Adaptive Course Enhancement through Iterative Reflection-Based Intervention Design

Ms. Sandra Monika Wiktor, University of North Carolina at Charlotte

Sandra Wiktor is a Ph.D. candidate in Computer Science at the University of North Carolina at Charlotte. She specializes in applied machine learning with a focus on generative AI, aiming to improve computer science education by leveraging the capabilities of large language models. Her research centers on designing adaptive educational interventions by analyzing real-world student data, such as written reflections and learning behaviors. Sandra has published in FIE, ICER, and ASEE, and brings years of teaching experience in software engineering and foundational computing courses.

Dr. Mohsen M Dorodchi, University of North Carolina at Charlotte

Dr. Dorodchi has been teaching in the field of computing for over 35 years of which 25 years as an educator. He has taught the majority of the courses in the computer science and engineering curriculum over the past 25 years such as introductory programming, data structures, databases, software engineering, system programming, etc. He has been involved in a number of National Science Foundation supported grant projects including Scholarship for STEM students (S-STEM), Researcher Practitioner Partnership (RPP), IUSE, and EAGER.

Adaptive Course Enhancement through Iterative Reflection-Based Intervention Design

Introduction

Student feedback can shape teaching practices and improve course design [1, 2]. Although many institutions collect student feedback, the pedagogical value of this practice depends on instructors' engagement with the responses [3, 4]. By listening to student feedback, instructors can tailor their courses to students' needs, fostering a dynamic and adaptable learning environment. The timing and structure of the feedback collection also influence its impact. Mid-semester feedback, for example, allows instructors to adjust courses in real-time, which benefits current and future students [5]. Furthermore, free-written feedback can unveil more detailed and valuable suggestions than quantitative survey items [6], but analyzing such feedback can be difficult [7]. Reflective prompts, in particular, can encourage students to think about their learning experience [8] and can "[uncover] misunderstandings and [reveal] personal learning experiences" [8]. Researchers have developed various text analysis methods to help instructors interpret unstructured student feedback [2, 3, 6]. However, current works lack a practical framework for solving the challenges.

Moving beyond interpretation alone, we introduce a process for designing course interventions directly informed by student feedback. We define course interventions as evidence-based actions to improve student educational outcomes. This approach, the reflection-based course intervention development cycle (CIDC), guides the design of feedback-driven interventions. To analyze reflections comprehensively, we develop a taxonomy of student challenges in software engineering (SCT-SE) and use it to classify and quantify common issues. Furthermore, as manual feedback analysis can be laborious, we present large language model (LLM) prompting methods designed to automate this process. LLMs are trained on a wide breadth of data, granting them significant flexibility to handle various text-processing tasks [9]. Therefore, we introduce a structured prompting approach based on the SCT-SE to build highly contextual prompts. This approach allows us to tailor the model's creative generation capabilities to our task.

We apply the CIDC to student reflections, designed initially as part of an experiential learning model [10]. This process encompasses four steps. (I) First, we periodically collect reflections from students during a semester. (II) Next, we identify, classify, and aggregate the frequency of common challenges in each reflection. The SCT-SE, which includes topic labels, facilitates the process of topic identification and classification. We prompt the LLM to classify new reflections into these topic labels, which we then use to generate frequency statistics. (III) We develop interventions for the most common proficiency challenges using these frequency statistics. These interventions comprise student support material (SSM) — an aggregation of supplementary materials such as videos, short articles, activities, and additional practice problems. The resources target students' most common weak areas. (IV) Finally, we evaluate the impact of these interventions.

We conducted this study under IRBIS-21-041. Reflections are anonymized, and we never share personally identifiable student information with any AI tool or large language model.

Literature Review

Although feedback is traditionally collected through end-of-semester student evaluations of teaching (SETs) [11], some utilize mid-semester reviews [5] or ongoing, periodic feedback [2]. Regardless of the timing, researchers have consistently found that student feedback can improve course design and enhance teaching effectiveness [1]. Beyond its impact on individual courses, feedback also serves as a valuable tool for maintaining quality assurance in education systems [12], particularly when paired with professional development initiatives [2]. Moreover, instructors generally respond positively to student feedback and report that they would not introduce "unjustified changes" to their courses based on such input [1].

Instructors leverage various styles of feedback prompting, ranging from Likert-scale to qualitative questions. Among these, free-response feedback often yields more detailed insights than quantitative survey items [6]. Students frequently raise unique points in open-ended feedback that closed-ended questions cannot capture. These comments identify a broader range of negative and positive course-related issues, providing deeper, student-centered, context-specific insights that help improve teaching outcomes [7, 13]. Free-response feedback can also unveil difficulties students experience during the course [14].

Moreover, the style of feedback itself can significantly shape the student experience. For instance, reflective writing can reveal "personal learning experiences" [8]. Research finds that reflective journaling improves content comprehension and promotes self-analysis, encourages self-efficacy, fosters student engagement (especially when faculty respond to comments), and strengthens career skills [4].

While collecting student feedback is valuable, instructors must organize it meaningfully to "make sense of [it]" [7]. Furthermore, if students do not see any changes during the semester, they become "very cynical" of the process [3]. Another study applied a *feedback action cycle* defined by the three steps of (1) gathering feedback, (2) reflecting on feedback, and (3) acting on feedback [2]. Other approaches use the One-Minute Paper (OMP), where students briefly state what they learned and pose an unanswered question at the end of class. The instructor then analyzes the answers and addresses the points students made in the reflections during class. This process can foster one-on-one dialogue, encourage student reflection, and support active learning [15]. Other work has used insights from student reflections to foster a sense of belonging [16].

Researchers have applied machine learning (ML) and natural language processing (NLP) techniques—such as clustering [14] and topic modeling (e.g., LDA) [17]—to automate feedback analysis. These works automatically analyze student feedback from different perspectives, producing a range of insights [6, 18]. Large language models (LLMs) have emerged as general-purpose, adaptive, and versatile alternatives to traditional ML techniques [9]. They demonstrate comprehensive reasoning abilities [19] and can perform a wide range of tasks [20]. As a result, researchers are exploring their educational applications, such as providing tutoring services or generating feedback for students [21, 22, 23]. With natural-language prompting, LLMs can perform various tasks without explicit training [9, 24], enabling them to extract topics from student feedback datasets through prompting alone [25].

While prior research highlights the value of student feedback, few approaches offer a structured, practical framework for translating that feedback into course-level interventions. In this work, we address that gap.

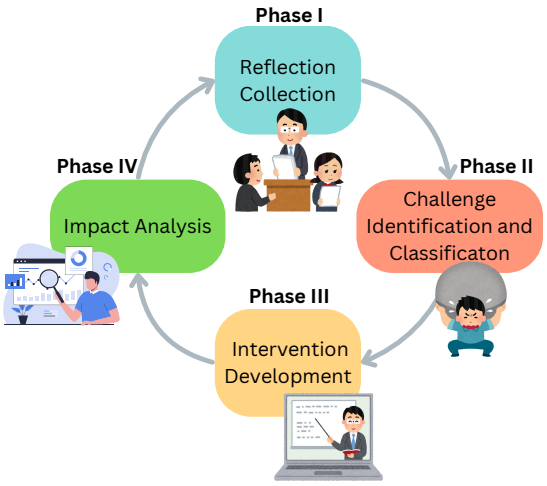

Figure 1: An overview of the reflection-based course intervention development cycle (CIDC).

Method Overview

We present a method for developing interventions that address the challenges students express in short-form reflections collected periodically throughout the semester. A reflection period denotes the interval between the assignment of the reflections and their submission deadline. For clarity, we number each round sequentially (e.g., Reflection 1, Reflection 2). This method, the *reflection-based course intervention development cycle* (CIDC), includes four phases shown in Figure 1. First, we collect reflections from students during each reflection period. We then develop a model to extract and classify the reflections into frequent topics and quantify their occurrence. Based on this analysis, we develop interventions, referred to as *student support material* (SSMs), and finally, we measure the impact of these interventions. The SSMs provide learning resources for areas where students express difficulty in their reflections. With each new reflection period or semester, we refine courses based on student challenges.

Leveraging insights from prior semesters enables us to continually refine course design toward a more effective, student-centered learning experience. In the following section, we detail each phase of the intervention development cycle alongside our implementation of the method in a software engineering course. We apply the CIDC using data from three consecutive semesters of software engineering (SE) courses (SE1, SE2, and SE3). SE3 includes two classes delivered simultaneously, which we refer to as SE3A and SE3B. Because our questions are reflective in style, we use the term *reflections* to denote student feedback throughout this work, although the method applies to various feedback types.

Phase I: Reflection Collection

In *Phase I: Reflection Collection*, we first collect reflections from students. We utilize two sets of reflection questions: emotion-challenge reflection questions (EC-RQ), containing emotion-extracting questions (EEQ) and challenge-centered questions (CCQs); and SSM reflection questions (SSM-RQ). The EC-RQs contain the following questions, alongside others designed to gauge their experience in the course: *(EEQ1)*: "How do you feel about the course so far?"; *(EEQ2)*: "Ex-

plain why you selected the above choice(s)."; *(CCQ1)*: "What was your biggest challenge(s) for these past modules?"; *(CCQ2)*: "How did you overcome this challenge(s)? Or what steps did you start taking towards overcoming it?"; *(CCQ3)*: "Do you have any current challenges in the course? If so, what are they?" We prompt students to respond to EC-RQs at regular intervals (4-5 times) during the semester, allowing us to track their evolving challenges and perceptions of the course. The challenge questions are based on similar works related to reflections [26, 14, 16, 27]. The last reflection contains questions about students' experiences with the interventions. Each EC-RQ is a free-response question, except for EEQ1, which is a multi-select, multiple-choice question. We selected the options for EEQ1 (excited, satisfied, neutral, confused, frustrated, anxious) based on research in academic emotions [28, 29]. We refer to reflections answering EC-RQs as EC-reflections and represent these reflections as $\mathbf{R} = \langle \mathbf{R}_1^i, ..., \mathbf{R}_n^i \rangle$ where *n* is the number of reflection sets collected and *i*=the semester from which we collected the reflections. For each SSM, the instructor asks students to answer a set of related SSM-RQs as a reflection activity. The SSM-RQs accomplish three goals: (SSM-RQ-G1) track students' completion of the material; (SSM-RQ-G2) obtain feedback about the usefulness of the material; and (SSM-RQ-G3) encourage the students to apply the concepts in the material to their learning. This reflection process allows instructors to monitor student development and adapt the course accordingly. By approaching student feedback from multiple angles, instructors can gain a comprehensive understanding of student experiences. Rather than relying on a single end-of-semester evaluation form, as standard in many universities, instructors can assist students in real-time, responding to challenges as they occur.

Phase II Part 1: Challenge Identification

In *Phase II: Challenge Identification and Classification,* to develop an LLM prompting model with a strong contextual background, we first perform manual challenge identification using five labelers based on reflections from SE1 and SE2, represented as $\mathbf{R}^{SE1}$ and $\mathbf{R}^{SE2}$ respectively. This manual labeling session took place during the delivery of the SE2 course. First, labelers labeled $\mathbf{R}_1^{SE1}$ and $\mathbf{R}_1^{SE2}$ and took unstructured notes on the most common challenges they found. A lead labeler (LL) reviewed the notes and produced an initial list of labels. All labelers then validate the labels via discussion. The labelers used the initial list of labels to label (classify) the following reflections, adding new labels as necessary and validating them within the group. Once the labeling process concluded, the LL proposed a final list of labels and coinciding descriptions, represented as $\langle y, d \rangle$. We represent the challenge labels as y=$\langle y^1, ..., y^n \rangle$, where y^n represents the *n*th topic label, as well as coinciding descriptions d=$\langle d^1, ..., d^n \rangle$ that explain each label.

Based on these results, we have developed a challenge classification system called the student challenge taxonomy (SCT). This system comprises two types of challenges: primary and secondary. *Primary labels* are challenges related to specific key concepts in the course, including tools, technologies, and subjects from the course curriculum. Within the primary labels, we include subcategories of *proficiency*, *structural*, and *other*. Proficiency issues relate to skills and knowledge students need to fulfill course objectives. Structural issues involve the design of the course or course materials. Within the secondary challenges category are the *grading* and *self* categories. The grading subcategory refers to specific types of syllabus categories of grading, and the *self* refers to challenges arising from personal perception or experience. Both primary and secondary labels include an *other* and *none* category, capturing miscellaneous topics and cases with

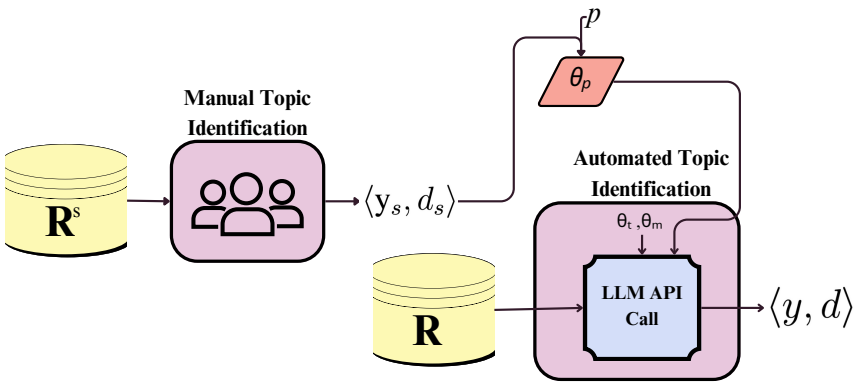

Figure 2: A flowchart demonstrating the few-shot identification of new challenges.

no such challenge. This paper provides a group-scoped intervention for proficiency issues. However, instructors can use other categories to develop other interventions; such cases are the subject of future work. Table 1 presents the software engineering SCT (SCT-SE).

We present a few-shot LLM-prompting method (i.e., we provide examples to the model to illustrate our intended output [30]) to generate contextualized topic labels, presented in Figure 2. $\mathbf{R}^s$ is the reflection dataset human labelers use to identify initial topics, and $\langle y_s, d_s \rangle$ represents the example labels. Our case study uses labels and descriptions from SCT-SE for $\langle y_s, d_s \rangle$. θ_p combines the examples with structured instructions (*p)*. We input a new reflection set $\mathbf{R}$ into the system to produce new labels. Alongside $\mathbf{R}$, we pass the following hyperparameters into the API call for the LLM: θ_p, θ_t, the *temperature* which controls the creativity of the model's generations, and θ_m, which determines the LLM variant used, in this case, OpenAI's *gpt-4o*. To illustrate this process, we provide an initial example by asking the LLM to produce new labels for the software engineering course based on previous labels. In this example, we compare the labels identified by human labelers to the model's label generations. We input labels from $\mathbf{R}_{1-2}^{SE1-2}$ to produce labels for the $\mathbf{R}_3^{SE1}$ dataset. The human-identified labels identified in $\mathbf{R}_3^{SE1}$ are *IDE Package and Software Installation* and *API*. Following the structure defined in Figure 2, we provide the definition for *p*, the *prompt text*, in Table 2.

As seen in Table 3, the labels and descriptions generated, *APIs* and *Virtual Environments*, overlap with those produced by the manual approach. The model captures the emergence of APIs as an issue directly, and its description overlaps with the manual approach's description in more depth. Furthermore, while the model used the term *virtual environment* rather than *IDE Software and Package Installation*, the installation issues occur within the virtual environments, and the IDE uses the virtual environment. When examining the label's description, it overlaps with the manual description, rendering it functionally equivalent. We can adapt this method to generate labels for future reflections, semesters, and courses; automating these tasks makes complex feedback analysis feasible.

Phase II Part 2: Challenge Classification

After identifying the y labels for each reflection in the $\mathbf{R}$ set, we calculate the frequency of each topic label to identify the most common challenges students face. Five labelers created this initial

Type	Category	Name	Description
Primary	Proficiency	Python and Coding [$\mathbf{R}_1^{SE2}$]	Students' challenge centers around learning and coding in Python. Non-specified coding challenges fall into this category as well. This category often includes learning Python for the first time or transitioning from another language to Python.
Primary	Proficiency	GitHub [$\mathbf{R}_2^{SE2}$]	Student describes a challenge with using GitHub. This can include difficulty with using GitHub in an assignment.
Primary	Proficiency	MySQL [$\mathbf{R}_2^{SE1}$]	Student describes a difficulty with using MySQL, the database management system.
Primary	Proficiency	API [$\mathbf{R}_3^{SE1}$]	Student describes difficulty with using or understanding APIs, especially in the context of FastAPI.
Primary	Proficiency	HTML [$\mathbf{R}_3^{SE2}$]	Students describe difficulty working with, coding, or learning HTML.
Primary	Proficiency	IDE Package and Software Installation [$\mathbf{R}_3^{SE1}$]	Students describe difficulty installing or working with packages, software, or tools within their IDE or other IDE problems. They may reference downloading or installing packages with PIP, using the terminal, and so forth. Other IDE problems may include issues with the interpreter or virtual environment.
Primary	Structural	Course Structure and Materials [$\mathbf{R}_1^{SE2}$]	Student mentions difficulty with some aspect of the course design (such as the lecture, the style of the assigned work, mode of delivery in the course, etc) or the materials provided for studying (such as resources, slides, etc). This may include suggestions for improving the design of the course or criticisms about some aspect.
Primary	Structural	Understanding Requirements and Instructions [$\mathbf{R}_1^{SE2}$]	Student describes feeling confused or unsure how to follow the instructions relating to assigned work in the course or requirements to complete any assigned work (activities, assignments, quizzes etc). This could include instructions on assigned work or submission instructions.
Primary	Proficiency	Time Management and Motivation [$\mathbf{R}_1^{SE2}$]	Student has difficulty managing their time and balancing their workload. This can include procrastination and falling behind. This also includes issues with motivation or focus.
Primary	Structural	Group Work [$\mathbf{R}_3^{SE2}$]	Students describe difficulty with collaborating with their peers in group work. This can include issues of poor communication or lack of participation from their teammates, among other issues.
Primary	Other	Other	Student describes a challenge that does not fit into the categories above, but falls under the scope of 'primary_label.'
Primary	Other	None	The student does NOT specify ANY challenges that would fall under the definition of Primary Labels. Only use this label if there are NO challenges mentioned that fall under the Primary Label. If so, this should be the only label. Sometimes, students will say they have no challenges, but do provide some challenges. These students do not fall under the category of none.
Secondary	Grading	Assignments [$\mathbf{R}_1^{SE2}$]	Student challenge discusses issues with the course assignments.
Secondary	Grading	Quizzes [$\mathbf{R}_1^{SE2}$]	The challenge references course quizzes.
Secondary	Grading	Projects [$\mathbf{R}_4^{SE2}$]	The challenge relates to the course project.
Secondary	Grading	Exam [$\mathbf{R}_2^{SE1}$]	Student mentions difficulty with an exam, test, or any other similar formal evaluation of performance.
Secondary	Proficiency	Learning New Material [$\mathbf{R}_1^{SE2}$]	The student describes challenges related to learning new concepts taught in the course. This may include difficulties understanding or memorizing specific concepts, or broader challenges with the process of learning new material.
Secondary	Self	Personal Issue [$\mathbf{R}_2^{SE2}$]	The challenge involves a personal issue that is not directly caused by the course, but arises from the student's unique situation or perception.This may include difficulties aligning personal preferences with the course content or the style of course administration, as well as challenges stemming from outside life events.
Secondary	Self	Course Future [$\mathbf{R}_1^{SE2}$]	The challenge involves apprehension or concern about future elements of the course, rather than a current situation.
Secondary	Other	Other	The challenge does not fit into the pre-defined categories for the Secondary Label definitions but is still considered a Secondary Label challenge.
Secondary	Other	None	No challenges that fall under the Secondary Labels definition are evident in the reflection.

Table 1: The student challenge taxonomy for the software engineering course (SCT-SE). The *Name* column includes the label name alongside the semester of origin.

Section	Heading	Subprompt
$p_{context}$	Context	You are given a list of student reflections from a course that teaches the principles of software development. You are also given a list of common challenge topics and coinciding descriptions that have been extracted from previous student reflections.
p_{ecrq}	Reflection Questions	{*A list of EC-RQs is provided*}
p_{task}	Task	Your goal is to identify the most common topics and descriptions from a new set of student reflections.
$p_{y,d}$	Prior Labels and Descriptions	$\{\mathbf{R}_{1-2}^{SE1-2}\}$
$p_{instruction}$	{*None*}	Create new labels if students repeatedly mention a new course topic or technology. If a new topic is frequently mentioned, then create a new label, rather than classifying it into "Learning New Material."

p_{output}	Output Format	Your output format must in json. Your new labels should be in the same style as the examples. You may add one or more new labels only if many students reflections fall in this label. Only show new labels.
		{
		topic: Insert the name of the topic here,
		justification: Provide a thorough explanation of why you believe this topic should be added,
		example: Provide one reflection from the set that falls in this category. Include the entire reflection text.
		}

Table 2: The prompt used to identify new topics based on labels manually identified in $\mathbf{R}_{1-2}^{SE1-2}$ for the $\mathbf{R}_{3}^{SE1}$ dataset. Represents the value of *p*.

Topic	Description	Justification
APIs	Students' challenge centers around understanding and working with APIs. This includes learning how to implement, integrate, and troubleshoot API-related tasks.	Multiple students mentioned having difficulties with APIs, including understanding their concepts and practical implementation. This suggests that APIs are a significant challenge area that warrants its own category.
Virtual Environments	Students' challenge involves setting up and working within virtual environments. This includes issues with configuration, management, and troubleshooting within these environments.	Several students indicated that they had trouble with virtual environments, which can be crucial for development work. This suggests that this is a recurring issue that should be documented separately.

Table 3: New topics discovered for $\mathbf{R}_{3}^{SE1}$ by the automated approach to label identification.

Issues	L1	L2	L3	L4	L5	$\bar{x}$	%	σ
GitHub	40	40	35	41	38	38.8	44.09%	2.39
Python and Coding	19	20	16	19	11	17	19.32%	3.67
None	14	13	13	12	12	12.8	14.55%	0.84
Assignments	10	13	8	15	10	11.2	12.73%	2.77
Time Management and Motivation	5	6	3	7	4	5	5.68%	1.58
Learning New Material	4	7	1	8	5	5	5.68%	2.74

Table 4: Frequency of topics identified in $\mathbf{R}_{2}^{SE1}$. n=88 where n is the number of submitted reflections. We omitted entries with a $\bar{x} < 5$ for brevity.

Topic	L1	L2	L3	L4	L5	$\bar{x}$	%	σ
Github	30	31	25	32	27	29	36.25%	2.92
Python and Coding	15	14	18	14	11	14.4	18.00%	2.51
Time Management and Motivation	14	17	6	19	6	12.4	15.50%	6.11
Quizzes	8	8	7	7	9	7.8	9.87%	0.84
None	7	7	8	8	7	7.4	9.37%	0.55
Learning New Material	7	10	3	9	5	6.8	8.50%	2.86
Assignments	6	4	7	5	6	5.6	7.00%	1.14

Table 5: Frequency of topics identified in $\mathbf{R}_2^{SE2}$. n=79 where n is the number of submitted reflections. We omitted entries with a $\bar{x} < 5$ for brevity.

Prompt Header	Label
Task	Annotate each reflection with topic labels based on the challenges a student identifies in their reflections. This is a multi-label classification task where you may use more than one label. You will also be asked to explain your selections and indicate whether each challenge has been resolved. This task answers the question 'What challenges do students discuss in reflections?'
Context	This is a reflection taken from an undergraduate computer science course where students are taught the fundamentals of software engineering.
Labels and Descriptions	-
a. Label Types	-
i. Category Label	$p_{category}$
ii. Resolution Label	$p_{resolution}$
b. Challenge Category Labels Defined	$\langle y_{category}, d_{category} \rangle$
c. Resolution Labels Defined	$\langle y_{resolution}, d_{resolution} \rangle$
Output Instructions	p_{output}

Table 6: The structure for the text classification prompt. The *Challenge Category Label* is either primary or secondary, and the parameters will hold different values depending on which version of the prompt is used.

dataset by annotating reflections from SE1 and SE2 during the SE2 course. We provide the results of the labeling in Table 4, demonstrating the average number of labels for each category. During the SE2 semester, we administered SSM interventions after $\mathbf{R}_2^{SE2}$ at the midway point between the beginning of the course and the midterm, using $\langle y, d \rangle$ labels from SCT-SE with labels from $\mathbf{R}_{1-2}^{SE1-2}$. We then develop another set of interventions after $\mathbf{R}_3^{SE2}$, delivered near the end of the semester.

We provide a structured prompt, depicted in Table 6, to the LLM to cleanly separate different components of the task description. The prompt comprises several headings with instructions appended beneath each heading. We perform primary and secondary label identification separately, but use the same prompt structure for both. We provide an overall task description (p_{task}) alongside course context ($p_{context}$). The $p_{category}$ label describes either the primary or secondary label (i.e., $p_{category}$ = $p_{primary}$ or $p_{secondary}$). $\langle y_{category}, d_{category} \rangle$ provides the category (primary or secondary) label and the description, as provided in Table 1. To direct the model to present the output in the proper JSON format needed for further analysis, we append output instructions (p_{output}) to the prompt. To enhance instructors' understanding of student difficulties, we prompt the model to determine the *resolution* status of each reflection, which indicates whether the student resolved their reported challenge. The resolution status also helps identify *whether* a reflection describes a challenge, as some students provide positive feedback or report that they are doing well. By understanding whether the challenge a student described in their reflections needs attention, the instructor can prioritize those who are currently struggling. We describe the task as $p_{resolution}$ = *"Resolution Labels include the labels 'resolved,' 'unresolved,' 'no_challenge,' and 'unspecified.' In*

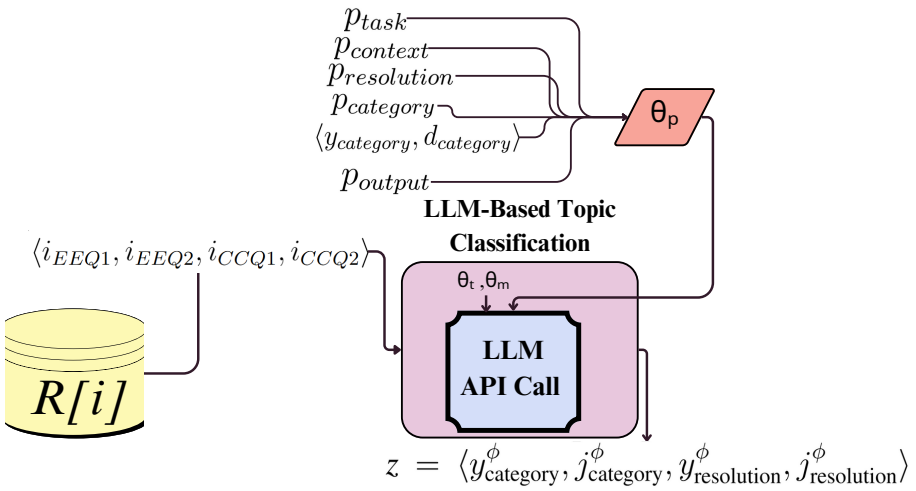

Figure 3: LLM-based topic classification flowchart.

the reflection questions, students often say or imply whether or not their challenge was resolved. Use the context from the rest of the reflection to ascertain this." Each label $y_{resolution}$ and coinciding descriptions $d_{resolution}$ is as follows: (1) *Resolved:* The student mentions a challenge but states or implies it has been resolved; (2) *Unresolved:* The student mentions a challenge that remains unresolved; (3) *no_challenge*: No challenges are mentioned in the reflection; and (4) *unspecified*: It is unclear whether the challenge has been resolved or not.

We instruct the model to justify each label selection, as this improves LLM performance while providing instructors with clear rationales to support the label decisions, which may demystify the "black-box" label generation. The input of the system is one reflection **R[*i*]** where i represents the *i*th reflection with answers to all five of the reflection prompts. Therefore, **R[*i*]** $= \langle i_{\text{EEQ1}}, i_{\text{EEQ2}}, i_{\text{CCQ1}}, i_{\text{CCQ2}} \rangle$. The output of the system $z = \langle y^{\phi}_{\text{category}}, j^{\phi}_{\text{category}}, y^{\phi}_{\text{resolution}}, j^{\phi}_{\text{resolution}} \rangle$, where ϕ represents one of either challenge category (primary or secondary) or resolution category labels, and *j* represents the free-written *justification* for each label. Figure 3 illustrates the system. Table 7 presents example outputs. The label justifications make the LLM's decision-making process transparent by highlighting the points within the reflection that led to the final choice. In the first example, the model isolates team communication as the biggest challenge and classifies this statement under *group work*. The model uses context from CCQ2 and CCQ3 to identify that the student resolved their reported challenge, as they created a group chat to manage team communication more efficiently and stated that they currently have no challenge.

While providing instructors with statistical analysis of reflections may deepen their general understanding of these challenges, the presentation and organization of this insight must be effective. By organizing this information into bar charts and adjustable tables, instructors can view this information in a digestible format. During SE3, we provided instructors a basic interface to visualize and interact with the data. Figure 4 exemplifies the frequency chart provided to instructors, which includes both the topic labels and resolution status for each topic. Combining both types of data depicts student challenges more accurately, isolating the groups that may require immediate attention. For example, API is the category of challenges with the highest frequency (n = 20) but contains only four (20%) unresolved reflections. Although the label *time management and motivation* contains only 13 challenges, a higher proportion (46.2%) are unresolved. The instructors of

Reflection	Primary Label	Primary Label Explained	Resolution Label	Resolution Label Explained
EEQ1 (Sentiment): Satisfied **EEQ2 (Why do you feel that way?):** i like my final project group **CCQ1: (Biggest Challenge?):** communication within the team but we've made it work **CCQ2 (Solution?):** making a group chat **CCQ3 (Current Challenge?):** not really, just working on finishing up the course	group_work	The student identified communication within the team as their biggest challenge, which falls under the 'group_work' category. This indicates difficulty collaborating effectively with peers on their final project.	resolved	The student mentioned they overcame the communication challenge by creating a group chat, which implies that the issue has been resolved. Additionally, when asked about current challenges, they state ' not really,' suggesting that previous issues have been addressed.

Table 7: Example of output from the LLM-prompting based automatic challenge classification model.

SE3 used the unresolved reflections to determine priority.

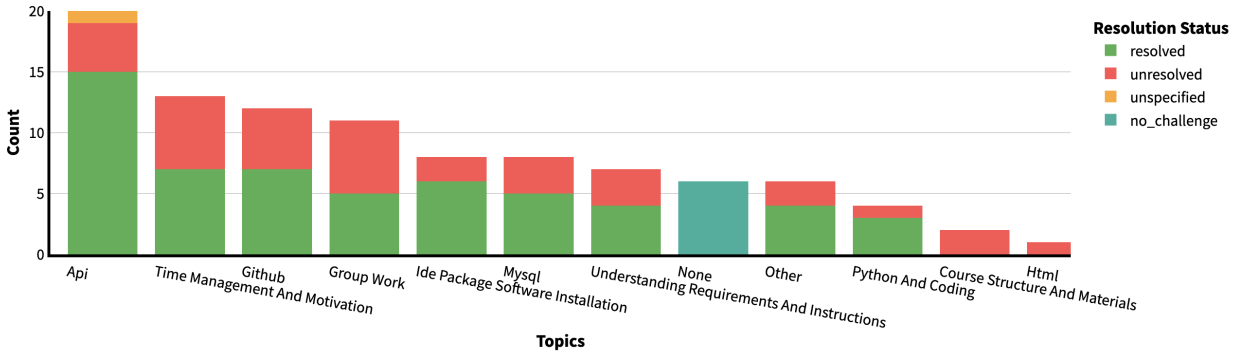

Figure 4: The primary label classification performed by the LLM prompting method for $\mathbf{R}_3^{SE3A}$

Identifying common challenges and classifying reflections into challenge labels is complex, as reflections may be sparse, incomplete, low-quality, or vague. They also often contain multiple topics that labelers must interpret, further complicating the task. Furthermore, classification is time-consuming to perform manually and is, therefore, infeasible as a regular course analysis method. This work aims to simplify this process to support effective communication between instructors and students by providing a reliable automation method. The proposed LLM-prompting method is a step towards facilitating this communication. In future work, we aim to develop these methods further and perform rigorous evaluations to measure their performance against human labelers for consistency and accuracy.

Phase III: Intervention Development

During *Phase III: Intervention Development*, based on each y, we identify a subset of labels ($y\prime$) for which we develop new learning resources, material or lessons to address each challenge in a contained module. We refer to the content created for each $y\prime$ as student support material (SSM), where each $y\prime$ has a coinciding SSM, developed developed during SE2. As proficiency challenges address specific skills students gain to accomplish course objectives, we determine that additional practice or supplementary instruction can address these challenges, so $y\prime$ contains specific proficiency challenge labels. Future work will target other issues as well. We present labels where

Issue	L1	L2	L3	L4	$\hat{x}$	%	σ
Assignments	8	14	13	19	13.5	17.76%	3.91
API	11	14	16	11	11.5	15.13%	2.22
Time Management and Motivation	9	18	11	7	11.25	14.80%	4.15
Group Work	6	14	13	4	9.25	12.17%	4.32
MySQL	8	10	9	6	8.25	10.86%	1.48
None	8	6	6	9	7.25	9.54%	1.30
Github	3	10	7	4	6	7.89%	2.74

Table 8: Results of labeling R_3^{SE2}.

$\bar{x} > 5$ in Table 8. The most common proficiency issues in $\mathbf{R}_2^{SE2}$ are *GitHub* (-7.84% change from $\mathbf{R}_2^{SE1}$) and *Python and coding* (-1.318%), followed by *time management and motivation* (+9.82%) and *learning new material* (+3.1%), and this subset becomes our $y\prime$. These are the *initial* SSMs developed based on $\mathbf{R}_2^{SE2}$, referred to as $\mathbf{R}_2^{SE2}$-SSMs. Based on results from analyzing $\mathbf{R}_3^{SE2}$, we create additional SSMs for $\langle$ *API*, *MySQL*, *group work*$\rangle$. We refer to these as $\mathbf{R}_3^{SE2}$-SSMs. The instructor assigned *group work* SSM as a for-credit activity, contrasting the rest of the SSMs, which were extra credit. The SSMs for each activity match the label's name, except for the *learning new material* label, which we refer to as *study strategies*, designed to improve students' overall study skills. This material includes online sources cultivated by the instructors, such as informative videos (helpful for visual learners [31]), articles, practice problems, interactive activities, university resources, and useful apps. For each question 1, we ask students to explain what resources they looked at, report on their usefulness (SSM-RQ-G2), and explain whether the material helped them improve the target skill (SSM-RQ-G3). We also ask questions specific to the SSM, such as asking them to provide screenshots for completing practice problems or interactive segments for Python and Coding and GitHub (SSM-RQ-G1), set a SMART goal for time management, and describe how they can incorporate the study strategies they learned into their study routines (SSM-RQ-G3).

Phase IV Part 1: Quantitative Analysis

First, we capture students' completion rates for each $\mathbf{R}_2^{SE2}$-SSM reflection activity. We consider the completion rates to measure students' interest in the material. For $\mathbf{R}_2^{SE2}$-SSMs, out of 81 total students, 60 (74.07%) completed at least one of the optional SSM reflection activities for extra credit by the due date. Out of the 60 students, 32 (53.33%) completed all four reflection activities. 11 completed 2 of 4 modules (18.33%), and 4 (6.67%) completed 3 of 4 reflection activities.

For $\mathbf{R}_3^{SE2}$-SSM, of 81 students, we find that 80.25% (65) completed at least one of the SSMs, and 25 (30.86%) completed all six (excluding the group work module). The highest number of completed SSMs is six (all activities), accounting for 30.8% of students, and nearly half of the students completed at least four of them, suggesting that students were generally interested in the modules. Furthermore, by examining the frequency of completion for each topic, instructors can gain a general understanding of which subjects students find valuable to explore further and where students may still have challenges. For example, the time management module has the highest participation, and the MySQL module has the lowest participation. Although participation may depend on various factors, such as semester workload, it also conveys student interest. Despite variations in individual completion, the average completion rate for each module is 52.26%. These results suggest that participating students value additional support and an opportunity to

achieve extra credit by completing relevant activities. In the final EC-RQ for the SE2 course, we included questions about students' experiences with the SSMs. One question asked students to answer whether they believed the SSMs should be included in the following semester. The students reported the following: 79% (52) of the respondents indicated *yes*; 3% (2) of the respondents responded *no*; 18% (12) of respondents marked *N/A*, indicating that they did not complete the SSM.

We conclude that the students found these resources valuable, given the relatively high participation rate. This perception may be attributed to the timely and relevant nature of the resources, as they directly addressed the most common challenges students faced in near real-time. Since we offer SSMs as optional, low-value extra-credit assignments, students had the freedom to choose whether or not to engage with them. By providing a manageable load of diverse and closely relevant materials that cater to various learning preferences, instructors allow students to select the resources that best match their needs. In addition, the voluntary nature of participation and the relevance of the issues likely motivate students to participate, as evidenced by their completion of multiple SSMs. Furthermore, the diversity of materials on the same topic is essential in such interventions, as students have a wide range of specialized learning needs.

These results further support the idea that students value the SSMs. However, the optional nature of the SSMs was a critical factor in that perception. In SE3A and SE3B, the instructors assigned the SSM as a for-credit activity in the course. In SE3A, in response to "Do you think that we should include these student support modules for future semesters?" 22 (34%) students replied *yes*, 3(5%) responded *no*, and 39 (60%) responded *yes, but they should be optional*, and 1 (2%) did not answer the question. SE3B produced similar results, with 42 (51%) responding *yes they should be optional*, and 39 (47%) responding *yes*, with 1 (1%) responding *no*. These results suggest that students may be more receptive to the SSMs as extra credit activities, allowing them to select which resources to use and avoid less relevant subjects.

Phase IV Part 2: Qualitative Analysis

Although multiple-choice surveys are valuable for aggregating general feedback, free-written text responses offer more nuanced insights. Instructors need both to illustrate a complete image of students' experiences. This section analyzes students' responses to EC-RQs and SSM-RQs to understand their perception of the interventions. They reported their motivation to complete the modules in one of the following ways: (1) to improve their understanding of the material, (2) to gain extra credit, (3) or both. One student, for example, wrote that they completed the material "[t]o get extra credit but also to [...] figure out essential things I needed to complete work." Another student wrote that they were motivated by the extra credit. However, the material was "extremely informative," and the student used what they learned to "enhance" their studying. Some students identified SSMs as a solution to their challenges; for example, one student mentioned time management as a challenge and referenced using the time management SSM to address it. Two students discussed using the GitHub support modules to improve proficiency with GitHub. One mentioned that they saved the resources for reference and utilized the "incredibly helpful" interactive resources, which helped them "navigate [GitHub]" better. Another student mentioned using the "GitHub commands resources" to "overcome" their challenge with Github. In $\mathbf{R}_4^{SE2}$, one student mentioned working with FastAPI as a challenge and the FastAPI support material as a solution, along with other Can-

vas resources. Another student who mentioned that their biggest challenge was "making sure the SQL database and FastAPI were correct" wrote that they "viewed many of the extra resources provided in the feedback modules" and that those resources led to others that helped deepen their understanding of course content.

These reflections suggest that providing diverse resource formats and subjects is vital because students have diverse needs. For example, one wrote that the SSMs "had a lot of varying types of information that made it very easy to find out what [they] needed to go through" and another expressed appreciation that the SSMs provide the information they needed in various formats and wording. Another student wrote that the information in the modules was "relevant and useful" and that the usefulness depends on the type of help students need.

Our results also suggest that what may work for one student may not work for another. For example, one student in the *time management* SSM-RQ reflection mentioned that they "found the videos most useful," but another student "liked reading the articles [...] rather than the videos." Moreover, several students found the time management SSM the most useful, one of which indicated that they knew they "wasted a lot of time throughout the day[,] but the module helped [them] truly see how much [they were] wasting" and another said that they were "always trying to improve in that area since [they] have a bad work/life balance." However, another student expressed less interest in the material, as it "contained a lot of the same advice [they] have heard many times before." The student explains that the issue is not "knowing what to do but having the discipline to do it." Another student stated that the interactive resources didn't help them learn, while a different student specifically indicated that the "interactive [resources] were much more helpful." This variation highlights the need for instructors to consider the full diversity of student experiences. Moreover, some students mentioned explicit examples of how the modules supported their coursework. For example, one student noted that the API videos helped them resolve a problem they encountered "at the beginning of the assignments."

Some students reported that they did not find the SSMs beneficial; for example, one mentioned that they "found them unnecessary," and another said that they had their own solutions that worked, so the modules were not helpful. One major criticism of the SSMs is their delivery as a graded activity during SE3, rather than extra credit. For example, one student commented that some SSMs were useful and some were "annoying" to complete, as they did not need them and felt "forced" to complete them since it was a graded activity. Other students explicitly said that they believed the modules would be better as extra credit, and one added that they did not have enough time to complete the regular coursework, let alone the SSMs.

These examples highlight the value of providing students with diverse supplemental material to complement their strengths and and promote self-directed learning.

Limitations and Future Work

This work introduces the first implementation of our course intervention development cycle (CIDC), leaving many avenues for improvement. One limitation of this study is the model's reliance on student opinions, which are inherently subjective and vary in depth and completeness, affecting the relevance of generated insights. In future work, more objective metrics, such as grades or attendance, could supplement the text responses for a more reliable analysis. We plan to integrate these objective measures into future versions of the theme extraction pipeline.

Moreover, this study emphasized the impact of interventions with a limited focus on model evaluation. To bolster reliability, we plan to rigorously evaluate the LLM's performance against human-labeled data, optimizing its accuracy. Providing strong examples of good reflections in the prompt across several semesters may also improve model performance, leading to iterative refinement over time. Furthermore, as AI advances, our method must align with emerging research and industry trends to fully leverage their evolving capabilities.

Finally, this method addresses a limited scope of student problems. While this study focuses on proficiency-related challenges, future work will expand interventions to address a broader range of student difficulties.

Conclusion

In this work, we present a student intervention method through our *intervention development cycle*, comprised of four phases: Phase I: Reflection Collection; Phase II: Challenge Identification and Classification; Phase III: Intervention Development; and Phase IV: Impact Analysis. We illustrate an implementation of this method using four courses over three semesters (SE1, SE2, SE3A, SE3B) of a software engineering course. To support this method, we define a taxonomy of software engineering challenges that instructors can adapt to other courses. We classify these challenges into *primary* and *secondary* labels, the former describing the challenge itself and the latter describing the context in which the challenge occurs.

Completion and reflection data suggest that students find supplementary materials based on their most common challenges (SSMs) helpful. Student needs vary widely within each course, and a diverse selection of subjects with various delivery mediums allows students from various backgrounds to improve their skills. Our results suggest that students are generally willing to complete these activities and can apply them to their learning but view them more favorably as extra credit rather than a for-credit component of the course. Furthermore, our initial investigation reveals that LLM-prompting methods can streamline phases of the intervention cycle.

Effective communication between students and instructors can provide valuable information to improve the course refinement process. By drawing on students' experiences, we can develop successful interventions that resonate with them. This work demonstrates the potential for designing interventions using insights extracted from student reflections. Our initial findings demonstrate the efficacy of using the CIDC to develop course interventions. In future work, we will refine and evaluate our method to further support instructors' efforts to improve student educational outcomes.

References

[1] J. Floden, "The impact of student feedback on teaching in higher education," *Assessment & Evaluation in Higher Education*, vol. 42, no. 7, pp. 1054–1068, 2017. [Online]. Available: https://doi.org/10.1080/02602938.2016.1224997

[2] L. Mandouit, "Using student feedback to improve teaching," *Educational Action Research*, vol. 26, no. 5, pp. 755–769, 2018. [Online]. Available: https://doi.org/10.1080/09650792.2018.1426470

[3] S. Wickramasinghe and W. Timpson, "Mid-semester student feedback enhances student learning," *Education for Chemical Engineers*, vol. 1, no. 1, pp. 126–133, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1749772806700173

[4] R. L. Miller, E. Amsel, B. Marsteller Kowalewski, B. C. Beins, K. D. Keith, and B. F. Peden, Eds., *Promoting Student Engagement (Vol. 1): Programs, Techniques and Opportunities*. Society for the Teaching of Psychology, 2011.

[5] S. Wickramasinghe and W. Timpson, "Mid-semester student feedback enhances student learning," *Education for Chemical Engineers*, vol. 1, no. 1, pp. 126–133, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1749772806700173

[6] S. Cunningham, M. Laundon, and A. Cathcart, "Beyond satisfaction scores: visualising student comments for whole-of-course evaluation," *Assessment Evaluation in Higher Education*, vol. 46, pp. 1–16, 08 2020.

[7] F. N.-A. Alhija and B. Fresko, "Student evaluation of instruction: What can be learned from students' written comments?" *Studies in Educational Evaluation*, vol. 35, no. 1, pp. 37–44, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0191491X09000066

[8] L. Mikalayeva, "Introduction: encouraging student reflection—approaches to teaching and assessment," *European Political Science*, vol. 19, no. 1, pp. 1–8, Mar 2020. [Online]. Available: https://doi.org/10.1057/s41304-018-0182-7

[9] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Comput. Surv.*, vol. 55, no. 9, jan 2023. [Online]. Available: https://doi.org/10.1145/3560815

[10] M. Dorodchi, A. Benedict, D. Desai, M. J. Mahzoon, S. MacNeil, and N. Dehbozorgi, "Design and implementation of an activity-based introductory computer science course (cs1) with periodic reflections validated by learning analytics," pp. 1–8, 2018.

[11] B. D. Jones, Y. Miyazaki, M. Li, and S. Biscotte, "Motivational climate predicts student evaluations of teaching: Relationships between students' course perceptions, ease of course, and evaluations of teaching," *AERA Open*, vol. 8, p. 23328584211073167, 2022. [Online]. Available: https://doi.org/10.1177/23328584211073167

[12] J. Leckey and N. Neill, "Quantifying quality: The importance of student feedback," *Quality in Higher Education*, vol. 7, pp. 19–32, 04 2001.

[13] C. D. Carly Steyn and A. Sambo, "Eliciting student feedback for course development: the application of a qualitative course evaluation tool among business research students," *Assessment & Evaluation in Higher Education*, vol. 44, no. 1, pp. 11–24, 2019. [Online]. Available: https://doi.org/10.1080/02602938.2018.1466266

[14] M. Dorodchi, A. Benedict, E. Al-Hossami, A. Quinn, S. Wiktor, A. Benedict, and M. Fallahian, "Clustering students' short text reflections: A software engineering course case

study (full paper)," in *Joint Proceedings of the Workshops at the International Conference on Educational Data Mining 2021 co-located with 14th International Conference on Educational Data Mining (EDM 2021), Held Virtually, 2021*, ser. CEUR Workshop Proceedings, T. W. Price and S. S. Pedro, Eds., vol. 3051. CEUR-WS.org, 2021. [Online]. Available: https://ceur-ws.org/Vol-3051/CSEDM_4.pdf

[15] D. R. Stead, "A review of the one-minute paper," *Active Learning in Higher Education*, vol. 6, no. 2, pp. 118–131, 2005. [Online]. Available: https://doi.org/10.1177/1469787405054237

[16] A. Benedict, S. Wiktor, M. Fallahian, M. Dorodchi, F. D. Pereira, and D. Gary, "A recommendation system for nurturing students' sense of belonging," in *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky*, N. Wang, G. Rebolledo-Mendez, V. Dimitrova, N. Matsuda, and O. C. Santos, Eds. Cham: Springer Nature Switzerland, 2023, pp. 130–135.

[17] I. Vayansky and S. A. Kumar, "A review of topic modeling methods," *Information Systems*, vol. 94, p. 101582, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306437920300703

[18] Y. Hu, S. Zhang, V. Sathy, A. T. Panter, and M. Bansal, "Setsum: Summarization and visualization of student evaluations of teaching," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*. Association for Computational Linguistics, 2022.

[19] W. Saeed and C. Omlin, "Explainable ai (xai): A systematic meta-survey of current challenges and future opportunities," 2021.

[20] M. U. Hadi, Q. Al-Tashi, R. Qureshi, A. Shah, A. Muneer, M. Irfan, A. Zafar, M. Shaikh, N. Akhtar, J. Wu, and S. Mirjalili, "Large language models: A comprehensive survey of its applications, challenges, limitations, and future prospects," 07 2023.

[21] J. Meyer, R. Urbanowicz, P. Martin, K. O'Connor, R. Li, P.-C. Peng, T. Bright, N. Tatonetti, K. Won, G. Gonzalez, and J. Moore, "Chatgpt and large language models in academia: opportunities and challenges," *BioData Mining*, vol. 16, 07 2023.

[22] J. Rajala, J. Hukkanen, M. Hartikainen, and P. Niemela, "Call me kiran – chatgpt as a tutoring chatbot in a computer science course"," in *Proceedings of the 26th International Academic Mindtrek Conference*, ser. Mindtrek '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 83–94. [Online]. Available: https://doi.org/10.1145/3616961.3616974

[23] J. K. Matelsky, F. Parodi, T. Liu, R. D. Lange, and K. P. Kording, "A large language model-assisted education tool to provide feedback on open-ended responses," 2023.

[24] B. Chen, Z. Zhang, N. Langrené, and S. Zhu, "Unleashing the potential of prompt engineering in large language models: a comprehensive review," 2023.

[25] M. J. Parker, C. Anderson, C. Stone, and Y. Oh, "A large language model approach to educational survey feedback analysis," 2023.

[26] M. Dorodchi, A. Benedict, D. Desai, M. J. Mahzoon, S. Macneil, and N. Dehbozorgi, "Design and implementation of an activity-based introductory computer science course (cs1) with periodic reflections validated by learning analytics," 12 2018.

[27] S. Wiktor, "Leveraging emotionally-charged student reflections to improve classroom communication," in *Proceedings of the 2023 ACM Conference on International Computing Education Research V.2 (ICER '23 V2)*. Chicago, IL, USA: ACM, August 7–11 2023, p. 4.

[28] R. Pekrun, "Emotions and learning; educational practices series; vol.:24; 2014," 2014.

[29] T. Hailikari, R. Kordts-Freudinger, and L. Postareff, "Feel the progress: Second-year students' reflections on their first-year experience," *International Journal of Higher Education*, vol. 5, no. 3, pp. 79–90, 2016.

[30] B. Chen, Z. Zhang, N. Langrené, and S. Zhu, "Unleashing the potential of prompt engineering in large language models: a comprehensive review," 10 2023.

[31] K. Caratiquit, "Youtube videos as supplementary materials to enhance computer troubleshooting and repair techniques for senior high school students in the philippines," *SSRN Electronic Journal*, 01 2022.