics.

Importantly, this work positions AI not as an automated evaluator of student performance, but as an interactive facilitator that augments instructor awareness and supports earlier, more targeted intervention. By grounding analytics in validated teamwork dimensions, the system maintains pedagogical coherence while leveraging the scalability of modern AI tools.

Future work will focus on expanding deployment in a bigger class with more samples, and conducting controlled studies to evaluate learning outcomes associated with earlier intervention. As engineering education continues to emphasize teamwork as a core professional competency, tools that support human instruction will become increasingly essential. This study demonstrates that thoughtfully designed AI agents can play a meaningful role in fostering healthier, more effective student collaboration while preserving instructor judgment, student agency, and educational values.

References

[1] A. Gupta, E. Schaefer, and J. Cox, “Teamwork as a core engineering competency: A review of instructional strategies,” in *Proc. ASEE Annu. Conf.*, 2023.

[2] M. Borrego and S. Cutler, “Curriculum adoption of project-based learning across engineering programs,” *Eng. Educ.*, vol. 115, no. 2, pp. 211–225, 2024.

[3] C. M. Samuel and A. L. Lee, “Scaling vertically integrated projects to support undergraduate learning,” in *Proc. ASEE Annu. Conf.*, 2024.

[4] A. Martin and R. Frey, “Sources of conflict in undergraduate engineering design teams,” *J. Eng. Educ.*, vol. 113, no. 1, pp. 45–62, 2024.

[5] H. Choi, L. Chen, and S. Ayazifar, “Uneven contribution patterns and student perception of fairness in teamwork,” in *Proc. ASEE Annu. Conf.*, 2022.

[6] B. Sharpe and D. Tran, “Communication breakdowns in distributed project teams,” *IEEE Trans. Educ.*, vol. 67, no. 4, pp. 515–526, Oct. 2024.

[7] D. Berkowitz and R. Shivakumar, “Invisible misalignment: The unseen failures of team collaboration,” in *Proc. ASEE Annu. Conf.*, 2024.

[8] H. Li and C. Han, “Digital-first teamwork: The shift from classroom teams to collaborative platforms,” *Comput. Educ.*, vol. 207, 2024, Art. 104826.

[9] M. Ohland, R. Layton, H. Loughry, and P. Woehr, “Developing a peer evaluation system for teamwork: The CATME Model,” *J. Eng. Educ.*, vol. 99, no. 3, pp. 339–353, 2019. (Foundational validation, often cited)

[10] R. Layton and M. Ohland, “Using CATME to improve engineering team effectiveness,” in *Proc. ASEE Annu. Conf.*, 2022.

[11] H. Loughry, M. Clayton, and D. Pawley, “Peer feedback insights from CATME deployment in large courses,” *IEEE Trans. Educ.*, vol. 66, no. 2, pp. 168–179, Apr. 2023.

[12] K. AlHabeeb and A. Brown, “Timing matters: Retrospective evaluations vs. real-time team monitoring,” in *Proc. ASEE Annu. Conf.*, 2023.

[13] Y. Sun and B. Zhang, “LLM-driven analysis of collaborative learning,” *IEEE Access*, vol. 12, pp. 112340–112356, 2024.

[14] P. Srinivas and A. Aggarwal, “Automating role identification in student teams using NLP,” in *Proc. ASEE Annu. Conf.*, 2023.

[15] A. Joshi and E. Romero, “Natural language processing to summarize student discussions,” *Comput. Educ.*, vol. 203, 2023, Art. 104919.

[16] X. Hu and J. Li, “Predicting team success using sentiment-aware models,” *IEEE Trans. Learn. Technol.*, vol. 17, no. 3, pp. 450–462, Jul. 2024.

[17] F. Milton and S. Barber, “Reflection prompts supported by automated feedback,” in *Proc. ASEE Annu. Conf.*, 2024.

[18] E. J. Kim and P. Sullivan, “Conversational agents for coaching undergraduate design teams,” *ACM Trans. Comput.-Hum. Interact.*, vol. 32, no. 1, pp. 1–29, 2025.

[19] D. Harris and A. Rogers, “Scaling formative assessment using LLM-based feedback,” in *Proc. ASEE Annu. Conf.*, 2024.

[20] Z. Feng and R. Iqbal, “Deploying AI teaching assistants in engineering design courses,” *IEEE Trans. Educ.*, vol. 67, no. 1, pp. 85–96, Jan. 2024.

[21] H. Zhu and P. Liang, “Automated summarization of team collaboration logs,” *Int. J. Eng. Educ.*, vol. 40, no. 2, pp. 189–205, 2024.

[22] J. Park and F. Pan, “Detecting isolation and dropout in virtual student teams,” in *Proc. ASEE Annu. Conf.*, 2023.

[23] M. Rangel and S. Gupta, “Topic-based modeling to assess workload equity,” *IEEE Trans. Learn. Technol.*, vol. 18, no. 1, pp. 20–35, Jan. 2025.

[24] L. Stevens and K. Wu, “Sentiment trends as predictors of team conflict,” in *Proc. ASEE Annu. Conf.*, 2022.

[25] D. Morgan and A. Kim, “Limitations of standalone dashboards for collaboration management,” *Comput. Educ.*, vol. 210, 2024, Art. 104996.

[26] T. Barnes and A. Rich, “Retrospective analytics in teamwork: Too little, too late,” *IEEE Access*, vol. 12, pp. 100234–100249, 2024.

[27] B. Chen and N. Davis, “Integrating CATME constructs within automated analytics pipelines,” in *Proc. ASEE Annu. Conf.*, 2024.

[28] S. Rahman and K. Patel, “Automated alerts for instructional triage,” *Front. Educ.*, vol. 9, Art. 1443225, 2024.

[29] Y. Lin, C. Ng, and W. Lee, “A Discord-based agent for monitoring teamwork signals,” in *Proc. ASEE Annu. Conf.*, 2025.

[30] E. Robinson, S. Kumar, and T. Jensen, “Proactive instructor-facing AI support in project courses,” *IEEE Access*, vol. 12, pp. 121923–121945, 2024.