Bridging Instead of Replacing Online Coding Communities with AI through Community-Enriched Chatbot Designs

JUNLING WANG, ETH Zurich, Switzerland
LAHARI GOSWAMI, ETH Zurich, Switzerland
GUSTAVO KREIA UMBELINO, ETH Zurich, Switzerland
KIARA CHAU, ETH Zurich, Switzerland
MRINMAYA SACHAN, ETH Zurich, Switzerland
APRIL YI WANG, ETH Zurich, Switzerland

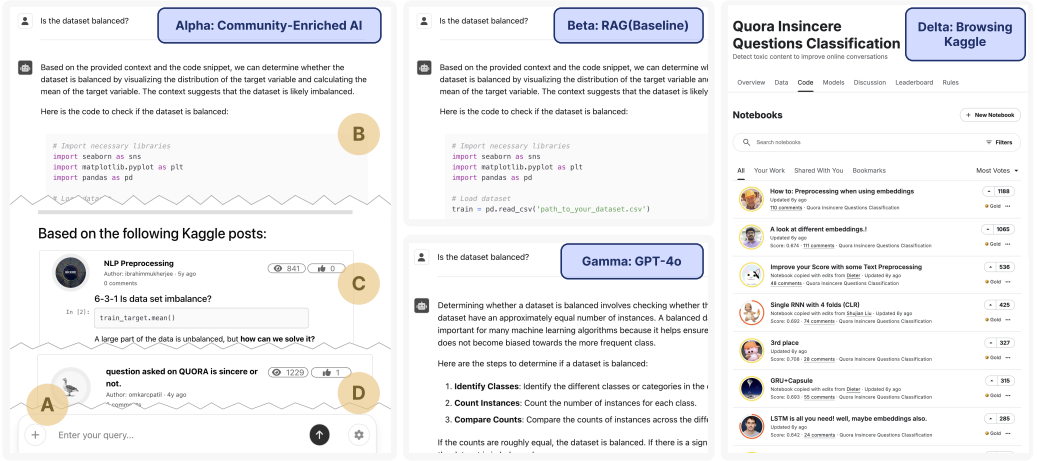


Fig. 1. An overview of ChatCommunity, a Community-Enriched AI chatbot designed to integrate user-generated content and social design features from online coding communities into AI-assisted interactions. The interface consists of four key components: (A) a query input bar with session controls, (B) a chat area where AI-generated responses appear, (C) a source document panel previewing the Kaggle posts referenced in responses, complete with social design features such as author identity and engagement metrics, and (D) an advanced search panel for customizing search parameters. We also illustrate the four conditions used in our evaluation: Alpha (Community-Enriched AI), which embeds social learning dynamics into the chatbot interface; Beta (RAG-baseline), which uses the same RAG-based generation model but does not visualize community content; Gamma (GPT-4o), a state-of-the-art LLM baseline without community grounding; and Delta (Browsing Kaggle posts), where users manually explore Kaggle without AI assistance.

Authors' Contact Information: Junling Wang, junling.wang@ai.ethz.ch, ETH Zurich, Zurich, Switzerland; Lahari Goswami, lahari.goswami@inf.ethz.ch, ETH Zurich, Zurich, Switzerland; Gustavo Kreia Umbelino, gus.umbelino@inf.ethz.ch, ETH Zurich, Zurich, Switzerland; Kiara Chau, kchaugarcia@student.ethz.ch, ETH Zurich, Zurich, Switzerland; Mrinmaya Sachan, mrinmaya.sachan@inf.ethz.ch, ETH Zurich, Zurich, Switzerland; April Yi Wang, april.wang@inf.ethz.ch, ETH Zurich, Zurich, Switzerland.

ACM 2573-0142/2026/4-ARTCSCW008
https://doi.org/10.1145/3788044

LLM-based chatbots like ChatGPT have become popular tools for assisting with coding tasks. However, they often produce isolated responses and lack mechanisms for social learning or contextual grounding. In contrast, online coding communities like Kaggle offer socially mediated learning environments that foster critical thinking, engagement, and a sense of belonging. Yet, growing reliance on LLMs risks diminishing participation in these communities and weakening their collaborative value. To address this, we propose Community-Enriched AI, a design paradigm that embeds social learning dynamics into LLM-based chatbots by surfacing user-generated content and social design features from online coding communities. Using this paradigm, we implemented a RAG-based AI chatbot leveraging resources from Kaggle to validate our design. Across two empirical studies involving 28 and 12 data science learners, respectively, we found that Community-Enriched AI significantly enhances user trust, encourages engagement with community, and effectively supports learners in solving data science tasks. We conclude by discussing design implications for AI assistance systems that bridge—rather than replace—online coding communities.

CCS Concepts: • **Human-centered computing** → **Empirical studies in HCI**; **Empirical studies in collaborative and social computing**.

Additional Key Words and Phrases: Online coding community, Reader-to-leader framework, Social transparency, Perceived reliability, RAG-based agent, Data science assistants, Large Language Models

ACM Reference Format:
Junling Wang, Lahari Goswami, Gustavo Kreia Umbelino, Kiara Chau, Mrinmaya Sachan, and April Yi Wang. 2026. Bridging Instead of Replacing Online Coding Communities with AI through Community-Enriched Chatbot Designs. *Proc. ACM Hum.-Comput. Interact.* 10, 2, Article CSCW008 (April 2026), 37 pages. https://doi.org/10.1145/3788044

1 Introduction

Online coding communities like Kaggle and Stack Overflow are thriving ecosystems where millions of programmers and data scientists engage socially to exchange knowledge, seek solutions, and collaboratively build expertise [18, 39]. By 2024, Kaggle alone hosts nearly 19 million active users [73], serving not merely as repositories of knowledge but as social spaces that facilitate knowledge exchanges and co-creation through meaningful social interactions. Social interactions within these communities offer several distinct benefits for data science learners: they expose learners to diverse problem-solving approaches [18, 89], surfacing active discussions and debates that promote critical thinking and reflection [67, 88, 92], and enabling personalized guidance and encouragement from more experienced peers [23, 93]. Additionally, social interactions cultivate a sense of belonging, encouraging sustained engagement and motivating ongoing participation and contribution [24, 89].

In recent years, Large Language Models (LLMs) have emerged as powerful assistive tools in various tasks, including answering programming questions [14], helping with debugging [100], and often serving as convenient alternatives to traditional search-and-explore methods in online communities [43]. Despite effectiveness of LLMs, social interactions such as knowledge exchange and engagement with others' content within coding communities remain highly meaningful [18, 90]. LLMs, while capable of rapidly generating responses, often provide impersonal, isolated, and static answers that lack context and nuanced understanding [50]. In contrast, coding communities allow users to engage with past conversations, clarify their problems through follow-up questions, and receive tailored responses from peers who share similar challenges [18, 80, 90]. Additionally, coding communities support critical thinking and engagement in ways LLMs currently do not [43]. Online coding communities like Stack Overflow presents multiple answers and public commentary for each query, enabling comparison, clarification, and evaluation of competing perspectives [18, 43, 80, 90]. This format cultivates argumentation literacy by encouraging users to reason through alternatives [35, 80]. In contrast, LLM-based tools such as ChatGPT typically provide a single confident response, which may reduce opportunities for deliberation and reinforce passive consumption and

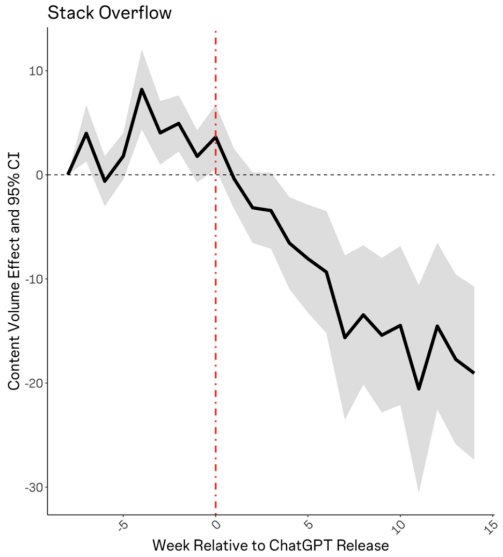


Fig. 2. Effects of ChatGPT over time (by week) on Stack Overflow question volumes per topic, with ChatGPT released on November 30th, 2022. Adapted from [10].

delegation among learners [104]. Moreover, while LLMs are helpful, they often suffer from hallucinations — generating plausible yet factually incorrect responses without proper citations [43, 77, 86]. These inaccuracies can significantly undermine user trust, especially in programming contexts where precision and reliability are critical [97].

On the other hand, the increasing reliance on LLM-based tools like ChatGPT endangers active participation to online coding communities [10]. As illustrated in Figure 2, the number of questions posted per topic on Stack Overflow has significantly decreased since the release of ChatGPT. As more people turn to LLM-based tools for quick answers instead of engaging in communities, not only are fewer contributions made, but the incentives to contribute are also diminished [10]. Recent research within the Computer Supported Cooperative Work (CSCW) and Human-Computer Interaction (HCI) community further emphasizes this trend, highlighting how dependency on LLMs can diminish essential social interactions among students, negatively impacting peer relationships and collaborative learning opportunities [32, 69].

Given the widespread adoption and practical benefits of LLM through chatbots, it is not feasible to abandon them; instead, there is an opportunity to redesign them in ways that better support community engagement. Therefore the question we pose is : ***How can we design LLM-based chatbots to augment, rather than displace community participation in online coding communities?*** In particular, we focus on designing a chatbot that redirects users' attention back toward the community rather than keeping it within the chatbot itself. In this paper, we introduce a novel design paradigm called *Community-Enriched AI*, which embeds social learning dynamics into AI responses by surfacing user-generated content and social design features from online coding communities like Kaggle. At the core of this paradigm is a Retrieval-Augmented Generation (RAG) mechanism that grounds AI responses in relevant community notebook posts. This grounding not only reduces response hallucinations [53, 82], but also ensures that responses reflect diverse, real-world problem-solving approaches from the community. Building on top of this, our design displays previews of source posts along with rich social cues such as author profiles, vote counts, view counts, and more. These features are deliberately chosen to increase social transparency and social

presence in the interaction. Drawing from social transparency theory [25, 85], we aim to make the social context of knowledge creation visible, allowing users to see who contributed the information and how the community engaged with it. This visibility supports trust calibration, informed decision making, and social interaction [25, 27, 85]. Similarly, social presence theory [30, 64] informs our inclusion of identity cues and interaction signals that help users feel connected to real community members, rather than interacting with an isolated AI. By embedding these theories into the interface design, we aim to not only encourage users to critically assess AI responses, but also support engagement with the online community. In particular, the design promotes "lurking" behavior — the passive consumption of community content — which research has shown to be a valuable early step toward active participation in online communities [51, 61, 74]. At the same time, it reduces the friction traditionally associated with browsing and searching Kaggle threads.

To evaluate our design, we instantiate it in ChatCommunity, a chatbot that operationalizes the concept of Community-Enriched AI for the Kaggle forum. We conducted two user studies to evaluate the effectiveness and explore the design space of Community-Enriched AI. In the first study, we compared ChatCommunity with three baseline approaches — RAG-baseline, GPT-4o, and traditional browsing — using a within-subjects lab study with 28 participants. The results indicate that learners leveraged the community-enriched previews to navigate toward Kaggle community posts, providing direct evidence of ChatCommunity's effectiveness in encouraging engagement and participation in online coding communities. This design significantly improves the perceived reliability of the content compared to presenting the same generated content without community context.

Building on these findings, we conducted a second design exploration study with 12 participants to examine how varying levels of community integration — ranging from minimal links to full summaries with social design features — influence users' trust and community engagement. Participants responded most positively to designs that paired AI's answers with rich visual previews of relevant posts and visible social signals, such as author profiles and vote counts. Designs that aggregated and summarized the community perspectives were deemed especially effective in enhancing trust and helping users feel more confident in their understanding. By making community knowledge and interactions visible through these features, users were able to engage with peer insights and explore content more efficiently — thereby fostering self-directed learning. Together, the two studies highlight the importance of integrating community knowledge in AI systems to facilitate socially-informed learning.

In summary, our work makes the following contributions:

- *Community-Enriched AI*, a design paradigm that enhances AI chatbots with user-contributed content from online coding communities, explicitly displaying previews of sources alongside rich social design features.
- ChatCommunity, a chatbot integrates Community-Enriched AI design paradigm to assist data science learners on the Kaggle platform.
- An empirical evaluation of ChatCommunity confirms that the Community-Enriched AI design can enable the first step toward improving community participation, enhance users' trust in model responses, and effectively support learners with data science tasks.
- Provides design insights on integrating varying levels of social features into Community-Enriched AI systems to facilitate users' trust in AI and toward meaningful engagement with the community.

2 Related Work

2.1 Community Engagement in Online Coding Communities

Learning programming and software development is often a community-driven endeavor where online coding communities and forums play a central role in facilitating knowledge sharing, learning through collective discussions, deliberation and sharing of feedback [3, 18, 79]. Online coding communities, including question-and-answer forums like Stack Overflow and domain-specific platforms with open innovation competitions, like Kaggle, are widely used by both novices and experienced developers for problem-solving, sharing code and solutions, and engaging in informal learning through unstructured activities and social interactions [18, 80, 90]. Learners participate in online coding communities not only to improve their programming and data science skills but also to engage in socially grounded practices like sharing solutions, commenting on alternative approaches, and collaboratively building community knowledge [18, 89, 99].

Participation in online communities has been explored through different theoretical lenses [48, 75]. For example, the reader-to-leader framework describe participants as *readers* who browse and search for content, *contributors* who post and rate content, *collaborators* who work together to improve the community, and *leaders* who promote participation and mentor novices [75]. A common characteristic across these frameworks is that most contributions are made by a minority of participants. Prior work suggests that over 90% of participants are not active contributors [13], indicating that the majority of users in online coding communities are readers. Understanding how to design features that enhance readers' social presence is crucial, as this can foster greater community engagement and potentially encourage readers to transition into active roles such as contributors. To achieve this, it is essential to consider how online communities promote their presence and encourage participation of potential members [45].

While participating in online coding communities are supported in multiple ways, there are still barriers to participation that might limit learning. Specifically, as communities grow, it becomes increasingly difficult for learners to screen relevant content [34, 84, 91]. Another limitation is that when learners post a question in an online coding community, the response time can vary, and some questions may not receive a response at all [5, 29].

To address these barriers, CSCW and HCI researchers have explored ways to pair novices with experienced community mentors, helping novices adhere to community cultural norms when asking questions, which may lead to quicker responses [33]. Other researchers have collected and processed Kaggle notebooks as datasets to facilitate research on code metrics in Jupyter notebooks [60]. However, these methods do not utilize community resources to provide timely and contextually relevant responses to learners' questions.

2.2 Automatic Help Seeking in Code Learning

Before the LLMs era, search engines and information retrieval tools were crucial for programming tasks, providing access to tutorials, documentation, and community discussions [8, 38]. In the HCI community, Hoffmann et al. [38] introduced Assieme, a search interface that consolidated distributed programming resources such as API documentation, sample code, and explanations. By resolving implicit references, Assieme enabled programmers to find better solutions with fewer queries compared to general-purpose search engines. Brandt et al. [8] further examined programmers' use of online resources, identifying opportunistic learning behaviors such as just-in-time learning, clarifying existing knowledge, and memory aids. These studies highlighted the limitations of traditional search tools in addressing programming learners' contextual needs — such as integrating fragmented resources and supporting task-specific information seeking [8, 38].

With the advancement of LLM, AI-assisted tools have shown great potential in providing on-demand help to programming learners. Recent work in CSCW and HCI has explored the design and deployment of LLM-based tools tailored to programming education, which can be broadly categorized into two directions: tools that serve as instructional scaffolds, and those that function as problem-solving aids. Firstly, LLMs have been used as instructional scaffolds to support conceptual understanding and cognitive development. For example, Kazemitabaar et al.[44] introduced a programming assistant that responds to conceptual questions and generates pseudocode with explanations to help learners reason about program structure. Extending this direction, Ma et al.[55] proposed DBox, an interactive system that scaffolds algorithmic problem-solving through learner-LLM co-decomposition. By guiding students to incrementally construct a step tree and aligning their logic with real-time LLM feedback, DBox fosters critical thinking, cognitive engagement, and independent problem-solving. Expanding this focus to collaborative contexts, Yan et al. [102] examined the role of LLMs in collaborative programming among middle school students. Their findings suggest that LLM-enhanced collaboration significantly improves computational thinking and reduces cognitive load.

LLMs have also been deployed as problem-solving aids to assist learners with concrete coding tasks. Yang et al.[103] examined help-seeking behaviors while learners interacted with a pedagogically designed chatbot to debug code, identifying strategies students use to engage with AI support. Complementing this perspective, Prather et al.[72] investigated novices' experiences with LLMs and found that learners with foundational knowledge benefited from LLM-supported coding. Moving beyond reactive help, Chen et al. [16] explored proactive LLM-based assistants that offer context-aware programming support without explicit user prompts. Their study showed that proactive suggestions integrated into the coding environment improved user productivity and experience, while also highlighting design trade-offs in mixed-initiative human-AI collaboration.

2.3 Challenges LLMs Present to Learners and Existing Solutions

Despite their benefits, LLM-based tools pose certain challenges for learners. Firstly, LLMs are prone to "hallucinations" [77] which can propagate low quality or incorrect knowledge to the generated content [43]. As identified by Park et al. [69] *hallucinations*, among other algorithmic problems (e.g., algorithmic bias, lack of transparency and interpretability), are a critical limitation of LLMs which impedes productive learning. As a result, learners need a mechanism to validate whether AI-generated responses are trustworthy. While HCI researchers have proposed the design and adoption of specialized LLMs for education and the incorporation of fact-checking algorithms [69], another approach is to provide source information relevant to the query for the generation system, a concept central to RAG [53]. The RAG system integrates a retriever and a generator to enhance question-answering. The retriever retrieves relevant information from a data collection, which is then used as context by the generator to produce answers with reduced hallucination [53, 82]. However, researchers have not yet explored whether or how providing sources for learners is beneficial. A recent HCI study show that incorporating community-curated experiences helps build trust in AI code generation tools [17]. Although their tools were only tested on a small, simulated community and focused solely on code generation, their findings inspired us to build ChatCommunity, which provides users with real community-curated insights for data science tasks.

Secondly, reliance on LLMs for accessing information on coding and software development can also reduce user participation in online coding communities, thereby hindering the dynamic exchange of knowledge and limiting opportunities for learning from these communities. Recent research has found a decline in daily web traffic on Stack Overflow by approximately 1 million individuals per day following ChatGPT's release [10]. It has also been observed, question posting

volumes per topic on Stack Overflow have markedly declined since ChatGPT's introduction — showing the disconnect of learners from online coding communities. Figure 2 from their paper illustrates the changes in question posting volumes per topic on Stack Overflow. In educational contexts, studies have shown that the use of LLMs can reduce social interactions among learners creating isolated learning experiences, thus emphasizing the importance of designing systems that encourage peer engagement alongside AI assistance [69]. Moreover, an over-reliance on LLMs for quick, convenient answers can undermine learners' opportunities for in-depth learning and the development of critical thinking skills. In programming, where the fostering of these skills is most acute, the effects of using GenAI have been shown to compound the difficulties in learning coding, especially for struggling learners [69, 72]. Reduced participation in online coding communities further diminishes learners' opportunities for social learning, narrowing the scope for developing deeper knowledge and problem-solving abilities through engagement with the communities. RAG models have the potential to provide sources to connect the learners back to online community. Recent efforts such as Social-RAG [98] have begun exploring how retrieval from prior group interactions can socially ground AI generated messages. However, HCI researchers have not yet explored the effects of such approach on engaging participation in online coding communities.

This paper bridges online coding communities and LLMs by integrating community-driven features to overcome limitations of current LLM tools. To do so, we explore designing a Community-Enriched AI system — ChatCommunity — that leverages RAG and social design features to reconnect users with peer-contributed content and foster community engagement for learning. We contribute to the understanding of socially enriched AI-assisted programming by evaluating the benefits of a Community-Enriched AI design compared to traditional information retrieval methods or general chatbots lacking such integration, providing insights relevant to broader educational ecosystems.

3 Community-Enriched AI: Design Paradigm and Instantiation

We introduce Community-Enriched AI, a design paradigm that integrates the social context of community-generated knowledge into LLM-based chatbots. To demonstrate this idea, we instantiate it in ChatCommunity, a RAG-based chatbot that grounds AI-generated responses to the posts of the Kaggle community.

3.1 Design Motivations for Community-Enriched AI

3.1.1 Provide Opportunities to Engage with Other Learners. Data science learners use online spaces to network, learn, and share resources and passions for topics — interacting continuously and fostering communities of practice [81, 101]. As an online platform for the data science community, Kaggle provides a space for learners to practice data science skills and engage with others through well-established competitions, community knowledge-building activities such as public code sharing, and social Q&A-based discussions [18]. The design of our Community-Enriched AI aims to create a congruence between the AI user and the social dynamics of the Kaggle community, enabling learners to complement their individual learning through social engagement.

Our design is informed by principles from social transparency theory [85] and social presence theory [30], as applied to AI system design [25]. Social transparency highlights the value of making the social context of knowledge creation visible to enhance collaboration and cooperative behaviors [25, 85]. Social presence, on the other hand, highlights the importance of fostering a sense of connection and human interaction within a system, where cues of social presence can increase community participation [30, 64]. Drawing from these two perspective, our design intends to display peer-generated content with social design features that can encourage users, like passive

readers [75] to engage with the community and also support the content creators by boosting the visibility of their profiles.

3.1.2 Toward Meaningful Engagement through Community-Guided Information Foraging. Traditional help-seeking approaches, such as using Google Search, can be time-consuming and overwhelming for learners to find relevant resources that meet their specific needs [36, 105], especially when tackling new or domain-specific competitions. An effective system should streamline this process by not only helping users find relevant content but also supporting meaningful engagement with that content to promote deeper exploration and critical thinking [56]. By leveraging a retrieval-based approach grounded in community curated knowledge space, our design aims to optimally enable this process for learners by providing them with contextually appropriate resources, and also building a segue for them to engage with community knowledge for thoughtful exploration and learning on the topic.

3.1.3 Calibrate User Trust in LLM-based Assistants. When learners seek support from LLM-based assistants like ChatGPT, they can encounter incorrect or vague responses that do not fully address their needs [43]. Effective LLM-based assistants should aim to reduce hallucinations by grounding responses in community-sourced information [6, 28], helping learners calibrate an appropriate level of trust in the provided responses [95]. Furthermore, learners' interactions with LLMs for programming and data science learning often occur within broader social and community contexts, rather than in isolation. Research in CSCW and HCI has emphasized the need to contextualize users' trust in AI tools within the broader socio-organizational environment, where relevant collective knowledge and community norms are embedded [15, 25]. Our system builds on prior research showing that RAG can help mitigate hallucinations [53, 82]. In developing ChatCommunity, we adopt these established mechanisms as a technical foundation, while our contribution lies in extending them with social features that surface community knowledge and cues. Rather than aiming to simply enhance users' confidence in AI-generated answers, ChatCommunity is designed to support more appropriate trust calibration by enabling learners to cross-check AI-generated content with community-derived examples. Reducing hallucination is not the focus of our study; however, by using RAG, our system inherits its benefits and may help lessen hallucination-related issues in practice.

3.2 The Design of ChatCommunity

We developed ChatCommunity, a Community-Enriched AI chatbot that provides answers grounded in community-sourced content [6, 28] from the Kaggle platform, previewed with rich community metadata. As shown in Figure 1, we adopted a UI design that aligns with popular LLM chatbots, where users can submit queries through a text box (Figure 1.A). Responses are streamed in real time, include syntax-aware formatting and session memory, with a "New Chat" option to reset.

3.2.1 Iterative Design Process. The development of ChatCommunity followed an iterative design process. Initially, the system provided post previews with links to source content. A pilot study (Appendix B) revealed the need to enhance user engagement, as participants expressed frustration over the lack of community context and recognition for contributors. This led us to include rich social design features — metadata such as author profiles, author names, publish dates, vote counts, view counts, comment counts, and post titles — to credit authors and foster interaction. Based on additional feedback, we added an advanced search panel with ranking options (relevance, vote count, and view count), and allowed users to adjust the number of posts used to generate each response.

3.2.2 Source Document Panel. The source document panel (Figure 1.C) displays a preview of the source post along with rich social design features including author profiles, vote counts, view counts, comment counts, publish date and more. The content preview helps users quickly grasp the relevant information from the source post at a glance, enabling them to decide whether the post is worth interacting with. Vote counts, view counts, and comment counts indicate the post's popularity, while the publish date helps users assess its recency. The author's profile picture fosters a sense of interacting with content created by real people, enhancing engagement with community peers. Together, these rich social design features present what other users select and interact with, fostering a sense of social presence among peers [64]. By making interactions visible, the system engages even passive participants, such as lurkers, by exposing them to valuable content and community activities — the first step toward deeper engagement [75]. This visibility not only facilitates effective communication and collaboration but also enhances the system's trustworthiness, as highlighted by social transparency theory [25, 85] and social presence theory [30, 64]. Overall, the source document panel supports all three design motivations (Sections 3.1.1–3.1.3).

3.2.3 Advanced Search Panel. To balance popularity with relevance, we designed an advanced search panel (Figure 1.D) with ranking modes – relevance, votes, and views. Relevance uses semantic similarity; votes and views integrate popularity signals. This avoids over-amplifying only the most popular content and gives users control to surface a broader range of posts. To grant users greater control, the advanced search panel allows them to adjust the number of posts each response is based on, ranging from 1 to 10. Overall, the advanced search panel helps users efficiently locate relevant and reliable posts, supporting the second design motivation (Section 3.1.2).

3.3 Technical Architecture

Our RAG system is specifically designed for online coding communities like Kaggle, leveraging its public notebook posts to generate responses. The RAG pipeline is illustrated in Figure 3, it comprises two key modules: the retriever and the generator. This section introduces the data source and provides a brief overview of both modules.

3.3.1 Data Source and Preprocessing. Our data sources are the "Meta Kaggle Code" [41] dataset (2023) and the "Meta Kaggle" [58] dataset (2016). Both datasets are officially released by Kaggle and are updated weekly until May 7, 2024. The Meta Kaggle dataset contains Kaggle's public data on competitions, users, submission scores, and kernels, where the Meta Kaggle Code dataset contains the raw source code of the submissions. These datasets, spanning 2015 to 2024, include 4.83 million code files [41, 58], with 4.36 million Jupyter notebooks forming the focus of our study. More details are shown in Appendix F.

We first group the code files by competition and filter files from a selection of four competitions (Appendix F.2). These competitions cover diverse topics, including natural language processing, computer vision, medical data analysis, and business data analysis, ensuring our technical pipeline works effectively with various types of data science competition notebooks.

Although we do not process all the code files due to resource limitations, our pipeline can be easily applied to different competitions by simply providing the competition identification number. We extract non-empty code files in Jupyter Notebook format that were written in Python. In summary, we processed 37,895 code files and obtained 36,945 valid Python-written Jupyter Notebook files, containing 849,900 code cells and 269,855 markdown cells. Our data collection adheres strictly to Kaggle's policies. More details are shown in Appendix F.3.

3.3.2 Retriever and Generator Modules. We construct the Kaggle post database by dividing each notebook into smaller chunks $\mathbf{c_i}$. Each chunk is formed by grouping consecutive markdown cells

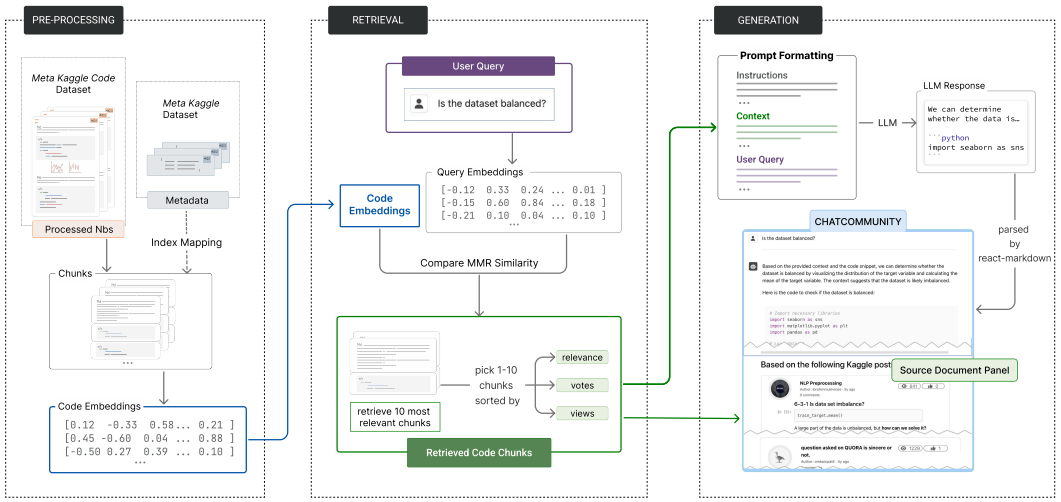


Fig. 3. RAG Pipeline: The process begins with preprocessing the Kaggle notebooks from the Meta Kaggle Code dataset to extract non-empty, Python-written Jupyter notebooks, which are then organized by individual competitions. Each notebook is further divided into several chunks, and these chunks are linked with the corresponding metadata from the Meta Kaggle dataset through indexing. Next, the chunks are passed through an encoder to generate embeddings. The embeddings, along with the corresponding chunk content, are stored in ChromaDB. When a user inputs a query, the query is converted into embeddings, and our system searches for and ranks relevant chunks using user-selected sorting methods (relevance, votes, or views). The retrieved chunks, combined with the user's query, are then formatted into a complete prompt and passed to LLMs to generate a coherent response. These responses are parsed by our system frontend using the react-markdown package and displayed as system outputs. The retrieved chunks, together with their metadata, are also shown in the source document panel below the system response.

followed by the consecutive code cells that immediately follow them. This chunking ensures efficient retrieval and balances prompt length. Each chunk is embedded into a vector representation using the `text-embedding-ada-002` model [66], with embeddings stored in a ChromaDB vector database [1]. As the Meta Kaggle dataset stores social signals (e.g., view counts) for the corresponding notebooks in the Meta Kaggle Code dataset, we use the notebook ID to link each chunk to its respective notebook's metadata, thereby attaching the relevant social signals to the chunks. This ensures that every chunk retains the social context of its original notebook.

When a user submits a query, we retrieve the top 10[1] most relevant chunks using the Maximal Marginal Relevance (MMR) [12] scoring method, which balances relevance and diversity by reducing redundancy. These chunks are ranked based on one of three user-specified criteria: relevance (MMR score), view count, or vote count. The top N chunks, as specified by the user, are then presented with contextual information about their respective Kaggle posts. The default search is based on relevance, giving newer or low-vote posts a chance to display if relevant to the query.

Finally, the retrieved chunks and the user query are combined into a structured prompt, which is processed by the GPT-4o model [65][2]. The model's responses are rendered using the react-markdown [62] package and displayed in a streaming, typewriter manner. For code cells, the raw

[1]The choice of the top 10 is common practice in information retrieval [42].
[2]We choose the GPT-4o model as our task focuses on general data science tasks. At the time of conducting the study, GPT-4o was proven to be one of the best-performing models in data science code generation and mathematics tasks [40, 68] and hence deemed as an appropriate choice.

source is wrapped in fenced Markdown (triple backticks with a language tag such as python) and parsed by react-markdown; these code blocks are then rendered with syntax highlighting using the react-syntax-highlighter [63] package. Markdown cells are passed directly to react-markdown. Retrieved chunks are displayed as source previews in the source document panel. For each chunk, we look up its entry in the metadata dataset using its ID and show the associated information, including the post's URL, which is rendered as a hyperlink that directs the user to the source post. Further details are available in Appendix G.

3.3.3 Implementation. We implemented ChatCommunity as a web application for easy access. The backend is built with Flask and integrates our RAG pipeline. We used LangChain [47] to handle API requests to GPT-4o model and ChromaDB [1] to store embeddings for retrieval within the RAG pipeline. The backend receives messages from the frontend, queries the pipeline, and returns the model's responses together with the retrieved chunks and their corresponding metadata. The frontend was developed from scratch using React. To render responses, we employed the react-markdown [62] package, which directly parses and displays LLM outputs in Markdown format, including both text and code blocks.[3]

4 Study 1: User Study of the Community-Enriched AI Design

To investigate the effectiveness of the Community-Enriched AI design paradigm, we first conducted a user study to address the following research question:

> **RQ1.** *How does the Community-Enriched AI design influence users' engagement with online coding communities, their perceived reliability, and their performance on data science tasks?*

We employed a within-subjects study with four experimental conditions, involving 28 data science learners who had prior experience with Kaggle or similar coding competitions. Participants were instructed to solve four tasks within a single Kaggle competition, using four different assistance methods (as shown in Figure 1), each constituting a distinct experimental condition.

4.1 Four Assistance Methods

To understand how learners perceive the benefits of Community-Enriched AI design, we implemented three chatbot variants for comparison, using the same study apparatus (as shown in Figure 1.). As a baseline condition, we also included a setting where users directly utilize the Kaggle platform without chatbot assistance. During the study, we referred to the conditions by their names (Alpha, Beta, Gamma, Delta) when introducing the session to participants to minimize any bias associated with the names.

- **Community-Enriched AI (Alpha)**: Community-Enriched AI incorporates all the designs and functions mentioned in Section 3.2. It is equipped with our specially designed RAG model and implements the Community-Enriched AI design paradigm;
- **RAG-baseline (Beta)**: RAG-baseline utilizes the same RAG model as Community-Enriched AI, but without the source document panel nor the advanced search panel. By omitting these, we can evaluate whether the presence of these elements impacts users' performance on data science tasks and their perceptions of the system;
- **GPT-4o-based mode (Gamma)**: In this setting, we leverage the GPT-4o API to generate responses to user queries. GPT-4o relies on its pre-trained language model to produce answers without directly referencing specific posts from coding communities like Kaggle.
- **Browsing mode (Delta)**: In this setting, we provide participants with a link to the current Kaggle competition page, allowing them to freely search and browse content on the platform.

[3]The pipeline is open-sourced at: https://github.com/ETH-PEACH-Lab/ChatCommunity.

4.2 Participants and Recruitment

We conducted a power analysis using G*Power [31] to estimate the required sample size for our within-subjects study design. Based on pilot studies (Appendix B), we assumed an effect size of $f = 0.7$, with a significance level of $\alpha = 0.05$ and power $(1 - \beta)$ of 0.81, resulting in a required sample size of 27 participants. After securing ethical approval, we recruited 29 graduate and undergraduate STEM students through social media platforms (e.g., LinkedIn). Qualified participants are self-identified as experienced data science learners, including 27 graduate students and 2 senior undergraduate, with 20 affiliated with the author's institution, and 27 residing in the author's country. Ages ranged from 19 to 29, with 18 males, 10 females, and 1 undisclosed. One participant who failed to complete questionnaires and appeared distracted was excluded, leaving 28 valid data points. More details are in Table 3.

4.3 Study Protocol

All participants signed the consent form before the study. During the study, participants first received a 10-minute introduction on how to set up the study environment and use ChatCommunity. Participants opened a Google Colab notebook with task descriptions alongside the ChatCommunity system. We asked them to place the notebook on the left half of the screen and ChatCommunity on the right half. To ensure participants fully understood how to use ChatCommunity, we demonstrated each function and then gave them time to explore the system on their own. We began the study only after participants indicated that they were familiar with the system. During this time, we encouraged them to ask any questions they had about using ChatCommunity.

Participants then completed four data science tasks, each with a time limit of 13 minutes and representing a typical challenge in data science projects. For each task, participants answered two questions using one of the four assisting methods. The first task required participants to write a few lines of Python code to inspect the task-related dataset and answer questions about its properties, whereas the remaining three tasks asked participants to respond to conceptual questions without writing code. Details about the task questions are listed in Appendix Table 5. We used a Balanced Latin Square design [7] to arrange the order of assisting methods for each participant, minimizing the order effects of the methods.

After each task, participants completed a Likert-scale post-task questionnaire to evaluate the assisting method. After completing all tasks, participants filled out a post-session questionnaire ranking the methods by usefulness and reliability. We also conducted 10-minute semi-structured exit interviews and collected completed notebooks for analysis.

The study lasted 60–90 minutes and was conducted virtually via Zoom. Participants used the chatbots in their own browsers, completed tasks in a provided Google Colab notebook. Each received 47 USD as compensation.

4.4 Study Task

We selected the "Quora Insincere Questions Classification" competition [76] for our user study due to its straightforward nature as a binary classification problem. We also ensured that none of the participants had previously worked on this competition through the screening survey. Grounded in this competition, we designed a set of questions tailored to key stages in data science workflow: data loading, data preprocessing, model development, and model evaluation. We set a time limit of 13 minutes for each task, based on our pilot study, where we found that participants could complete the tasks within this time. We designed two data-oriented or decision-making questions crucial for solving data science problems in each task, as shown in Appendix C. These questions were

reviewed by two data science experts and reflect common challenges encountered in data science projects.

4.5 Data Collection and Analysis

This study collected data from multiple sources, including screening questionnaires, notebook results, post-task and post-session questionnaires, exit interviews, observation notes, and system usage logs. We report the significance levels (p) and test statistics (Z) for each statistical test in this section and Section 5.

4.5.1 Task Performance. We evaluate the task performance through the notebook grades and task completion time. Notebook grades were assigned on a scale from 0 to 3, where 3 indicates a fully correct, complete, and relevant answer. Two researchers independently graded each notebook's results using predefined evaluation criteria (Appendix D). Out of 224 grades, there were five discrepancies between the researchers. The two researchers discussed these differences and reached a consensus on the final grades.

Regarding notebook grades, the linear regression analysis showed no learning effect between tasks ($p = 0.53$) [49]. The Shapiro–Wilk test indicated grades were not normally distributed. We then used Friedman tests to analyze the effects of task type and assisting method. The analysis revealed significant effects of both task type ($p<0.01$, $Z=13.34$), and assisting method ($p<0.01$, $Z=33.87$), on notebook grades. We normalized grades within each task using Z-scores [4] to minimize the effects of task type. As participants in the Alpha condition had higher average grades, we performed a post-hoc analysis using a one-sided Wilcoxon signed-rank test with Bonferroni correction, comparing Alpha to other conditions.

Regarding task completion time, the Shapiro–Wilk test showed they were not normally distributed. Friedman tests were conducted and revealed significant effects of assisting method ($p < 0.01$, $Z = 24.66$), but no significant effect of task type ($p = 0.49$, $Z = 2.40$), on task completion time. Therefore, we conducted a post-hoc analysis using a two-sided Wilcoxon signed-rank test with Bonferroni correction on task completion time, comparing Alpha to other conditions. The analysis results of both notebook grades and task completion time are displayed in Table 1. Detailed grading results for each question are provided in Appendix C.

4.5.2 Post-Task Questionnaires and Post-Session Questionnaires. The post-task questionnaires used 7-point Likert scales to evaluate assisting methods on various attributes. The post-session questionnaire contains two ranking questions, with results shown in Figure 5. The Shapiro–Wilk test showed none of the post-task questionnaire scores were normally distributed. Friedman tests revealed no significant effect of task type on any score, while the assisting method had a significant effect across all scores. As Alpha consistently had a higher mean score than the other conditions, we performed one-sided post-hoc Wilcoxon signed-rank tests with Bonferroni correction to compare Alpha with other conditions. Results are in Table 2.

4.5.3 Exit Interview. We conducted exit interviews after the session, recording 316 minutes of video to gather additional feedback on the assisting methods. We transcribed and proofread participants' answers and the recordings to produce the interview transcripts. We use the interview transcripts as anecdotal evidence to support our quantitative findings. Therefore, we used *in vivo* [78] coding to analyze the interviews and attune ourselves to the users' perspectives on the different designs.

4.5.4 Observation Notes and Usage Logs. During each session, a researcher took observational notes while the system recorded participants' queries, responses, and function usage. These data were used to capture user behavior and provide insights into how participants interacted with

the different methods. The screen recordings were also transcribed and used to review participant behaviors.

5 Study 1: Findings

5.1 How Does Community-Enriched AI Encourage Engagement in Online Coding Communities?

We define participation in an online coding community as behaviors such as viewing, voting, commenting, contacting peers, or contributing posts to the community [37, 94]. Beyond increased viewing through chatbot previews, most participants (22) clicked on the previews to read the original Kaggle posts in Alpha, leading to a total of 77 posts read, with each participant clicking on more than two posts on average. We also observed that two participants gave votes to three posts that they found useful. We asked participants for their reasons for clicking on post previews, which included verifying the system response (10), viewing more details of the post (6), and interacting with the post author (4). This suggests that the source document panel design drives specific engagement behaviors by encouraging participants to verify AI outputs for trust, pursue detailed insights, and initiate direct interactions with post authors, highlighting its role in supporting meaningful community engagement. In terms of prompt iteration, participants in the Alpha condition submitted the highest number of input prompts (M = 5.25, SD = 2.84), followed by those in the Gamma (M = 4.04, SD = 2.22) and Beta (M = 3.96, SD = 1.75) conditions. This pattern is consistent with our observational notes: participants in the Alpha condition frequently engaged with post previews and original posts to gather additional information and used the advanced search panel to refine their retrieval strategies. On average, each participant in the Alpha condition used the advanced search panel more than once. Additional behavioral statistics are available in Appendix H.

We further probed into how this design affects participants as potential contributors to the posts. Nearly all participants (27) preferred having their past and future posts linked by the chatbot system, with most (22) already having experience contributing posts to online communities. Participants recognized that having their posts linked by the chatbot would lead to more people reading their posts (12) and provide help to more people (9), potentially increasing engagement in the community. Participants also highlighted conditions for linking their posts, including only linking their "*open-sourced posts*" (P1, P26) and ensuring that the "*author's name is displayed*" (P8, P15). Most participants (22) affirmed that having their public posts explicitly referenced by the chatbot, potentially increasing views, likes, or comments from peers, would motivate them to contribute more to the community. Participants felt that this design would allow more people to use and interact with their posts, enabling real communication, encouraging them to contribute more. As P4 mentioned: "*thanks to the chatbot, more people would know [about my post]; so yeah, it would definitely encourage me*", and as P7 said: "*Its kind of like you not talking to the wall, you are talking to people, so its better*".

5.2 How Do Participants Perceive the Reliability of Community-Enriched AI?

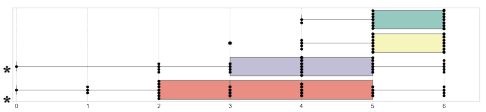
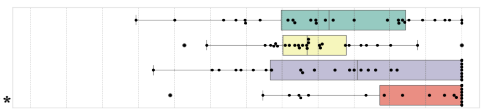
We probe into perceived reliability by analyzing the results of post-session questionnaires, post-task questionnaires, and interviews. As shown in Figure 5, all participants ranked Alpha as the most (18) or second most (10) reliable method.

When comparing Alpha with Beta, Alpha has significantly higher scores in "information provided is reliable" ($p < 0.01$, $Z = 1.32$) and "felt confident using the method" ($p < 0.01$, $Z = 0.03$) according to post-hoc analysis. There are 21 participants mentioned that displaying the source post previews along with the social features made the system's responses more reliable, as it "*helps to check whether it gives the correct answer*" (P1) and "*could see what Kaggle posts were used by the chatbot*"

(P16). Our observation notes further support this. For example, during the task with Alpha, P1 first read through the model's response, then examined each post preview, repeatedly switching back and forth between the response and the previews to cross-check the information. Similarly, P16 read the model's response, clicked on a post preview with a high view count (798) to open the original Kaggle post, compared its content with the chatbot's answer, and subsequently enriched their own task response. This result demonstrates that, even with identical generated responses in the two conditions, Community-Enriched AI fosters perceived reliability by improving social transparency [25, 85].

This aligns with participants' high reliability ratings for Delta, where 14 participants ranked it as the first (8) or second (6) most reliable method. Participants appreciated reading content and comments written by "real people", as mentioned by P11: "*Best thing was that there were so many different posts and discussions. Was nice to see real people thinking about the problem*". In the post-hoc analysis, on comparing Alpha with Gamma, we found Alpha has significantly higher score in "information provided is reliable"($p < 0.01$, $Z = 2.78$) and "felt confident using the method"($p < 0.01$, $Z = 0.96$). During interviews, 15 participants explicitly mentioned that Gamma was not reliable. Participants described they found Gamma less reliable than Alpha because Gamma did not provide source posts as references, and its responses were sometimes not relevant and off-topic. As P2 mentioned: "*no reference make it less reliable*", and as P15 said: "*gives weird answers. It is a little bit off-topic.*"

Table 1. Notebook grades and task completion time analysis. We used Friedman tests to assess the effects of task type and condition on notebook grades and task completion time. Significant effects were found for task type (p < 0.01) and condition (p < 0.01) on grades, and method on task completion time (p < 0.01), but no effect of task on task completion time (p = 0.49). Post-hoc one-sided Wilcoxon-Signed-Rank test with Bonferroni correction on Z-scores, which were normalized to account for task-related effects, revealed that participants using Alpha condition had significantly higher grades than Gamma and Delta. While the post-hoc two-sided Wilcoxon Signed-Rank test with Bonferroni correction on task completion time found that participants in the Delta condition had significantly higher task completion times than those in the Alpha condition.

Measurement	Cond.	N	M	SD	p1	p2	Value
Notebook grades	Alpha	28	5.61	0.63	**0.00****	-	
	Beta	28	5.21	0.83		0.08	
	Gamma	28	4.00	1.52		**0.00****	
	Delta	28	3.32	1.74		**0.00****	
Completion time (mins)	Alpha	28	9.52	2.50	**0.00***	-	
	Beta	28	8.88	1.60		0.45	
	Gamma	28	9.95	2.66		1.00	
	Delta	28	11.57	2.25		**0.01****	

5.3 How Does Community-Enriched AI Improve Participants' Performance in Data Science Tasks?

We probe into user performance by analyzing notebook grades and task completion time as presented in Table 1. For task completion time, the post-hoc analysis revealed that students using Alpha took significantly less time than Delta ($p = 0.01$, $Z = -3.36$). Most participants (26) mentioned that Delta is ineffective and time-consuming, as it is hard to find posts relevant to their task. For notebook grades, participants in the Alpha condition had the highest average grade across the four tasks. The results of post-hoc analysis revealed that students using Alpha had significantly

higher grades than Gamma ($p < 0.01$, $Z = 4.01$) and Delta ($p < 0.01$, $Z = 4.42$). When analyzing task types (detailed in Appendix C), we found that participants performed similarly on data-oriented questions (e.g., Is the training dataset balanced?), but their performance varied significantly on decision-making questions (e.g., Which model should we use and why?). This indicates that Community-Enriched AI might be particularly effective in supporting higher-order cognitive tasks, which require synthesizing information, evaluating options, and justifying choices, compared to data-oriented questions that involve basic comprehension and retrieval of factual information.

In addition, we probe into participants' perceived learning. As shown in Table 2. Alpha has significantly higher score in "helpful in learning coding" comparing to other conditions. While our evidence does not directly prove its effectiveness for learning, the system's ability to encourage reading more posts may lead to potential learning gains [57].

5.4 Perceived Usefulness and Usability with Community-Enriched AI

Lastly, we reported perceived usefulness [20] and perceived usability [52] by analyzing the results of post-session questionnaires, post-task questionnaires, and exit interviews. Most participants (26) ranked Alpha as the most useful method compared to others, as shown in Figure 5. Table 2 presents the statistical results of the post-task questionnaires. In summary, participants find Community-Enriched AI more useful and have higher usability than others.

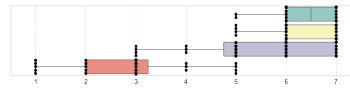
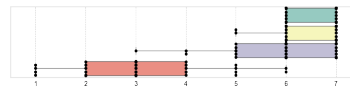
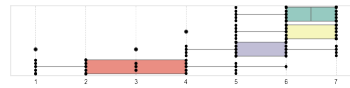
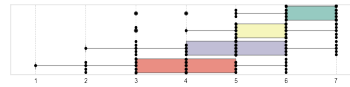
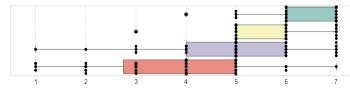
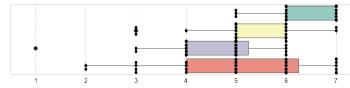
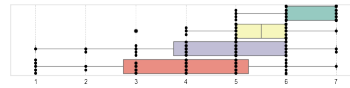
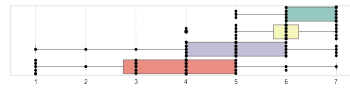
When comparing Alpha with all other conditions, it scored significantly higher in "helpful for solving coding problems" and "liking to use frequently." Compared to Gamma and Delta, participants rated Alpha significantly better in "easy to use," "easy to learn," and "enjoyable to use" according to post-hoc analysis. Most participants (24) mentioned that Alpha, by displaying the source preview, made the system more useful compared to Beta. Reasons mentioned by participants include that Alpha "*provides more evidence... which are good for solving tasks*" (P7) and allows users to "*get useful solutions from others*" (P2). That is, participants preferred receiving the original sources from Alpha because the posts provided evidence to support the response and helped them find useful solutions from others.

Additionally, 20 participants emphasized that providing advanced search panel enhances the system's usefulness. P8 said this panel offers more options and makes the information more relevant, while P5 said it increased their trust by including ranking options based on vote and view counts. During the study, 21 participants used the advanced search panel to switch posts rankings from relevance to votes (16) or views (4) and increase the number of retrieved posts (16).

There are 23 participants mentioned that Gamma is not useful for completing data science tasks comparing to Alpha, given their general (10) and lengthy (7) responses. The median word count for Alpha's responses is 330, compared to 362 for Gamma. While the difference is not significant, Gamma's general responses may increase cognitive load [21, 87], making the responses feel overly lengthy to users.

Similarly, the Delta condition may also impose a high cognitive load by requiring users to browse and interpret large amounts of community posts manually [21, 87]. Most participants (26) mentioned Delta is ineffective and time-consuming. P24 said it's hard to find relevant information with Delta. We observed that four participants gave up Delta midway through the task and attempted to complete it on their own. While participants are allowed to use any function including the search engine in Kaggle, and most (24) participants have used the search engine, they did not find it useful, as P1 mentioned "*It is the least efficient method I can find. Because you need to go through multiple layers of searching...You need to look into it... it's quite tiring.*"

Table 2. Analysis of the post-task questionnaire results: We conducted two Friedman tests to examine the effects of task type and assisting method on the post-task questionnaire responses. The analysis revealed no significant effect of task type on any question, while assisting method had a significant effect across all questions (p-values are reported in column p1). Given that the Alpha condition consistently had a higher mean score than the other conditions, we performed one-sided post-hoc Wilcoxon signed-rank tests with Bonferroni correction for each question, comparing Alpha with the other conditions (p-values are reported in column p2).

Statement	Cond.	N	M	SD	p1	p2	Agreement: 1 to 7
The current assisting method is easy to use.	Alpha	28	6.43	0.63	**0.00****	-	
	Beta	28	6.25	0.70		0.40	
	Gamma	28	5.43	1.40		**0.00****	
	Delta	28	2.86	1.27		**0.00****	
The current assisting method is easy to learn.	Alpha	28	6.61	0.50	**0.00****	-	
	Beta	28	6.32	0.61		0.14	
	Gamma	28	5.89	1.07		**0.01****	
	Delta	28	3.18	1.49		**0.00****	
The current assisting method is enjoyable to use.	Alpha	28	6.32	0.77	**0.00****	-	
	Beta	28	6.07	0.81		0.31	
	Gamma	28	5.18	1.31		**0.00****	
	Delta	28	2.82	1.52		**0.00****	
The current assisting method is helpful for me in learning coding.	Alpha	28	6.29	1.05	**0.00****	-	
	Beta	28	5.68	1.09		**0.01****	
	Gamma	28	4.93	1.46		**0.00****	
	Delta	28	3.82	1.44		**0.00****	
The current assisting method is helpful for me in solving coding problems.	Alpha	28	6.43	0.84	**0.00****	-	
	Beta	28	5.82	0.90		**0.00****	
	Gamma	28	4.96	1.55		**0.00****	
	Delta	28	3.75	1.60		**0.00****	
The information provided by the current assisting method is reliable.	Alpha	28	6.50	0.69	**0.00****	-	
	Beta	28	5.21	1.03		**0.00****	
	Gamma	28	4.64	1.16		**0.00****	
	Delta	28	5.07	1.63		**0.00****	
I felt very confident using the current assisting method.	Alpha	28	6.43	0.74	**0.00****	-	
	Beta	28	5.36	0.87		**0.00****	
	Gamma	28	4.57	1.55		**0.00****	
	Delta	28	3.89	1.87		**0.00****	
I would like to use this assisting method frequently.	Alpha	28	6.57	0.69	**0.00****	-	
	Beta	28	5.93	0.86		**0.01****	
	Gamma	28	4.89	1.47		**0.00****	
	Delta	28	3.54	1.71		**0.00****	

6 Study 2: Design Exploration Study of Levels of Community Features Integration in AI Chatbots

Through our Study 1, we demonstrate the significance of Community-Enriched AI as a design paradigm and its measurable impact on users' decisions. We show that incorporating community

contributions to AI chatbots' responses encourages users to engage with the community and also enhances their trust and perceived reliability on the responses. However, the specific design elements of these community-enriched features and the extent to which they should be integrated into the design of socially-enhanced AI systems remain underexplored. To better understand the design space of pertinent social features for Community-Enriched AI systems, we conduct Study 2 and pose the following research question:

> **RQ2.** *Which community-enriched design attributes do users value most for fostering trust in AI chatbot responses and encouraging engagement with the community?*

6.1 Study Protocol

To address RQ2 we conducted a qualitative design exploration study examining four levels of community integration in AI chatbots — ranging from minimal cues such as titles and links, to previews and social features, and finally to rich summaries reflecting community consensus and disagreement. This study investigates how the varying degrees of community features shape user preferences, perceptions and their behavior, particularly in fostering trust in AI chatbot responses and promoting community engagement.

6.1.1 Participants. The last twelve participants from the previous study (P17–P28; see Table 3) took part in this study as an extended session following a short break (5–10 minutes) after Study 1. The Study 2 session lasted 30–40 minutes. As a token of appreciation for completing both sessions (total duration: 100–120 minutes), participants received a total of $53 USD, which exceeds the local minimum wage.

6.1.2 Procedure. In this study, each session began with an introduction to four design variations, all of which were implemented as high-fidelity prototypes in Figma using a Wizard-of-Oz approach [19, 71]. This was followed by a task in which participants used each variation to answer a data science question about the same Kaggle competition used in the previous user study: "Which model should we use for this competition?". Following the task, we conducted interviews to understand users' perceptions of the various attributes of the different designs, their perceived benefits and drawbacks for each design variants.

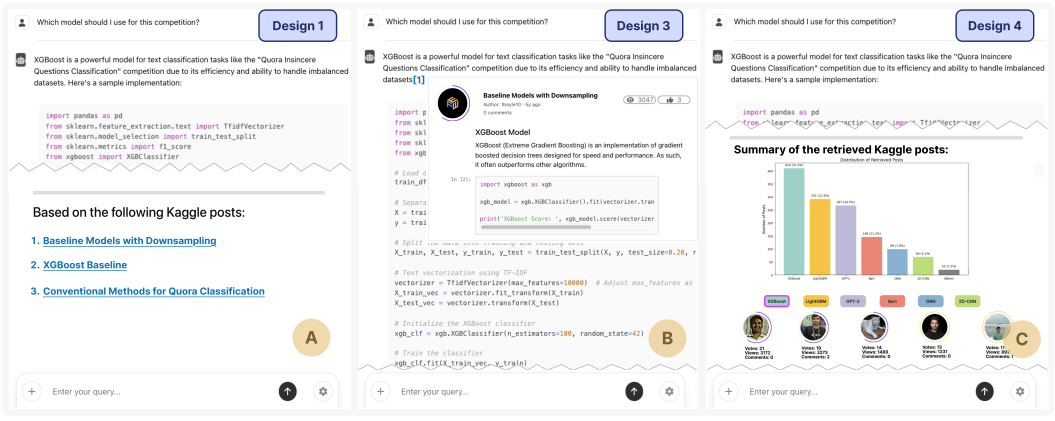


Fig. 4. An overview of design variations: Design 1: Vanilla Link, Design 3: Community-Enriched Inline and Design 4: Community-Enriched Summary. Design 2: Community-Enriched Preview is the same as Figure 1.Alpha.

6.1.3 Design Variations. All four design variations presented the same textual chatbot response and shared the same ranking of relevant posts. Designs 1–3 used the same set of retrieved posts, while Design 4 included additional posts to support each solution. The main differences across the variations lie in how the retrieved posts and associated social design features are presented to the user.

- **Design 1: Vanilla Link.** The Vanilla Link design (Figure 4A) displays the source post as a simple hyperlink to the post title, providing a straightforward and minimal design.
- **Design 2: Community-Enriched Preview.** The Community-Enriched Preview is the same design as Figure 1 Alpha from the previous study, providing post previews along with social features like author profiles, view and vote counts.
- **Design 3: Community-Enriched Inline.** The Community-Enriched Inline (Figure 4B) provides sentence-level references with clickable links to community-enriched previews, offering in-context support. This design was inspired by participant feedback in Study 1, where P6, P9, and P14 emphasized a need for fine-grained inline references to better pinpoint details in the original post.
- **Design 4: Community-Enriched Summary.** The Community-Enriched Summary (Figure 4C) summarizes retrieved posts, allowing users to view the distribution of solutions across all relevant posts. This design is inspired by prior research [106, 107] showing that summarizing online coding community discussions can help users compare alternative solutions more effectively. The top 10 posts, ranked by relevance, vote count, or view count as chosen by the user, are displayed alongside the social design features such as user profiles, vote counts, view counts, and comment counts.

6.1.4 Data Analysis. A total of 306 minutes of interview was recorded. The interviews were transcribed and two authors of the paper analyzed the transcripts using Reflexive Thematic Analysis [9]. The goal of the analysis was to understand which community features, and to what degrees of their integration, shaped users' trust and confidence in the AI chatbots' responses, influenced their community engagement, and provided them with opportunities to engage with the content. The first two authors independently coded five randomly sampled transcripts. They then discussed the codes and reached an agreement level of 97% [59]. The remaining transcripts were divided equally at random and coded independently by the two authors. The resulting codes were jointly discussed and organized into relevant themes.

7 Study 2: Findings

The coding process from our thematic analysis yielded 167 unique codes. These were iteratively interpreted and arranged into four themes. In the following sections, we present each theme with illustrative quotes from participants.

7.1 Fostering trust and reliability through encapsulating community-enriched content

One of the key design attributes emerging from our design variations, and preferred by all users, is the integration of community knowledge with AI-generated assistance — presenting both sources and responses together. Participants reflected that **viewing a comprehensive community enriched context as an overview of posts alongside the chatbot's responses** helped them gain reliability on its response. As P20 describes, "*I think [in Design 2] since knowing who answered this question and when they answer and how many people view it, this information makes me trust the answer more. This is because I can directly see the content of the answer, and also since the source and answer appear together.*" In addition to fostering trust, having access to this contextual information also helped users assess the relevance of the answer to their needs. As P25 noted: "*From the preview*

I could know whether it's related to what I want to find. If it's interesting I would click on that and go to the original post."

Our findings also reveal variation in users' preferences regarding how this encapsulated information should be displayed. All users (12) preferred minimizing the effort required to access community-enriched information by **visualizing it as a preview of relevant posts**, rather than embedding it as inline references within the chatbot's response. This preview-based approach was perceived as more usable for navigating information related to the chatbot's answers. As P18 noted: "*Inline reference discourages users from interacting with posts, users need to do extra interaction to see the post... sometimes you just want all the posts below [chatbot's answers], you can just compare the number of every post, it's quite intuitive and useful for users to check details.*" Participants also suggested that these previews could be further enriched by seeing keywords in the preview relevant to their topic. However, some users also perceived value in inline references as depicted in Design 3, in fostering trust in the AI's responses. This format was seen as similar to citing scientific articles, allowing users to **inspect sources directly at specific points** in the response. Nonetheless, participants felt that while inline references supported transparency, they did not fully convey source credibility or support reliability as effectively as the preview overview.

7.2 Fostering trust and reliability through social attributes of community-enriched AI chatbot responses

As reflected in the previous theme, users expressed that integrating community-enriched information alongside chatbot responses enhances both the perceived reliability and relevance of the answers. This integration allows them to make social inferences by drawing on the diverse attributes present in community knowledge which also contributes to calibrating users' trust in the answers of the AI chatbot. Specifically, social design features from our design variations: the Community-Enriched Posts Preview feature from Design 2 and the Community-Enriched Summary feature from Design 4 contributed to users' reliability and trust in AI's responses. Below we present the social design attributes from these two design variations that users deemed useful and necessary for fostering trust and reliability in AI responses.

First, the users emphasized the importance of **author identity transparency**, such as displaying the author's name, profile information, and image on the posts along with the response of the AI chatbot. These elements acted as cues, helping users trust and feel confident in the AI response. From the context of social transparency in AI systems [25], this social design attribute introduces human elements of decision making in the system that helps shape people's perception of and trust in the AI's response. As P18 reflected, "*[In Design 2]...the post is created by real people. GPT only provides you with articles, you don't have access to the author name, profile picture, you don't know who created the content, making you not confident enough about these contents.*"

Second, the other social design attribute that contributes to building reliability on the AI's response is the visibility of community engagement with the posts, achieved through **interaction transparency** [85] — specifically by displaying interaction metrics such as votes, views and clicks on the post previews as implemented in Design 2. Of these interaction attributes, in particular most users emphasized the importance of votes, followed by metrics on views on the posts as a cues to build a socially-situated significant perception of the AI's answer. P19 explained, "*The like and view feature definitely encourages reliability [on the chatbot's response], it is saying many other people are doing it, other people agree with it.*" Providing information about community engagement allows users to infer the significance of the answer, fostering transitive trust in the AI system by trusting the community's endorsement [25]. Beyond ratings, metrics such as the number of comments on a post preview also helped build transitive trust. As P28 noted, "*Comment is important. More comments means more people review this article.*"

Third, extending the design attributes that foster transitive trust in the AI's responses, our feature of summarizing retrieved posts and presenting the distribution of solutions across all relevant posts was perceived as further reinforcing users' trust and confidence in the chatbot's answers. This design attribute not only makes the technological grounding of the AI's answers visible by anchoring the summary in a "scientific way" — but also allows users to see an **aggregated view of community engagement with the collective information**, infusing the human element to complement AI's reasoning [25]. As P27 shared: "*It makes me feel the decision of clicking links is statistics-based, science-based. I like that.*" By providing an overview of how solutions are being used by others in the community, the design helps users visualize the human element embedded in decision-making, reinforcing trust through social heuristics like social endorsement, As P21 explained: "*[Design 4] shows the comparison from the whole picture — which one is the most used one. We have a comparison, and it provides more confidence regarding reliability.*", or by inferring relevance from others' action, as mentioned by P28, "*More users use this method, I will try to think about why more users use it. Because more users use this, this may be more suitable for this problem.*"

7.3 Fostering social engagement with the community

Findings from our interviews further show that while providing community-enriched information about source posts fosters users' trust in AI responses, it also serves as a bridge between the AI and online coding communities by offering affordances for users to engage with the original community posts.

Firstly, participants expressed that providing the community enriched information about the source posts as **direct visual previews** are more enticing for users to engage with the posts. P17 expressed "*making preview more directly embedded is better for community discussion.*" Secondly, the **social design attributes** of these post previews — highlighted in the previous theme, including author profiles, post titles, metrics capturing others' interactions with the posts, and temporal context — enable users to more readily engage with the posts. As P18 noted, "*The features [in Design 2] definitely encourage you to interact with the community. You naturally like to click on things here...click the post here, comment on something, vote for him if the post is useful.*" These social attributes help users find relevance in AI's answers, which essentially encourages them to read and engage with the posts. And through the transparency of the community attributes as users build transitive trust on AI responses, they are also encouraged to further contribute to the community by expressing their own views and interacting with other community members [75]. This progression reflects the stages described in the *Reader-to-Leader Framework* [75], where users gradually evolve from passive readers to active contributors as they develop trust in the system's responses. As P17 said "*If I find the answer is good, I can click; If I have more time, I would connect with the author, if I find this post is exactly what I need, I would comment on it.*"

However, most of the users also expressed that direct interactions with the community from our chatbot, based on the post previews, are best limited to **small non-conversational contributions** through likes and votes. This preference stemmed from the recognition that the preview alone does not convey the full context necessary for deeper engagement. As P19 explained, "*The chatbot provides only a short preview most of the time, I need to go to the Kaggle page to get the whole context.*" To contribute more substantively or to collaborate meaningfully, participants preferred accessing the complete context of the post with its social attributes **directly within the native online coding community**, where they felt better **equipped to engage responsibly**. As P27 noted, "*I don't think interaction should happen in the chatbot; the comment should come from someone who has really read the post… it feels unjustified or unfair for me to comment on or vote for a post after only reading a preview.*"

7.4 Fostering social knowledge from community enriched AI responses

Finally, our findings show that community-enriched features integrated with AI responses not only enhance the interaction but also enabled users for deeper learning through social knowledge-building. Through the social feature of summarized information on posts, some users experienced a sense of **agency in productive exploration and learning**, enabling them to identify preferred models and methods from the rich insights of community consensus. P24 mentioned "*[Design 4] increases my willingness to explore different posts and methods. It shows me there are also other possibilities and makes my curious about how other people using different method. I would like to use* `lightgbm` *and see how different methods perform.*" Furthermore, having access to collective knowledge from the community, along with AI's response, allows users to **broaden their perspectives and deepen their social learning**. P19 noted "*I think the process of looking for a solution is good for you to learn. Here you search for a solution in the community, you are in the product, you also know what other people are doing...broaden my views and get more information.*" Participants also emphasized that the tool's **community-specific focus** helped them find more "fine-tuned" information and fostered a sense of social connectedness with others.

8 Discussion

8.1 Toward Designing Community-Enriched AI Systems

8.1.1 Significance of Community-Enriched Design in Perceived Trust. Our work demonstrates that bridging online coding communities with AI through a Community-Enriched design can significantly enhance users' trust in AI responses. This trust emerges through two mechanisms: (1) encapsulating community knowledge in AI-generated answers, and (2) displaying social attributes alongside those answers.

Firstly, our Community-Enriched design uses a community-tailored RAG model to retrieve relevant posts to meet diverse user needs. The advanced search panel further leverages social design features such as vote count and view count, helping users identify popular and trustworthy content. This design improves perceived reliability and usability by building on prior work on RAG to reduce hallucinations [53, 82] and lowering extraneous cognitive load [21], compared to GPT-4o and traditional search/browsing methods.

Secondly, by making community knowledge visible through social design features, our design allows users to anchor their trust in the AI's responses based on the collective insights of the community. When users see that the AI's answer aligns with content they recognize as trustworthy, they develop transitive trust [25] in the AI — trusting it because it reflects the community's shared understanding. By embedding visible cues of peer interactions — what users select, vote for, or like, as well as how they engage with the community's shared knowledge — the interface enables users to draw social inferences from community interactions, leading to more informed trust in the AI [25, 85]. Additionally, incorporating identity transparency in design [85], such as showing post authors' profile information, provides additional social signals that help users assess about sources and relevance of the AI's answers. Insights from Study 2 highlight that the level of social attribute visibility also shapes users' trust perceptions. While simple visual previews of source posts help users ground their judgment, enriching these previews with summaries that capture diverse community perspectives provides greater clarity and reinforces a sense of reliability.

8.1.2 Supporting Community Engagement and Learning Agency. In alignment with Cai et al.[11]'s suggestion to raise students' awareness and autonomy in conversational AI, our findings show that Community-Enriched AI not only enhances trust but also supports user engagement and learning agency by surfacing community sourced content that encourages self-directed learning and social participation. By presenting previews of related posts along with social design features, our design

encourages users to critically examine AI responses, compare alternative perspectives, and discover more tailored solutions from the community. This process fosters learner agency by enabling users to make informed decisions rather than passively accepting AI outputs.

In our second study, participants reported that these community-enriched previews lowered the effort to access additional knowledge and made them more inclined to explore and interact with the original posts. Participants shared that they felt comfortable engaging through lightweight actions such as viewing, liking, or voting from within the AI interface but preferred to transition to the full community platform for more substantive contributions. This behavior reflects early stages of engagement described in the Reader-to-Leader framework [75], where users evolve from passive readers to active community participants. Our design thus not only supports surface-level engagement but also provides potentials for deeper involvement and responsible knowledge sharing.

Finally, access to community knowledge was also seen as fostering a sense of social connectedness [70, 96]. Participants emphasized that seeing how others solved similar problems, and understanding which solutions were most used, helped them develop a broader perspective on problem-solving. This collective insight strengthened their learning experience and increased their sense of connectedness within the community.

Our findings also highlight a broader question about where engagement should occur, whether it should remain within the chatbot or return to the online community. Participants appreciated the convenience of lightweight interactions such as viewing or voting from within the chatbot, yet most emphasized that conversations with post authors or other users should take place on the original community platform. They noted that the chatbot previews often lacked the full context needed for responsible commenting or collaboration, and that deeper engagement felt more appropriate within the native environment where posts, authors, and discussions were fully visible. This distinction raises important ecological and ethical considerations about the ownership and sustainability of community knowledge in the age of generative AI: when community data are used to train or augment LLMs, who owns the user and who should own the engagement? Design paradigms like Community-Enriched AI can serve as mediators in this ecosystem: using community data to enhance AI assistance while still directing recognition, credit, and interaction back to the online communities.

8.1.3 Design Takeaways. Drawing on insights from our studies, we offer the following design takeaways for future designers and researchers in creating community-supported AI systems.

- **Community-generated content as source previews**: Comprehensive information of sources grounded in community-generated content should be provided as a deictic context to the AI's response. Specifically, these source information should be organized in visual formats like post-previews to allow users easily gain access to relevant community insights and make informed inferences about the AI's answers.
- **Social attributes presented through transparent design**: Alongside source information, AI interface designs should explicitly present the social attributes of referenced content by incorporating socially transparent features — such as author identity and community interaction transparency [85] to help users build trust in the AI's response.
- **Aggregate community perspectives**: The aggregated engagement data (e.g., votes, views) and community perspectives from source posts should be presented as well-structured summaries to help users make socially informed inferences, identify relevant content, and develop transitive trust in the AI's responses.
- **Embed community information directly within AI responses**: Designs should directly embed visual previews of community posts and their socially transparent design features

within AI responses to encourage user engagement. These embedded previews enable lightweight, non-conversational interactions with the community, such as likes or votes, allowing users to readily engage with the community content.
- **Enable full-context engagement within the community**: While embedded previews focus on non-conversational community interactions, designs should also support seamless transitions to the original platform for deeper, more responsible contributions — recognizing users' need for full context when engaging.

8.2 Ethical Considerations

8.2.1 Data policy. When integrating Community-Enriched AI, it is essential to ensure that the data used for generating responses adheres to ethical and legal standards. Thus, the open sourced Meta-data dataset provides us the opportunity to build and evaluate our ideas. A key feature of ChatCommunity is displaying sources for each response. However, this necessitates careful consideration of the data policies associated with the sources. One participant mentioned, "*as long as these data [posts] are open-sourced*" (P1). This indicates that the participant believes the prerequisite for linking posts with the chatbot is that the posts must already be open-sourced. Future systems should only access publicly available data or content explicitly permitted for public use. Using private or restricted data without proper authorization can lead to ethical and legal issues, especially when the generated responses will link to the source content. Developers must carefully review data sources to ensure compliance with relevant policies and regulations, such as GDPR [2], ensuring that personal or sensitive data is appropriately anonymized or excluded.

8.2.2 Control by users. Empowering users with control over retrieval ranking is a key aspect of maintaining fairness and transparency in ChatCommunity. We provide users with an advanced search panel that offers three objective post retrieval ranking methods: relevance, votes, and views. There are 20 out of 28 participants found that the advanced search panel increased the system's usefulness. Designing retrieval methods requires careful consideration to avoid unintended biases [22]. For example, if not implemented thoughtfully, certain posts could consistently rank at the top, while others are overlooked, leading to skewed visibility and bias in the information presented. To prevent this, future systems building on our work should ensure that the retrieval methods are balanced and do not favor certain types of content disproportionately (e.g., based on business profit).

8.3 Limitations and Future Work

Our studies have a number of limitations. For both of our user studies, we designed tasks based on one Kaggle competition [76], covering stages like data loading, pre-processing, model development, and evaluation. However, it would be valuable to assess ChatCommunity on other tasks and other platforms. Besides, we focused on data-oriented and decision-making questions, which do not represent all possible data science challenges. However, evidence suggests RAG systems improve LLM performance in general question-answering tasks, supporting ChatCommunity's generalizability [53, 83].

In addition, while measuring the perceived reliability, we did not assess the accuracy of the individual AI response. And the chunking method may miss contextual information. However, our contribution focuses on the design paradigm of Community-Enriched AI, which remains applicable even to an ideal RAG model without the need for chunking.

Although ChatCommunity improved perceived learning gains, our studies did not assess actual learning outcomes. Therefore, we cannot make definitive claims about its impact on learning

effectiveness. However, ChatCommunity offers better access to Kaggle posts than typical chatbots, potentially supporting user learning through active reading of community-generated content [57].

Finally, although ChatCommunity is designed to support socially-informed learning, the customization options in the advanced search panel are currently limited to fixed attributes. Future work could explore dynamically adjusting results based on user input. For example, users might describe their needs in natural language — "I only want posts written in PyTorch, ranked by the number of views", and the system could retrieve and rank posts accordingly.

9 Conclusion

This paper introduces Community-Enriched AI, a design paradigm that grounds AI responses in user-contributed content from online coding communities and displays source previews alongside social design features. Through two user studies with 28 and 12 data science learners respectively, we demonstrate that a chatbot implementing this design not only enhances perceived system reliability but also encourages user engagement with the community and effectively supports learners in solving data science tasks. Our findings highlight the potential of Community-Enriched AI to bridge, rather than replace online coding communities — offering design implications for building socially grounded AI assistance systems that facilitate productive social learning.

Acknowledgments

This project was made possible by ETH AI Center Doctoral Fellowships to Junling Wang, with partial support from the ETH Zurich Foundation. Additionally, the authors wish to thank the reviewers, members of the PEACH Lab at ETH Zurich, and the participants in the user study.

References

[1] 2023. GitHub - chroma-core/chroma: the AI-native open-source embedding database — github.com. https://github.com/chroma-core/chroma. [Accessed 03-07-2024].

[2] 2024. General Data Protection Regulation (GDPR) – Legal Text — gdpr-info.eu. https://gdpr-info.eu/. [Accessed 03-09-2024].

[3] Pierpaolo and Suha Shaheen. 2020. Is StackOverflow an Effective Complement to Gaining Practical Knowledge Compared to Traditional Computer Science Learning?. In *Proceedings of the 11th International Conference on Education Technology and Computers* (Amsterdam, Netherlands) *(ICETC '19)*. Association for Computing Machinery, New York, NY, USA, 132–138. doi:10.1145/3369255.3369258

[4] Herve Abdi, Lynne J Williams, et al. 2010. Normalizing data. *Encyclopedia of research design* 1 (2010), 935–938.

[5] Muhammad Asaduzzaman, Ahmed Shah Mashiyat, Chanchal K. Roy, and Kevin A. Schneider. 2013. Answering questions about unanswered questions of Stack Overflow. In *2013 10th Working Conference on Mining Software Repositories (MSR)*. 97–100. doi:10.1109/MSR.2013.6624015

[6] Orlando Ayala and Patrice Bechard. 2024. Reducing hallucination in structured outputs via Retrieval-Augmented Generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, Yi Yang, Aida Davani, Avi Sil, and Anoop Kumar (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 228–238. doi:10.18653/v1/2024.naacl-industry.19

[7] James V Bradley. 1958. Complete counterbalancing of immediate sequential effects in a Latin square design. *J. Amer. Statist. Assoc.* 53, 282 (1958), 525–528.

[8] Joel Brandt, Philip J Guo, Joel Lewenstein, Mira Dontcheva, and Scott R Klemmer. 2009. Two studies of opportunistic programming: interleaving web foraging, learning, and writing code. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1589–1598.

[9] Virginia Braun and Victoria Clarke. 2020. One size fits all? What counts as quality practice in (reflexive) thematic analysis? *Qualitative Research in Psychology* 18, 3 (Aug. 2020), 1–25. doi:10.1080/14780887.2020.1769238

[10] Gordon Burtch, Dokyun Lee, and Zhichen Chen. 2024. The consequences of generative AI for online knowledge communities. *Scientific Reports* 14, 1 (2024), 10413. doi:10.1038/s41598-024-61221-0 Place: England.

[11] Zhenyao Cai, Seehee Park, Nia Nixon, and Shayan Doroudi. 2024. Advancing Knowledge Together: Integrating Large Language Model-based Conversational AI in Small Group Collaborative Learning. In *Extended Abstracts of the CHI*

Conference on Human Factors in Computing Systems (Honolulu, HI, USA) *(CHI EA '24)*. Association for Computing Machinery, New York, NY, USA, Article 37, 9 pages. doi:10.1145/3613905.3650868

[12] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. 335–336.

[13] Bradley Carron-Arthur, John A. Cunningham, and Kathleen M. Griffiths. 2014. Describing the distribution of engagement in an Internet support group by post frequency: A comparison of the 90-9-1 Principle and Zipf's Law. *Internet Interventions* 1, 4 (Oct. 2014), 165–168. doi:10.1016/j.invent.2014.09.003

[14] John Chen, Xi Lu, Yuzhou Du, Michael Rejtig, Ruth Bagley, Mike Horn, and Uri Wilensky. 2024. Learning Agent-based Modeling with LLM Companions: Experiences of Novices and Experts Using ChatGPT & NetLogo Chat. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 141, 18 pages. doi:10.1145/3613904.3642377

[15] John Chen, Xi Lu, Yuzhou Du, Michael Rejtig, Ruth Bagley, Mike Horn, and Uri Wilensky. 2024. Learning Agent-based Modeling with LLM Companions: Experiences of Novices and Experts Using ChatGPT & NetLogo Chat. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 141, 18 pages. doi:10.1145/3613904.3642377

[16] Valerie Chen, Alan Zhu, Sebastian Zhao, Hussein Mozannar, David Sontag, and Ameet Talwalkar. 2025. Need Help? Designing Proactive AI Assistants for Programming. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 881, 18 pages. doi:10.1145/3706598.3714002

[17] Ruijia Cheng, Ruotong Wang, Thomas Zimmermann, and Denae Ford. 2024. "It would work for me too": How Online Communities Shape Software Developers' Trust in AI-Powered Code Generation Tools. *ACM Trans. Interact. Intell. Syst.* 14, 2, Article 11 (may 2024), 39 pages. doi:10.1145/3651990

[18] Ruijia Cheng and Mark Zachry. 2020. Building Community Knowledge In Online Competitions: Motivation, Practices and Challenges. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW2, Article 179 (oct 2020), 22 pages. doi:10.1145/3415250

[19] Nils Dahlbäck, Arne Jönsson, and Lars Ahrenberg. 1993. Wizard of Oz studies: why and how. In *Proceedings of the 1st international conference on Intelligent user interfaces*. 193–200.

[20] Fred D Davis. 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly* (1989), 319–340.

[21] Ton De Jong. 2010. Cognitive load theory, educational research, and instructional design: Some food for thought. *Instructional science* 38, 2 (2010), 105–134.

[22] Alejandro Diaz. 2008. Through the Google goggles: Sociopolitical bias in search engine design. In *Web search: Multidisciplinary perspectives*. Springer, 11–34.

[23] Martin Dittus and Licia Capra. 2017. Private Peer Feedback as Engagement Driver in Humanitarian Mapping. *Proc. ACM Hum.-Comput. Interact.* 1, CSCW, Article 40 (Dec. 2017), 18 pages. doi:10.1145/3134675

[24] Pierpaolo Dondio and Suha Shaheen. 2020. Is StackOverflow an Effective Complement to Gaining Practical Knowledge Compared to Traditional Computer Science Learning?. In *Proceedings of the 11th International Conference on Education Technology and Computers* (Amsterdam, Netherlands) *(ICETC '19)*. Association for Computing Machinery, New York, NY, USA, 132–138. doi:10.1145/3369255.3369258

[25] Upol Ehsan, Q. Vera Liao, Michael Muller, Mark O. Riedl, and Justin D. Weisz. 2021. Expanding Explainability: Towards Social Transparency in AI systems. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 82, 19 pages. doi:10.1145/3411764.3445188

[26] Neuron Engineer. 2018. Handle Overfitting & Error Analysis of Glove-GRU — kaggle.com. https://www.kaggle.com/code/ratthachat/handle-overfitting-error-analysis-of-glove-gru. [Accessed 11-09-2024].

[27] Thomas Erickson and Wendy A. Kellogg. 2000. Social translucence: an approach to designing systems that support social processes. *ACM Trans. Comput.-Hum. Interact.* 7, 1 (March 2000), 59–83. doi:10.1145/344949.345004

[28] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 6491–6501.

[29] Jingchao Fang, Jia-Wei Liang, and Hao-Chuan Wang. 2023. How People Initiate and Respond to Discussions Around Online Community Norms: A Preliminary Analysis on Meta Stack Overflow Discussions. In *Companion Publication of the 2023 Conference on Computer Supported Cooperative Work and Social Computing* (Minneapolis, MN, USA) *(CSCW '23 Companion)*. Association for Computing Machinery, New York, NY, USA, 221–225. doi:10.1145/3584931.3606966

[30] Rosta Farzan, Laura A. Dabbish, Robert E. Kraut, and Tom Postmes. 2011. Increasing commitment to online communities by designing for social presence. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work* (Hangzhou, China) *(CSCW '11)*. Association for Computing Machinery, New York, NY, USA, 321–330.

doi:10.1145/1958824.1958874

[31] Franz Faul, Edgar Erdfelder, Axel Buchner, and Albert-Georg Lang. 2009. Statistical power analyses using G* Power 3.1: Tests for correlation and regression analyses. *Behavior research methods* 41, 4 (2009), 1149–1160.

[32] Christopher Flathmann, Wen Duan, Nathan J. Mcneese, Allyson Hauptman, and Rui Zhang. 2024. Empirically Understanding the Potential Impacts and Process of Social Influence in Human-AI Teams. *Proc. ACM Hum.-Comput. Interact.* 8, CSCW1, Article 49 (April 2024), 32 pages. doi:10.1145/3637326

[33] Denae Ford, Kristina Lustig, Jeremy Banks, and Chris Parnin. 2018. "We Don't Do That Here": How Collaborative Editing with Mentors Improves Engagement in Social Q&A Communities. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3173574.3174182

[34] Saskia Gilmer, Avinash Bhat, Shuvam Shah, Kevin Cherry, Jinghui Cheng, and Jin L.C. Guo. 2023. SUMMIT: Scaffolding Open Source Software Issue Discussion Through Summarization. *Proc. ACM Hum.-Comput. Interact.* 7, CSCW2, Article 297 (Oct. 2023), 27 pages. doi:10.1145/3610088

[35] Kira V Gudkova. 2021. Developing argumentative literacy and skills in ESP students. *Journal of Teaching English for Specific and Academic Purposes* (2021), 229–237.

[36] Jacek Gwizdka. 2010. Distribution of cognitive load in web search. *Journal of the American Society for Information Science and Technology* 61, 11 (2010), 2167–2187.

[37] Jazlyn Hellman, Jiahao Chen, Md. Sami Uddin, Jinghui Cheng, and Jin L. C. Guo. 2022. Characterizing user behaviors in open-source software user forums: an empirical study. In *Proceedings of the 15th International Conference on Cooperative and Human Aspects of Software Engineering* (Pittsburgh, Pennsylvania) *(CHASE '22)*. Association for Computing Machinery, New York, NY, USA, 46–55. doi:10.1145/3528579.3529178

[38] Raphael Hoffmann, James Fogarty, and Daniel S Weld. 2007. Assieme: finding and leveraging implicit references in a web search interface for programmers. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*. 13–22.

[39] Keman Huang, Jilei Zhou, and Shao Chen. 2022. Being a Solo Endeavor or Team Worker in Crowdsourcing Contests? It is a Long-term Decision You Need to Make. *Proc. ACM Hum.-Comput. Interact.* 6, CSCW2, Article 494 (nov 2022), 32 pages. doi:10.1145/3555595

[40] Zhen Huang, Zengzhi Wang, Shijie Xia, and Pengfei Liu. 2024. OlympicArena medal ranks: Who is the most intelligent AI so far? *arXiv preprint arXiv:2406.16772* (2024).

[41] Megan Risdal Jim Plotts. 2023. Meta Kaggle Code — kaggle.com. https://www.kaggle.com/datasets/kaggle/meta-kaggle-code. [Accessed 21-08-2024].

[42] Steve Jones and Mark S Staveley. 1999. Phrasier: a system for interactive document retrieval using keyphrases. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. 160–167.

[43] Samia Kabir, David N. Udo-Imeh, Bonan Kou, and Tianyi Zhang. 2024. Is Stack Overflow Obsolete? An Empirical Study of the Characteristics of ChatGPT Answers to Stack Overflow Questions. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 935, 17 pages. doi:10.1145/3613904.3642596

[44] Majeed Kazemitabaar, Runlong Ye, Xiaoning Wang, Austin Zachary Henley, Paul Denny, Michelle Craig, and Tovi Grossman. 2024. Codeaid: Evaluating a classroom deployment of an llm-based programming assistant that balances student and educator needs. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–20.

[45] Robert E. Kraut and Paul Resnick. 2011. *Building Successful Online Communities: Evidence-Based Social Design*. MIT Press. Google-Books-ID: IIvBMYVxWJYC.

[46] Tuan Manh Lai, Quan Hung Tran, Trung Bui, and Daisuke Kihara. 2020. A simple but effective bert model for dialog state tracking on resource-limited systems. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 8034–8038.

[47] LangChain. 2025. GitHub - langchain-ai/langchain: Build context-aware reasoning applications — github.com. https://github.com/langchain-ai/langchain. [Accessed 28-08-2025].

[48] Jean Lave and Etienne Wenger. 2001. Legitimate peripheral participation in communities of practice. In *Supporting Lifelong Learning*. Routledge. Num Pages: 16.

[49] Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. 2017. *Research methods in human-computer interaction*. Morgan Kaufmann.

[50] Hyunji Lee, Se June Joo, Chaeeun Kim, Joel Jang, Doyoung Kim, Kyoung-Woon On, and Minjoon Seo. 2024. How Well Do Large Language Models Truly Ground?. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 2437–2465. doi:10.18653/v1/2024.naacl-long.135

[51] Yu-Wei Lee, Fei-Ching Chen, and Huo-Ming Jiang. 2006. Lurking as participation: a community perspective on lurkers' identity and negotiability. In *Proceedings of the 7th International Conference on Learning Sciences* (Bloomington, Indiana) *(ICLS '06)*. International Society of the Learning Sciences, 404–410.

[52] James R Lewis. 2018. The system usability scale: past, present, and future. *International Journal of Human–Computer Interaction* 34, 7 (2018), 577–590.

[53] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS '20)*. Curran Associates Inc., Red Hook, NY, USA, Article 793, 16 pages.

[54] Guangqiang Lu. 2018. Advanced Deep Models with pretrained embeddings — kaggle.com. https://www.kaggle.com/code/manrunning/advanced-deep-models-with-pretrained-embeddings. [Accessed 10-09-2024].

[55] Shuai Ma, Junling Wang, Yuanhao Zhang, Xiaojuan Ma, and April Yi Wang. 2025. DBox: Scaffolding Algorithmic Programming Learning through Learner-LLM Co-Decomposition. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 585, 20 pages. doi:10.1145/3706598.3713748

[56] Pratyusha Maiti and Ashok Goel. 2025. Can an AI Partner Empower Learners to Ask Critical Questions?. In *Proceedings of the 30th International Conference on Intelligent User Interfaces (IUI '25)*. Association for Computing Machinery, New York, NY, USA, 314–324. doi:10.1145/3708359.3712134

[57] Karen Manarin. 2019. Why read? *Higher Education Research & Development* 38, 1 (2019), 11–23.

[58] Timo Bozsolik Megan Risdal. 2023. Meta Kaggle — kaggle.com. https://www.kaggle.com/datasets/kaggle/meta-kaggle. [Accessed 21-08-2024].

[59] Matthew B. Miles, A. Michael Huberman, and Johnny Saldaña. 2020. *Qualitative Data Analysis: A Methods Sourcebook* (4th ed.). SAGE Publications, Thousand Oaks, CA.

[60] Mojtaba Mostafavi Ghahfarokhi, Arash Asgari, Mohammad Abolnejadian, and Abbas Heydarnoori. 2024. DistilKaggle: A Distilled Dataset of Kaggle Jupyter Notebooks. In *Proceedings of the 21st International Conference on Mining Software Repositories* (Lisbon, Portugal) *(MSR '24)*. Association for Computing Machinery, New York, NY, USA, 647–651. doi:10.1145/3643991.3644882

[61] Blair Nonnecke and Jenny Preece. 2000. Lurker demographics: counting the silent. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands) *(CHI '00)*. Association for Computing Machinery, New York, NY, USA, 73–80. doi:10.1145/332040.332409

[62] npmjs. 2025. react-markdown — npmjs.com. https://www.npmjs.com/package/react-markdown. [Accessed 27-08-2025].

[63] npmjs. 2025. react-syntax-highlighter — npmjs.com. https://www.npmjs.com/package/react-syntax-highlighter. [Accessed 14-09-2025].

[64] Catherine S Oh, Jeremy N Bailenson, and Gregory F Welch. 2018. A systematic review of social presence: Definition, antecedents, and implications. *Frontiers in Robotics and AI* 5 (2018), 409295.

[65] OpenAI. 2024. GPT-4o. https://platform.openai.com/docs/models/gpt-4o. [Accessed 01-07-2024].

[66] OpenAI. 2024. New embedding models and API updates. https://openai.com/index/new-embedding-models-and-api-updates/

[67] Andrew L Oros. 2007. Let's debate: Active learning encourages student participation and critical thinking. *Journal of Political Science Education* 3, 3 (2007), 293–311.

[68] Shuyin Ouyang, Dong Huang, Jingwen Guo, Zeyu Sun, Qihao Zhu, and Jie M Zhang. 2025. DSCodeBench: A Realistic Benchmark for Data Science Code Generation. *arXiv preprint arXiv:2505.15621* (2025).

[69] Hyanghee Park and Daehwan Ahn. 2024. The Promise and Peril of ChatGPT in Higher Education: Opportunities, Challenges, and Design Implications. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24)*. Association for Computing Machinery, New York, NY, USA, 1–21. doi:10.1145/3613904.3642785

[70] SangAh Park, Yoon Young Lee, Soobin Cho, Minjoon Kim, and Joongseek Lee. 2021. "Knock Knock, Here Is an Answer from Next Door": Designing a Knowledge Sharing Chatbot to Connect Residents: Community Chatbot Design Case Study. In *Companion Publication of the 2021 Conference on Computer Supported Cooperative Work and Social Computing* (Virtual Event, USA) *(CSCW '21 Companion)*. Association for Computing Machinery, New York, NY, USA, 144–148. doi:10.1145/3462204.3481738

[71] Martin Porcheron, Joel E Fischer, and Stuart Reeves. 2021. Pulling back the curtain on the wizards of Oz. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW3 (2021), 1–22.

[72] James Prather, Brent N Reeves, Juho Leinonen, Stephen MacNeil, Arisoa S Randrianasolo, Brett A Becker, Bailey Kimmel, Jared Wright, and Ben Briggs. 2024. The Widening Gap: The Benefits and Harms of Generative AI for Novice Programmers. In *Proceedings of the 2024 ACM Conference on International Computing Education Research-Volume 1*. 469–486.

[73] Gabriel Preda. 2024. Kaggle Users Evolution — kaggle.com. https://www.kaggle.com/code/gpreda/kaggle-users-evolution?scriptVersionId=186795574. [Accessed 04-07-2024].
[74] Jenny Preece, Blair Nonnecke, and Dorine Andrews. 2004. The top five reasons for lurking: improving community experiences for everyone. *Computers in human behavior* 20, 2 (2004), 201–223.
[75] Jennifer Preece and Ben Shneiderman. 2009. The Reader-to-Leader Framework: Motivating Technology-Mediated Social Participation. *AIS Transactions on Human-Computer Interaction* 1, 1 (March 2009), 13–32. https://aisel.aisnet.org/thci/vol1/iss1/5 Number: 1.
[76] Quora. 2018. Quora Insincere Questions Classification — kaggle.com. https://www.kaggle.com/c/quora-insincere-questions-classification. [Accessed 05-07-2024].
[77] Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar, S.M Towhidul Islam Tonmoy, Aman Chadha, Amit Sheth, and Amitava Das. 2023. The Troubling Emergence of Hallucination in Large Language Models - An Extensive Definition, Quantification, and Prescriptive Remediations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 2541–2573. doi:10.18653/v1/2023.emnlp-main.155
[78] Johnny Saldaña. 2013. *The coding manual for qualitative researchers* (2nd ed.). SAGE, Los Angeles.
[79] Subhasree Sengupta. 2020. 'Learning to code in a virtual world': A Preliminary Comparative Analysis of Discourse and Learning in Two Online Programming Communities. In *Companion Publication of the 2020 Conference on Computer Supported Cooperative Work and Social Computing* (Virtual Event, USA) *(CSCW '20 Companion)*. Association for Computing Machinery, New York, NY, USA, 389–394. doi:10.1145/3406865.3418319
[80] Subhasree Sengupta and Caroline Haythornthwaite. 2020. Learning with Comments: An Analysis of Comments and Community on Stack Overflow. In *Hawaii International Conference on System Sciences*. https://api.semanticscholar.org/CorpusID:213162741
[81] Nischal Shrestha, Titus Barik, and Chris Parnin. 2021. Remote, but Connected: How #TidyTuesday Provides an Online Community of Practice for Data Scientists. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW1, Article 52 (apr 2021), 31 pages. doi:10.1145/3449126
[82] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval Augmentation Reduces Hallucination in Conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, Punta Cana, Dominican Republic, 3784–3803. doi:10.18653/v1/2021.findings-emnlp.320
[83] Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the Domain Adaptation of Retrieval Augmented Generation (RAG) Models for Open Domain Question Answering. *Transactions of the Association for Computational Linguistics* 11 (2023), 1–17. doi:10.1162/tacl_a_00530
[84] Claire Stansfield, Kelly Dickson, and Mukdarut Bangpan. 2016. Exploring issues in the conduct of website searching and other online sources for systematic reviews: how can we be systematic? 5, 1 (2016), 191. doi:10.1186/s13643-016-0371-9
[85] H Colleen Stuart, Laura Dabbish, Sara Kiesler, Peter Kinnaird, and Ruogu Kang. 2012. Social transparency in networked information exchange: a theoretical framework. In *Proceedings of the ACM 2012 conference on computer supported cooperative work*. 451–460.
[86] Weiwei Sun, Zhengliang Shi, Shen Gao, Pengjie Ren, Maarten de Rijke, and Zhaochun Ren. 2023. Contrastive learning reduces hallucination in conversations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 13618–13626.
[87] John Sweller. 2011. CHAPTER TWO - Cognitive Load Theory. In *Psychology of learning and motivation*, Jose P. Mestre and Brian H. Ross (Eds.). Psychology of Learning and Motivation, Vol. 55. Academic Press, 37–76. doi:10.1016/B978-0-12-387691-1.00002-8
[88] Thitaree Tanprasert, Sidney S Fels, Luanne Sinnamon, and Dongwook Yoon. 2024. Debate Chatbots to Facilitate Critical Thinking on YouTube: Social Identity and Conversational Style Make A Difference. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 805, 24 pages. doi:10.1145/3613904.3642513
[89] Yla Tausczik and Ping Wang. 2017. To Share, or Not to Share? Community-Level Collaboration in Open Innovation Contests. *Proc. ACM Hum.-Comput. Interact.* 1, CSCW, Article 100 (Dec. 2017), 23 pages. doi:10.1145/3134735
[90] Valerio Terragni and Pasquale Salza. 2022. APIzation: generating reusable APIs from StackOverflow code snippets. In *Proceedings of the 36th IEEE/ACM International Conference on Automated Software Engineering* (Melbourne, Australia) *(ASE '21)*. IEEE Press, 542–554. doi:10.1109/ASE51524.2021.9678576
[91] Laton Vermette, Shruti Dembla, April Y. Wang, Joanna McGrenere, and Parmit K. Chilana. 2017. Social CheatSheet: An Interactive Community-Curated Information Overlay for Web Applications. *Proc. ACM Hum.-Comput. Interact.* 1, CSCW, Article 102 (Dec. 2017), 19 pages. doi:10.1145/3134737

[92] Stacy E Walker. 2003. Active learning strategies to promote critical thinking. *Journal of athletic training* 38, 3 (2003), 263.
[93] Thiemo Wambsganss, Matthias Soellner, Kenneth R Koedinger, and Jan Marco Leimeister. 2022. Adaptive Empathy Learning Support in Peer Review Scenarios. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 227, 17 pages. doi:10.1145/3491102.3517740
[94] Hua Wang, Jae Eun Chung, Namkee Park, Margaret L McLaughlin, and Janet Fulk. 2012. Understanding online community participation: A technology acceptance perspective. *Communication Research* 39, 6 (2012), 781–801.
[95] Junling Wang, Hongyi Lan, Xiaotian Su, Mustafa Doga Dogan, and April Wang. 2026. UI Remix: Supporting UI Design Through Interactive Example Retrieval and Remixing. In *Proceedings of the 31st International Conference on Intelligent User Interfaces* (Paphos, Cyprus) *(IUI '26)*. Association for Computing Machinery, New York, NY, USA. doi:10.1145/3742413.3789154
[96] Qiaosi Wang, Shan Jing, and Ashok K. Goel. 2022. Co-Designing AI Agents to Support Social Connectedness Among Online Learners: Functionalities, Social Characteristics, and Ethical Challenges. In *Proceedings of the 2022 ACM Designing Interactive Systems Conference* (Virtual Event, Australia) *(DIS '22)*. Association for Computing Machinery, New York, NY, USA, 541–556. doi:10.1145/3532106.3533534
[97] Ruotong Wang, Ruijia Cheng, Denae Ford, and Thomas Zimmermann. 2024. Investigating and designing for trust in AI-powered code generation tools. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*. 1475–1493.
[98] Ruotong Wang, Xinyi Zhou, Lin Qiu, Joseph Chee Chang, Jonathan Bragg, and Amy X. Zhang. 2025. Social-RAG: Retrieving from Group Interactions to Socially Ground AI Generation. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 162, 25 pages. doi:10.1145/3706598.3713749
[99] Shaowei Wang, David Lo, and Lingxiao Jiang. 2013. An empirical study on developer interactions in StackOverflow. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing* (Coimbra, Portugal) *(SAC '13)*. Association for Computing Machinery, New York, NY, USA, 1019–1024. doi:10.1145/2480362.2480557
[100] Thomas Weber, Maximilian Brandmaier, Albrecht Schmidt, and Sven Mayer. 2024. Significant Productivity Gains through Programming with Large Language Models. *Proc. ACM Hum.-Comput. Interact.* 8, EICS, Article 256 (jun 2024), 29 pages. doi:10.1145/3661145
[101] Etienne Wenger, Richard McDermott, and William Snyder. 2002. *Cultivating Communities of Practice: A Guide to Managing Knowledge*. Harvard Business School Press, USA.
[102] Yi-Miao Yan, Chuang-Qi Chen, Yang-Bang Hu, and Xin-Dong Ye. 2025. LLM-based collaborative programming: impact on students' computational thinking and self-efficacy. *Humanities and Social Sciences Communications* 12, 1 (2025), 1–12.
[103] Stephanie Yang, Hanzhang Zhao, Yudian Xu, Karen Brennan, and Bertrand Schneider. 2024. Debugging with an AI Tutor: Investigating Novice Help-seeking Behaviors and Perceived Learning. In *Proceedings of the 2024 ACM Conference on International Computing Education Research-Volume 1*. 84–94.
[104] Yeonsun Yang, Ahyeon Shin, Mincheol Kang, Jiheon Kang, and Jean Young Song. 2024. Can We Delegate Learning to Automation?: A Comparative Study of LLM Chatbots, Search Engines, and Books. *arXiv preprint arXiv:2410.01396* (2024).
[105] Saber Zerhoudi and Michael Granitzer. 2025. SearchLab: Exploring Conversational and Traditional Search Interfaces in Information Retrieval. In *Proceedings of the 2025 ACM SIGIR Conference on Human Information Interaction and Retrieval (CHIIR '25)*. Association for Computing Machinery, New York, NY, USA, 382–389. doi:10.1145/3698204.3716475
[106] Haoxiang Zhang, Shaowei Wang, Tse-Hsun Chen, Ying Zou, and Ahmed E Hassan. 2019. An empirical study of obsolete answers on stack overflow. *IEEE Transactions on Software Engineering* 47, 4 (2019), 850–862.
[107] Wenhan Zhu, Haoxiang Zhang, Ahmed E Hassan, and Michael W Godfrey. 2022. An empirical study of question discussions on Stack Overflow. *Empirical Software Engineering* 27, 6 (2022), 148.

A Participants' Demographics

We present the participants' demographics in Table 3.

B Pilot Study

We conducted two pilot studies with two participants: a master's student in data science and a bachelor's student in computer science. We gathered feedback from them on both the system and model design.

Table 3. Participants' Demographics: We recruited 29 participants with STEM backgrounds and experience in Kaggle or similar data science competitions. One participant was excluded due to incomplete questionnaires and noticeable distraction during the study, resulting in 28 valid participants.

PID	Educational Level	Majors	Age	Gender	Experience with Online Coding Communities	Knowledge of Data Science	Knowledge of Python
1	Master	Computer Science	25	Male	yes	yes	yes
2	Master	Data Science	22	Male	yes	yes	yes
3	Ph.D	Computer Science	27	Male	yes	yes	yes
4	Master	Data Science	25	Female	yes	yes	yes
5	Master	Mechanical Engineering	25	Male	yes	yes	yes
6	Master	Data Science	24	Female	yes	yes	yes
7	Master	Quantitative Finance	23	Male	yes	yes	yes
8	Master	Mathematical Engineering	23	Female	yes	yes	yes
9	Master	Artificial Intelligence	28	Male	yes	yes	yes
10	Master	Physics	23	Male	yes	yes	yes
11	Master	Data Science	24	Female	yes	yes	yes
12	Master	Computer Science	23	Male	yes	yes	yes
13	Master	Computer Science	22	Female	yes	yes	yes
14	Ph.D	Machine Learning	25	Male	yes	yes	yes
15	Master	Electrical Engineering	26	Male	yes	yes	yes
16	Bachelor	Computer Science	19	Prefer not to say	yes	yes	yes
17	Master	Robotics, Systems and Control	24	Male	yes	yes	yes
18	Master	Computational Linguistics	25	Male	yes	yes	yes
19	Master	Computational Linguistics	23	Female	yes	yes	yes
20	Master	Computer Science	24	Male	yes	yes	yes
21	Master	Computational Linguistics	23	Male	yes	yes	yes
22	Master	Data Science	25	Female	yes	yes	yes
23	Ph.D	Computational Social Science	29	Female	yes	yes	yes
24	Bachelor	Mathematics	27	Female	yes	yes	yes
25	Ph.D	Computational Geography	25	Male	yes	yes	yes
26	Master	Computer Science	25	Female	yes	yes	yes
27	Ph.D	Robotics	26	Male	yes	yes	yes
28	Master	Data Science	25	Male	yes	yes	yes

Feedback from the first pilot study highlighted a need to enhance user engagement, as participants expressed frustration over the lack of community context and recognition for contributors. To address this, we integrated rich social design features, such as author profiles, author names, publish dates, vote counts, view counts, comment counts, and post titles, to credit authors and foster interaction.

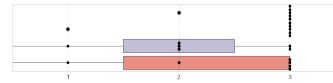
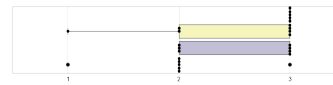
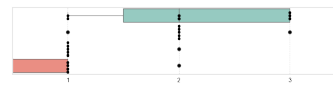
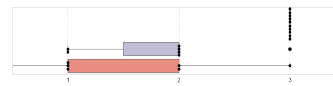
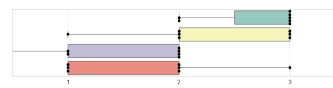
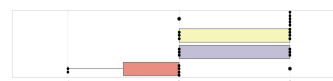
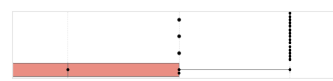
In the second pilot study, user feedback emphasized the importance of balancing relevance and popularity in post rankings. Participants highlighted the value of social design features, such as vote and view counts. In response, we introduced an advanced search panel with three ranking options—relevance, vote count, and view count—and allowed users to adjust the number of posts used to generate responses.

Additionally, participants showed a preference for asking follow-up questions, which informed our decision to design ChatCommunity's backend to retain memory of previous conversations. We also observed participants switching between different chatbot systems during testing (Alpha, Beta, and Gamma). To ensure a controlled experiment, we hid the system-switching button and instructed participants to switch systems only when prompted. Further adjustments included repositioning the "New Chat" button and refining the backend logic for fetching Kaggle community meta-information.

C Grade Distribution for Each Question

To better understand the differences in grades, we present the detailed grade distributions for the various questions in Table 4. Participants using Alpha achieves the best average grades in

Table 4. Notebook grades distribution

Question	Cond.	N	M	SD	Grade: 0 to 3
1.1 What is the meaning of each attribute in the training and the testing dataset? Are there any missing values in the training dataset?	Alpha	7	2.86	0.38	
	Beta	7	2.71	0.76	
	Gamma	7	1.86	1.07	
	Delta	7	2.14	1.21	
1.2 Is the training dataset balanced in terms of class distribution?	Alpha	7	3.00	0.00	
	Beta	7	3.00	0.00	
	Gamma	7	2.14	1.46	
	Delta	7	2.14	1.46	
2.1 If the training dataset is imbalanced, what methods can we use to handle the imbalanced classes effectively? How might these methods impact the performance of our model?	Alpha	7	3.00	0.00	
	Beta	7	2.43	0.79	
	Gamma	7	2.57	0.53	
	Delta	7	2.00	0.58	
2.2 Can we feed the training dataset directly into the model? Explain why?	Alpha	7	2.14	0.90	
	Beta	7	2.00	0.58	
	Gamma	7	1.14	0.38	
	Delta	7	0.86	0.69	
3.1 Which embedding method should we use for this competition and why is this method suitable for this competition?	Alpha	7	3.00	0.00	
	Beta	7	3.00	0.00	
	Gamma	7	1.86	0.69	
	Delta	7	1.43	0.98	
3.2 Which model should we use for this competition and why is this model suitable for this competition?	Alpha	7	2.71	0.49	
	Beta	7	2.29	0.76	
	Gamma	7	1.43	0.79	
	Delta	7	1.71	0.76	
4.1 What evaluation metrics should be used to assess the performance of the models in this competition and why?	Alpha	7	2.86	0.38	
	Beta	7	2.57	0.53	
	Gamma	7	2.14	1.07	
	Delta	7	1.86	0.69	
4.2 Suppose you are using the Random Forest model as classifier. After initial model training and evaluation, what strategies can further improve the model's performance?	Alpha	7	2.86	0.38	
	Beta	7	2.86	0.38	
	Gamma	7	2.86	0.38	
	Delta	7	1.14	1.21	

seven out of eight questions. While all participants performed similarly on data-oriented questions (e.g., Is the training dataset balanced in terms of class distribution?), their performance varied significantly on decision-making questions (e.g., Which model should we use for this competition and why is this model suitable for this competition?). Specifically, when comparing the answers from Alpha and Gamma on decision-making questions, we found that Gamma tends to provide general suggestions that may not be suitable for the current task. For example, in response to the question: "Which model should we use for this competition, and why is it suitable?", GPT-4o is likely to recommend the BERT model, a transformer-based language model, as a general solution, as reflected in one participant's answer: "*I will use BERT model to handle the problem in this competition. It's popular model, easy to finetune and could result a good perfomance in classification task*" (P4). The participant chose the BERT model recommended by GPT-4o, despite it being computationally expensive and having a large number of parameters, which makes it less ideal for this binary

classification task [46]. In contrast, Community-Enriched AI recommends a suitable model based on posts that have already demonstrated the model's effectiveness for this task, as reflected in one participant's answer:

> *We can use the simple DNN model. According to the high rated posts, DNN model is able to solve the task with less computational resources than models like transformer. If we want the higher f1-score, we can try the models like transformer or BiLSTM with attention* (P2)

This participant chose the simple DNN model which has good flexibility and is able to solve the task with less computational resources. DNN models were used in highly voted posts [26, 54] and performed well in this competition. In summary, the Community-Enriched AI design is effective in helping students achieve better performance in specific data science tasks without significantly increasing completion time.

D Grading Criteria

The grading criteria were defined by two researchers and reviewed by two data science experts. The grading criteria grade each result based on correctness, relevance, and completeness. The detailed grading criteria are shown in Table 5.

E Results of Post-Session Questionnaire

We present the results of post-session questionnaire in Figure 5.

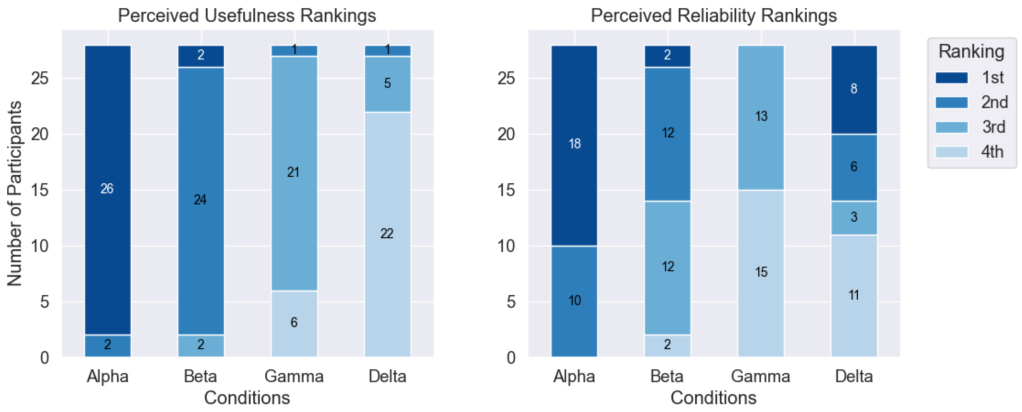


Fig. 5. Perceived usefulness and reliability ranking: 1st represents the most useful/reliable, while 4th represents the least useful/reliable. The number on each bar indicates the number of participants who gave this ranking.

F Details of Data Source and Data Processing

F.1 Details of Data Source

The Meta Kaggle and Meta Kaggle Code datasets span from 2015 to 2024, including a total of 4.83 million code files [41, 58]. Notably, 4.36 million of these files are Jupyter notebooks, which form the focus of our study. The datasets cover 5,688 competitions and include contributions from approximately 1.22 million users [58]. In addition to the code files, the dataset also contains rich social features, such as count of views, votes, comments, and other forms of engagement, providing valuable insights into community interactions.

Table 5. Grading criteria for each question

Question	Grading Criteria
1.1 What is the meaning of each attribute in the training and testing dataset? Are there any missing values in the training dataset?	**3 points:** Answer correctly defines `qid` as "unique identifier" and explains the target as sincere or insincere. And answer no missing values. **2 points:** Misses either `qid` or target definition. Others the same as above. **1 point:** Misses both `qid` and target definition but correctly identifies no missing values. **0 points:** Incorrect information on missing values or other factual errors.
1.2 Is the training dataset balanced in terms of class distribution?	**3 points:** Correctly identifies the dataset as imbalanced. **0 points:** Incorrectly identifies the dataset as balanced.
2.1 If the training dataset is imbalanced, what methods can we use to handle the imbalance? How might these methods impact model performance?	**3 points:** Identifies 2 or more methods with proper explanations. **2 points:** Identifies 1 method with explanation, or 2 methods without proper explanation. **1 point:** Identifies 1 method without explanation. **0 points:** Incomplete or incorrect answer. **Methods to consider:** Resampling (Oversampling, Undersampling), Ensemble Methods, Data Collection, Threshold Adjustment, Class-weight Adjustment, Algorithm Selection, Cost-sensitive Learning.
2.2 Can we feed the training dataset directly into the model? Explain why.	**3 points:** Provides 4 or more preprocessing steps with proper explanations. **2 points:** Provides 3 preprocessing steps with explanation. **1 point:** Provides only 1 preprocessing step or says "No" without proper explanation. **0 points:** Incomplete or incorrect answer. **Steps to consider:** Text Cleaning, Tokenization, Text Normalization, Embedding, Padding, Handling Imbalanced Classes, Feature Engineering, Train/Test Split.
3.1 Which embedding method should we use for this competition, and why is it suitable?	**3 points:** Mentions any of the following with proper explanation: Google News, FastText, GloVe, Paragram_300, Wiki-News, Word2Vec. **2 points:** Mentions BERT with a proper explanation, or mentions any of the above methods without proper explanation. **1 point:** Mentions BERT without proper explanation or other irrelevant methods. **0 points:** Incomplete or incorrect answer.
3.2 Which model should we use for this competition, and why is it suitable?	**3 points:** Identifies models like 2D-CNN, DNN, GPT-2, BART, XGBoost, or LightGBM with proper explanation. **2 points:** Mentions BERT, LSTM, or any traditional machine learning models with proper explanation. **1 point:** Mentions BERT without explanation, or other models with or without explanation. **0 points:** Incomplete or incorrect answer.
4.1 What evaluation metrics should be used to assess the performance of the model?	**3 points:** Identifies 2 or more metrics (e.g., F1 score, AUC, Confusion Matrix) with proper explanation. **2 points:** Mentions F1 score, AUC, or Confusion Matrix with proper explanation. **1 point:** Mentions 1 metric without proper explanation. **0 points:** Incomplete or incorrect answer. **Metrics to consider:** F1 Score, AUC, Precision, Recall, Confusion Matrix, Balanced Accuracy.
4.2 What strategies can further improve the performance of the Random Forest model after training and evaluation?	**3 points:** Identifies 3 or more methods with proper explanations. **2 points:** Identifies 2 methods with proper explanations. **1 point:** Identifies 1 method with or without explanation. **0 points:** Incomplete or incorrect answer. **Methods to consider:** Hyperparameter Tuning, Feature Engineering, Ensemble Methods, Regularization, Feature Importance Analysis, Data Augmentation, Threshold Tuning, Calibration.

F.2 Competition Details

Details about extracted competitions are listed in Table 6.

F.3 Details of Data Processing

In summary, we processed 37,895 code files and obtained 36,945 valid Python-written Jupyter Notebook files, containing 849,900 code cells and 269,855 markdown cells. For each valid file, we collect the following meta information: public post URL, title, vote count, view count, comment

Table 6. Selected Competitions

Competition	Description	Domain	Python Files
Quora Insincere Questions Classification	Detect toxic content to improve online conversations	Natural Language Processing	9,639
Ubiquant Market Prediction	Make predictions against future market data	Business Data Analysis	4,529
Cassava Leaf Disease Classification	Identify the type of disease present on a Cassava Leaf image	Computer Vision	12,911
Mechanisms of Action (MoA) Prediction	Can you improve the algorithm that classifies drugs based on their biological activity?	Medical Data Analysis	9,866

count, notebook submission date, author name, and the notebook author's public profile avatar. Our data collection adheres strictly to Kaggle's policies, ensuring that all utilized information is open source and publicly accessible.

G RAG Model

G.1 The Retriever Module

G.1.1 Technical Details. We begin by constructing comprehensive Kaggle post databases, creating a separate database for each Kaggle competition. Each notebook submission is divided into smaller chunks, consisting of a continuous group of markdown cells followed by corresponding code cells, up until the next markdown cell. This approach balances prompt length and retrieval efficiency—chunking the entire notebook risks exceeding the input length limits of LLMs and increasing generation time, while using individual code cells would miss the connection between markdown explanations and code. By chunking markdown and code together, users can retrieve relevant information using either text explanations or code. On average, each chunk in our processed dataset contains 1.2 markdown cells and 3.6 code cells.

To encode these chunks into vector embeddings, we use OpenAI's `text-embedding-ada-002` [66] model. The embedding process for a given chunk c_i is formally defined as[4]:

$$\mathbf{e}_{c_i} = \texttt{LinearProjection}\left(\texttt{Pool}\left(\texttt{TransformerLayers}\left(\texttt{TokenEmbedding}\left(\texttt{Tokenize}(c_i)\right)\right)\right)\right)$$

where: - $\texttt{Tokenize}(c_i)$ converts the chunk c_i into a sequence of tokens, - $\texttt{TokenEmbedding}(\cdot)$ maps each token to an initial dense vector representation, - $\texttt{TransformerLayers}(\cdot)$ applies multiple transformer layers to capture contextual information across tokens, - $\texttt{Pool}(\cdot)$ aggregates the sequence of token embeddings into a single vector using pooling, - $\texttt{LinearProjection}(\cdot)$ maps the pooled vector to the final embedding space.

The resulting embedding $\mathbf{e}_{c_i}$ captures the semantic content of the entire chunk, integrating information from both markdown text and associated code blocks. We use the same embedding for both markdown and code to ensure uniform retrieval, enabling the model to effectively retrieve relevant information regardless of the input text type. These embeddings are then stored in a vector database, along with their corresponding chunk indices. For this study, we utilize ChromaDB [1] to manage and query these embeddings efficiently.

When a user inputs a query q, we calculate the Maximal Marginal Relevance (MMR) score using the vector representation of the query $\mathbf{q}$ and the embeddings of each chunk $\mathbf{e}_{c_i}$ to identify the top 10 most relevant chunks. This process can be formally writen as:

[4]The exact implementation of the text-embedding-ada-002 model may vary, and components like pooling or linear projection may differ or be omitted.

$$\text{MMR}(c_i, \mathbf{q}) = \arg\max_{c_i \in C} \left[\text{Relevance}(\mathbf{q}, \mathbf{e}_{c_i}) - \lambda \max_{c_j \in S} \text{Sim}(\mathbf{e}_{c_i}, \mathbf{e}_{c_j}) \right]$$

where C is the set of all chunks, S is the set of already selected chunks, $\text{Relevance}(\mathbf{q}, \mathbf{e}_{c_i})$ represents the similarity between the query vector $\mathbf{q}$ and chunk embedding $\mathbf{e}_{c_i}$, and λ is a parameter controlling the diversity of the selected chunks. The choice of the top 10 is consistent with established practices in information retrieval [42].

These top 10 chunks are then ranked according to the user's preference using one of the following methods:

(1) **Relevance (MMR Score)**: The chunks are ranked by their $\text{MMR}(c_i, \mathbf{q})$ scores, which can be expressed as:

$$\text{Rank}_{\text{MMR}}(c_i) = \text{Sort}\left(\text{MMR}(c_i, \mathbf{q}), \text{descending}\right)$$

(2) **View Count**: The chunks are ranked according to the view count V_{c_i} of the corresponding post:

$$\text{Rank}_{\text{View}}(c_i) = \text{Sort}(V_{c_i}, \text{descending})$$

(3) **Vote Count**: The chunks are ranked according to the vote count R_{c_i} of the corresponding post:

$$\text{Rank}_{\text{Vote}}(c_i) = \text{Sort}(R_{c_i}, \text{descending})$$

Based on the user's selection of the ranking method, top N chunks are selected from the sorted list:

$$\{c_i\}_{\text{top } N} = \text{Rank}_{\text{Method}}(c_i)[1 : N]$$

where $\text{Rank}_{\text{Method}}(c_i)$ denotes the list of chunks sorted by the selected ranking method (relevance, view count, or vote count). These top N chunks are then presented to the user along with links to their respective Kaggle notebook posts.

G.2 The Generator Module

The retrieved posts, along with the user's query, are formulated into a comprehensive prompt P:

$$P = \text{Format}(\mathbf{q}, \{c_i\}_{\text{top } N})$$

where $\mathbf{q}$ is the user's query, and $\{c_i\}_{\text{top } N}$ represents the top N retrieved chunks selected in the retriever module. The $\text{Format}(\cdot)$ function constructs a structured prompt by integrating the user's query $\mathbf{q}$ with the content of the top N chunks.

The prompt P is then fed into the GPT-4o model:

$$\mathbf{r} = \text{GPT-4o}(P)$$

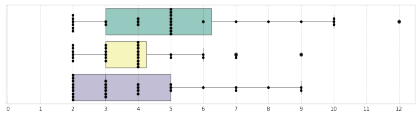
where $\mathbf{r}$ is the generated response. The GPT-4o model synthesizes the information from the prompt P to produce a coherent and tailored response $\mathbf{r}$ that addresses the user's query.

We use the following prompt for generating response:

```
Task: You are an expert in data science. Your goal is to assist a user in dealing
with a task related to a Kaggle competition: {title of competition}. Whenever
possible, provide answers in the form of code snippets that directly address the
user's needs.
```

```
Information Provided:
 (1) Competition Description: {One sentence description of the competition}
 (2) Context: {context}
 *Note:* The context includes other people's code, which contains information necessary for answering the user's question. Please rely solely on the provided context to craft your response. Assume all questions pertain specifically to this competition. If the context is empty, state "There is no relevant information in previous notebooks" and then proceed to answer the question based on your expertise.
User Query: {question}
Expected Output: Please provide your response as a string, including code snippets where applicable.
```

Table 7. User behavior statistics.

Attribute	Cond.	N	Mean	SD	
Input Prompts	Alpha	28	5.25	2.84	
	Beta	28	3.96	1.75	
	Gamma	28	4.04	2.22	
Times Clicking Preview	Alpha	28	2.75	2.55	
Times Using Advanced Search Panel	Alpha	28	1.71	1.82	

H Detailed Analysis of User Interactions

We present user behavior statistics across different conditions in Table 7.

Received May 2025; revised November 2025; accepted December 2025