

LessonPlanner: Assisting Novice Teachers to Prepare Pedagogy-Driven Lesson Plans with Large Language Models

Haoxiang Fan

fanhx6@mail2.sysu.edu.cn

Sun Yat-sen University

Guangzhou, China

Xingbo Wang

xiw4011@med.cornell.edu

Cornell University

New York, NY, United States

Guanzheng Chen

chengzh59@mail2.sysu.edu.cn

Sun Yat-sen University

Guangzhou, China

Zhenhui Peng*

pengzhh29@mail.sysu.edu.cn

Sun Yat-sen University

Guangzhou, China

ABSTRACT

Preparing a lesson plan, e.g., a detailed road map with strategies and materials for instructing a 90-minute class, is beneficial yet challenging for novice teachers. Large language models (LLMs) can ease this process by generating adaptive content for lesson plans, which would otherwise require teachers to create from scratch or search existing resources. In this work, we first conduct a formative study with six novice teachers to understand their needs for support of preparing lesson plans with LLMs. Then, we develop *LessonPlanner* that assists users to interactively construct lesson plans with adaptive LLM-generated content based on Gagné's nine events. Our within-subjects study ($N = 12$) shows that compared to the baseline ChatGPT interface, *LessonPlanner* can significantly improve the quality of outcome lesson plans and ease users' workload in the preparation process. Our expert interviews ($N = 6$) further demonstrate *LessonPlanner*'s usefulness in suggesting effective teaching strategies and meaningful educational resources. We discuss concerns on and design considerations for supporting teaching activities with LLMs.

CCS CONCEPTS

- Human-centered computing → User interface toolkits;
- Computing methodologies → Natural language generation;
- Applied computing → Computer-assisted instruction.

KEYWORDS

Large language models, lesson plan preparation, pedagogy-driven system

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

UIST '24, October 13–16, 2024, Pittsburgh, PA, USA

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0628-8/24/10

<https://doi.org/10.1145/3654777.3676390>

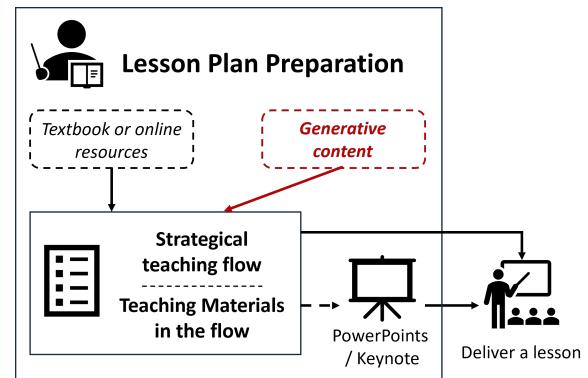


Figure 1: An example lesson plan preparation process. We build an interactive system *LessonPlanner* to assist teachers with generated content to prepare a documented lesson plan, which can be optionally transformed into slides by users and delivered in their courses.

ACM Reference Format:

Haoxiang Fan, Guanzheng Chen, Xingbo Wang, and Zhenhui Peng. 2024. *LessonPlanner: Assisting Novice Teachers to Prepare Pedagogy-Driven Lesson Plans with Large Language Models*. In *The 37th Annual ACM Symposium on User Interface Software and Technology (UIST '24)*, October 13–16, 2024, Pittsburgh, PA, USA. ACM, New York, NY, USA, 20 pages. <https://doi.org/10.1145/3654777.3676390>

1 INTRODUCTION

Effective teaching often stems from a well-prepared and organized lesson plan [67], which serves as a teacher's detailed description of the course of instruction or learning [36]. In this paper, we focus on the process of preparing a documented plan that normally describes the teaching flow and materials [49] for each lesson (e.g., 45-minute or 90-minute). Such a documented plan can serve as the basis for users to optionally create slides in PowerPoint or Keynote to deliver the lesson, or to be directly used as speaking notes in the class (Figure 1). However, constructing an effective lesson plan could be challenging for teachers, especially those who are novices or teaching a course for the first time [56]. For one thing, novice teachers lack experience in delivering a course using effective strategies [6, 23], e.g., Gagné's Nine Events of Instruction [28], which

helps teachers to prepare and deliver instructional content while considering and addressing the students' conditions, like gaining attention and presenting a stimulus at the beginning of a class. For another, even when novice teachers have a mind of planning an event or activity (e.g., in-class exercises) at a certain time of a course, they could face difficulties in curating relevant materials to execute the activity [23, 59, 61]. As a common practice, novice teachers often refer to others' lesson plans and search related materials in textbooks or online [59]. Nevertheless, these traditional approaches could be limited by the diversity of referred materials and the ability to support teachers in adapting the content in the lesson plan based on their personal thoughts.

Recent advances in large language models (LLMs) show great potential to address these challenges in composing a lesson plan by suggesting in-situ-generated content that adapts to teachers' thoughts. In fact, many teachers have explored the usage of LLMs to prepare for their lectures [47, 48]. For example, Mollick and Mollick [48] suggested the prompts to LLMs to implement effective teaching strategies in the classroom, such as providing diverse examples to help students comprehend abstract concepts and collecting test questions to help students assess their knowledge. However, querying the LLMs in the wild, e.g., via the web app of ChatGPT, could lead to a fragmented lesson plan because of the difficulties in organizing all of the LLM outputs. More importantly, due to the lack of experience in teaching and guidance for prompting, the generated content may not be comprehensive and readily usable, thereby requiring iterative refinement [3, 9, 38]. There is a need to build an interactive system with customized features to ease the process of adopting LLMs to prepare a comprehensive lesson plan. Nevertheless, little is known about what are the design requirements for such a lesson planning system, how LLMs can help to satisfy these requirements, and how would teachers perceive and collaborate with the system.

In this paper, we introduce *LessonPlanner*, an interactive system that offers pedagogy-driven generated content to assist users in constructing lesson plans. We first conduct interviews with six participants, including three novice teachers and three teaching assistants in the universities, to understand their challenges and needs for support when using LLMs to prepare lesson plans. The findings reveal users' demands for the generated content that aligns with effective teaching strategies and flexible interactions with LLMs. Based on the derived design goals from the interviews, we build *LessonPlanner* as a web app powered by the LLM GPT-4.0. In *LessonPlanner*, users can first input basic course information (e.g., course name, topic of one lecture) to generate an outline of the lesson plan with teaching strategies suggested by Gagne's nine events [28]. Then, for each section of the outline, users can extend it with user-specified LLM-generated activities belonging to each instructional event to get suggested teaching materials, and they have the option to customize the materials via prompts. At any time of the lesson plan preparation process with *LessonPlanner*, users can select any content to ask the LLM to explain it and freely query the LLM about anything, including knowledge delivery strategies and presentation suggestions for creating slides.

We first evaluated *LessonPlanner* via a within-subjects study in which twelve graduate students or senior undergraduates act as teachers and prepare lesson plans for their familiar course topics.

The results showed that *LessonPlanner* can significantly improve the quality of outcome lesson plans and ease users' workload in the preparation process. We further conducted expert interviews with five novice teachers (who have less than three years of teaching experience) and one experienced teacher across various educational stages and subjects. Experts further demonstrated that *LessonPlanner* is useful in lesson planning because it offers a well-organized outline and inspiring content. We discuss the concerns and implications of our study on facilitating users in teaching activities with LLMs.

In summary, this paper has three main contributions. First, we present *LessonPlanner*, an interactive system that leverages generated content to assist teachers in preparing lesson plans. Second, via a within-subjects study and an interview study, we offer empirical evidence on *LessonPlanner*'s effectiveness and usefulness in helping novice teachers prepare lesson plans. Third, based on our findings, we offer design implications for future systems that use large language models to assist teachers.

2 RELATED WORK

2.1 Preparation of Lesson Plans

A lesson plan is the instructor's road map of what students need to learn and how it will be done effectively during class time [49]. A carefully constructed lesson plan allows teachers to enter the classroom with more confidence and achieve effective teaching outcomes of their courses [31, 32]. As suggested by the CIPP (Context, Input, Process, and Product) evaluation model [63], an effective lesson should be needs-oriented, resource-adequate, systematically executed, and outcome-focused. A carefully constructed lesson plan can reveal the first two aspects of the CIPP model [5], that is, goals aligned with subject and societal demands as Context and a well-organized content plan as Input.

However, teachers, especially those who are novices or teach a course for the first time, could find it difficult to construct a well-designed lesson plan. For example, they may need to spend a lot of time writing a lesson plan from scratch [23], finding high-quality resources related to the lesson [6, 59], and thinking of the strategies to teach each knowledge concept [23]. Researchers have explored various methods that can assist in the preparation of lesson plans. They have tried using knowledge graphs [16, 56, 58] and recommendation systems [17] to integrate existing web resources, which could make it convenient for teachers to find needed teaching materials. For example, CollectiveTeach [56] retrieves and collects documents related to a specific topic based on a particular optimization objective, and then rearranges the documents into a coherent lesson plan. They have also proposed a variety of lesson planning systems [4, 15, 17, 50, 62, 72]. For instance, Pender et al. [50] have developed the CLEVER digital web platform, which includes Didactic Guidance, Content Management, Platform Services, and the Data section. It assists users in selecting existing content from the platform's library or in creating new content. Zain [72] also proposed CIDS (Collaborative Instructional Design System), a lesson plan design system aimed at assisting teachers in designing and implementing the requirements of 21st-century learning, integrating features of the ASIE (Analyze, Strategize, Implement, and Evaluate) Instructional Design Model and the Professional Learning

Community. Nevertheless, these previous approaches and systems largely rely on existing high-quality teaching resources related to the lessons, while such resources may not always be available for any lesson that the teachers want to deliver.

Our work is motivated by the benefits of preparing a carefully constructed lesson plan and is aligned with previous work that aims at easing this preparation process. Different from previous work, we explore the usage of machine-generated content to assist lesson plan preparation.

2.2 Large Language Models in Education

Researchers have started to explore the usage of large language models (LLMs) in educational settings due to their ability to provide personalized, efficient, and engaging learning and teaching experiences [46]. For learners, depending on a student's needs and learning style, LLMs are able to create self-study and self-assessment materials [18, 55], such as knowledge flashcards or course-specific questions [8, 10, 24, 27]. For example, since ChatGPT demonstrated strong insights in explaining medical questions without any additional training [39], Divito et al. [24] have utilized it to support medical learners in problem-based learning. Furthermore, LLMs have the capability to encourage students to think critically [1] e.g., by creating chatting chances during learning and adding suitable problems if needed [22].

For teachers, LLMs can help to automate the knowledge assessment process and provide personalized feedback to students [7, 69]. Additionally, teachers can get support from generated teaching materials including varied examples, instruction notes [11], explanations [48], post-class exercises [60], a list of frequently asked questions [45], and so on. For instance, Mitra et al. [45] proposed a system named RetLLM-E that assists educators in acquiring frequently asked questions from students related to their courses and in generating responses. The system initially retrieves context from student questions previously answered by teachers on a forum and from related course materials. It then uses this information to prompt LLMs to generate specific, high-quality, and precise answers to students' questions, which generally have a better quality than other available answers on the forum.

Despite the promising potential, few works have explored using LLMs to help teachers prepare their lesson plans. In this paper, we will first work with six novice teachers to understand their opinions about using LLMs for preparing lesson plans and identify their needs for support.

2.3 Interactive Systems that Supports Users with Large Language Models

Human-Computer Interaction (HCI) researchers have proposed a bunch of interactive systems that leverage LLMs to improve the efficiency and user experience in various tasks, such as group decision making [19, 29], software engineering [42, 51], human-UI interaction design [65], and so on. Specifically, many HCI researchers have embedded LLMs in their systems to support co-writing tasks [33, 35, 40, 44, 52], which is similar to our setting of co-writing a lesson plan with LLMs. For example, Lee et al. [40] developed *CoAuthor* and conducted a user study, which highlighted

that co-writing with LLMs can aid users by enhancing fluency, pooling ideas, and improving writing quality. *Wordcraft* [71] consists of an editor and some buttons used to call an LLM to generate various kinds of content. Compared with using a chatting interface, users find out this tool is more helpful and collaborative [71]. This finding inspires our design by replacing the chat box with simple pre-set buttons. Furthermore, *Jamplate* [70] is an idea reflection system that integrates LLM responses into the templates originated from the traditional reflection theory. It is shown that the well-organized generated content helps users a lot in expanding and refining their ideas [70]. Just like Jamplate does, we integrate education theory into our system to structure LLM responses in a lesson plan template.

We contribute an interactive system *LessonPlanner* that is in line with these previous interactive systems and offer insights into how LLMs can facilitate lesson plan preparation.

3 FORMATIVE STUDY

This study aims to help novice teachers design lesson plans in an efficient way and get high-quality outcomes. To achieve this, we conduct a formative study with six novice teachers or teaching assistants. The insights from the study will inform the design goals for *LessonPlanner*. The participants, with a mean age of 29.83 ($SD = 5.15$), include three females and three males, and we note as P1 to P6. Half of them (P2, P5, P6) are novice teachers with less than three years of teaching experience, and the other half are teaching assistants (P1, P2, P4) who have taught undergraduate students during one to three academic terms. Four of the participants teach courses related to computer science, while P1 teaches Architecture-related courses and P5 teaches Korean Intensive Reading.

3.1 Pedagogies for Lesson Planning

Before the interviews, we prepared three educational theories usually used to guide teachers in preparing structured lesson plans, which have proven to be effective [2, 53, 57]. They are easy to implement in the system to assist teachers in designing lesson plans. **Bloom's Taxonomy** is a hierarchical classification of cognitive skills that educators use to structure curriculum learning objectives, assessments, and activities, ranging from lower-order thinking skills like remembering, understanding, and applying, to higher-order skills like analyzing, evaluating, and creating. **Kolb's Learning Cycle** is a four-stage model of experiential learning that emphasizes the process of learning through experience, consisting of concrete experience, reflective observation, abstract conceptualization, and active experimentation. **Gagné's Nine Events of Instruction** involve the actions of both teachers and learners throughout the teaching process [37]. The nine events include: **Gaining Attention**: Present introductory activities that engage learners; **Informing Learners of Objectives**: Clearly state the learning goals and outcomes; **Stimulating Recall of Prior Learning**: Encourage learners to remember and connect previous knowledge; **Presenting Stimulus**: Introduce new content and information; **Providing Learner Guidance**: Offer instructions and strategies to help learners understand and process the new content; **Eliciting Performance**: Have learners practice what they have learned; **Providing**

Feedback: Give constructive feedback on learners' performance; **Assessing Performance:** Evaluate learners' understanding and skills; **Enhancing Retention and Transfer:** Use activities that help learners retain information and apply it to new situations. The nine events and corresponding activities prepared for the discussion in the formative study are summarized in Table 1. Each event may occur sequentially in a class as described above, but they can also be repeated to better organize the instructional content.

3.2 Procedure

To assist novice teachers in solving the difficulties in lesson planning and offer a better user experience, we conduct a two-phase formative study to comprehensively understand the users' needs. Initially, the participants are involved in semi-structured interviews. A set of questions is presented, and they are encouraged to express their viewpoints. The discussions are intended to investigate the processes involved in creating their ways of routine lesson plan creation (e.g., “Do you typically design instructional notes or slides for a 90-minute class?”, “Where do you like to access related materials when developing lesson plans?”), the potential of integrating the three educational theories (introduced in subsection 3.1) into a lesson plan (e.g., “Do you agree that Gagne's Nine Events can effectively guide instruction?”), and the challenges encountered during the design process (e.g., “What challenges do you face during this process?”). Furthermore, we specifically investigate their perspectives regarding LLMs such as ChatGPT in the design of lesson plans. Each interview has a duration of around 45 minutes in this phase.

Subsequently, they are invited to participate in a co-design session aimed at exploring and evaluating various interface designs. The objective is to identify the user interface requirements for a system designed for lesson plan design. Four slide decks (shown in Appendix A) are presented to participants, featuring example designs. These include a page for input meta-data of class, an outline overview page, and two editing pages with LLMs in different forms. Participants are provided with the ability to resize and crop screenshots, draw shapes, and use text boxes for the purpose of designing. One author assists with sketching by making edits based on participants' responses. We also ask questions to provide more detailed explanations of their ideas or explore different design options. Each participant spends about 15 minutes in this session.

3.3 Findings

We use the reflexive thematic analysis method [12] to analyze the transcribed recordings of each participant's semi-structured interviews and co-design sessions. All mentioned that they primarily prepared a lesson plan for a lecture in the form of PowerPoint slides. P2 said that he sometimes chose to annotate some text in the slides or textbook to remind themselves of extra information, such as proof of formulas or supplemental examples. Except for P2, no participants reported that they left notes under the slides or in a separate paper as a reminder of critical information during the lecture, such as the logic of proof and supplemental examples. Nevertheless, they all agreed that having a word-like document to list the main flow and content of a lecture was helpful, as they could “easily build up slides based on the documented lesson plan” (P6). We

summarize the participants' challenges in lesson plan preparation and using large language models (LLMs) to assist the preparation as below.

3.3.1 Challenges in Lesson Plan Preparation. C1. Lack of adaptive support in planning effective teaching strategies in the lessons. All participants said that they mostly relied on their teaching experience and others' plans to design their teaching strategies in a lesson. Nevertheless, after we introduced Gagne's Nine Events of Instruction (the list of the events is shown in Table 1), all participants agreed that they actually adopted some of the events in their lectures. *“I did not get official training on educational or teaching theories, but I found that I had enacted several strategies in the suggested events of instruction. These nine events will be generally helpful in my course”* (P2). Compared to the other two theories mentioned in subsection 3.1, two participants (i.e., P1, P2) confirmed that Gagne's theory was more feasible to be embedded in the system because *“it is relatively more specific and provides teachers with comprehensive guidelines”* (P2). Furthermore, four participants (i.e., P1, P3, P4, and P6) with insufficient teaching experience further mentioned that they would like to exercise all the nine events in their lectures but they lack adaptive support in incorporating these events into a specific lesson. *“I would like to try the suggested ‘gaining attention’ and ‘providing learner guidance’, but the example usages of these strategies could not be directly adopted in my Experiments of Data structures course”* (P1).

C2. Time-consuming process in searching for adequate teaching materials. All participants indicated that preparing for a lesson, especially the one they taught for the first time, is a time-consuming process, where they spent most of the time preparing the teaching materials. *“I usually need to spend a considerable amount of effort finding and determining the proper materials that support the teaching activities in my course, e.g., finding a good case to illustrate the real-world application of the taught concepts”* (P5). Even though they can get started from textbooks or others' lesson plans, all participants expressed their desire to customize the teaching flow and materials based on their thoughts. As such, they spent time looking for related high-quality materials online or creating needed materials by themselves. *“I was used to prepare an in-class exercise after explaining a key concept. I normally search for suitable exercises online but the returned results were often disorganized or unrelated. I had to pay a lot of attention to identify the ones I need”* (P1).

3.3.2 Challenges on Using LLMs in Lesson Plan Preparation. C3. High mental demand in manipulating prompts to get related content. Three participants (P3, P4, and P6) had experiences using LLMs (Specifically, ChatGPT) to prepare for their lessons. For example, P3 reported that she had used ChatGPT to generate precise answers to the questions that students may ask when delivering knowledge points. The others also see the potential of using LLMs to generate suggestions on the teaching flow and materials based on their needs. However, P5 and P6 commented that it was or would be mentally demanding to think of the prompts for LLMs to get needed content. *“Searching for information directly seems more efficient than querying ChatGPT with specific questions.”* (P6). *“If LLMs cannot*

provide satisfying materials for me instantly, I prefer not to engage in back-and-forth dialogue to refine the results” (P5).

C4. Distrust of the usefulness of generated content. Except for P3, the remaining five participants conveyed a sense of distrust toward the outputs of the LLM. On one hand, some (P2, P5, P6) doubted the accuracy of the content. “*I don’t trust the detailed content generated by LLMs, and I would definitely double-check it before using it. Therefore, I generally only use LLMs to create a rough outline, and I search for the specific content myself*” (P1). On the other hand, some (P1, P2, P6) have raised concerns about the expertise of LLMs in specific subjects, suggesting that they may struggle to generate useful content. “*ChatGPT can generate many correct questions, but that doesn’t mean they are good questions in this course (Computer Graphics). Because it may lack a deep understanding of the subject*” (P6).

3.4 Users’ Preference on the Interface

During the co-design phase, all participants reached a consensus that the layout of our outline overview page, which organizes the outline into distinct blocks based on the generated subtitles, was a good design. P4 advised that “*the instructional events are important cues for me, and they should be more eye-catching.*”. For the course metadata collection page, P2 questioned that “*the available data input was excessively restricted. I believe it is necessary to enter the course designed for graduate or undergraduate students. Without this information, how can the system accurately determine the specific type of content to generate?*” Moreover, participants were requested to compare the LLM assistant interface integrated within the editor as opposed to a sidebar interface. P1, P2, P3, P5, and P6 expressed a preference for the LLM as a sidebar. “*It resembled common design practices*” (P3).

3.5 Teachers’ Needs for Generated Content in Lesson Plans

Table 1 summarizes the particular instructional activities our participants commonly incorporate during each of Gagné’s Nine Events in the classroom. Additionally, we ask their perspectives on the capacity of LLMs to contribute to the development of materials for the activities they have mentioned in each interview. The results are displayed in the third column of Table 1, with ‘Y’ if they think the LLMs can facilitate the activity and ‘N’ if they can not. The results indicate that, with the exception of subject-specific activities (e.g., providing source code, providing example sentences), most teachers organize comparable activities for a particular instructional event. For the majority of classrooms, LLM can help teachers in creating relevant instructional materials. This highlights the potential of LLMs to serve as assistants in lesson planning.

3.6 Design Goals of LessonPlanner

Our participants actively offered suggestions on the design of *LessonPlanner* to address their challenges. Based on their suggestions and related work, we derive the following four design goals (DGs) of *LessonPlanner*.

DG1: *LessonPlanner* should encourage and facilitate teachers to apply effective teaching strategies in the planned lessons.

Due to the insufficient adaptive support provided by teachers in lesson planning for the development of effective teaching strategies (C1), *LessonPlanner* to motivate and assist teachers in applying efficient teaching strategies, especially those who lack experience in teaching.

DG2: *LessonPlanner* should generate high-quality materials adapted to the planned teaching activities and provide guidance on how to deliver these materials. The process of searching for suitable teaching materials is time-consuming (C2). Hence, *LessonPlanner* is expected to produce high-quality materials that are in line with the teaching activities and offer suggestions on how to effectively deliver these materials in the classroom.

DG3: *LessonPlanner* should offer pre-set prompts about what users often want to ask the large language models for. The utilization of LLMs for lesson planning requires a high cognitive load (C3). Therefore, *LessonPlanner* offers pre-set prompts to assist users in getting materials more efficiently from LLMs.

DG4: *LessonPlanner* should provide flexible user control for interacting with the large language models and editing the lesson plans. Granting users the autonomy to select their preferred method of managing the content generated by LLMs, and to discard, modify, or regenerate it as needed, can mitigate users’ distrust towards LLMs (C4).

4 LESSONPLANNER

In this subsection, we present the design and evaluation of *LessonPlanner*, an interactive system that supports novice teachers to create lesson plans with large language models (LLMs). *LessonPlanner* is designed as a web application. The front end is implemented using Vue 3 and JavaScript, while the back end is created with Python FastAPI. Furthermore, we opt to utilize the OpenAI gpt-4-1106-preview as the LLM to provide generated content in *LessonPlanner*. The front end is in Chinese, so the Edge translation plugin was utilized to translate the website in order to present the illustrations in this chapter.

We carefully designed *LessonPlanner*’s interface and interaction based on the design goals obtained from the formative study. To plan a lesson with *LessonPlanner*, users can engage with the LLM to set teaching objectives to adhere to Bloom’s Taxonomy [25] and generate an initial outline of the lesson plan conditioned on Gagné’s Nine Events (DG1, Figure 2). For each section in the outline, users can see suggested teaching materials and strategies to enact this event (DG2, Figure 3: C1, D2). Based on the identified teachers’ needs for generated content in lesson plans in formative study, *LessonPlanner* presets some activities for each event about what users often want to ask the LLM for (DG3, D1, D2). Besides, *LessonPlanner* also offers users flexible options to regenerate the content of an event and copy it to the editor (D), freely edit the content (Figure 4: E, F), select wanted events (Figure 5), check the history of generated content (Figure 4: G), and so on (DG4). In the following subsections, we will detail the design and implementation of the key features in *LessonPlanner*.

Goal-Setting Page

A. Input the meta-data of the lesson

What is your course name? (required)
Data Structures and Algorithms

What is your course about? (required, please fill in the course content you want to cover in this course)
Quick sort

What stage of students are your courses designed for? (required)
Sophomore

Course objectives (Lesson Goals)

Please fill in your course objectives for this lesson (optional, just keywords, it will be automatically generated if not filled in).

Course objectives:

- Students will master the basic steps of quick sort, including selecting reference points, partition operations and recursive calls.
- Students can describe the "divide and conquer" principle of the quick sort algorithm and its application in data structures.
- Students will practice quick sort through exercises, manually sort a set of numbers and implement the algorithm through programming.
- Students will analyze the running time of quick sort and understand the impact of different benchmark point selections on performance.
- Students can evaluate and compare the efficiency of quick sort with other common sorting algorithms.
- Students will be able to explore strategies to improve the efficiency of quick sort, such as using median partitioning or randomizing the selection of benchmark points.

B. Refine lesson goals with LLMs

Course objectives:

- Students will master the basic steps of quick sort, including selecting reference points, partition operations and recursive calls.
- Students can describe the "divide and conquer" principle of the quick sort algorithm and its application in data structures.
- Students will practice quick sort through exercises, manually sort a set of numbers and implement the algorithm through programming.
- Students will analyze the running time of quick sort and understand the impact of different benchmark point selections on performance.
- Students can evaluate and compare the efficiency of quick sort with other common sorting algorithms.

Programming using python.
Remove item 6.
Add requirements.

C. Generate Lesson Goals

Build target Generate outline

Figure 2: Goal-setting page of *LessonPlanner*, with the illustration of the process of refining lesson goals with LLMs. The text in parentheses serves as a correction to the mistranslated output.

Lesson-Planning Page

A. Generate or Re-generate the Lesson Plan Outline

Generate a lesson plan outline for this lesson

B. Advanced Markdown Editor

Edit using Markdown editor

C. A Block Containing a Subsection

Open question: special requirements in sorting

C.1 The Content Area

C.2 Set the teaching event of this module
(Set Instructional Events)

C.3 Delete the module content

C.4 attract attention

5 minutes

Related application cases:
Do you want to know in advance where a specific element will be ranked in the entire sequence before the sorting is completed?

Introduce open questions related to teaching content:
Is there an algorithm that can efficiently sort a large number of elements when space complexity is high?

D. Assistant (Click on E.1)

Lesson Plan Assistant

regenerate Content evaluation and explanation suggestions
Generate slide presentation suggestions...

In this section you may need:
D.1 Core Action Set
D.2 Contextual Action Set:
Open questions or application cases
I need to (I need)

LLM outputs: D.4
History
D.5

D.3 Output Text Area

The current lesson plan content raises an attractive question, but does not clearly propose an algorithm to solve the problem or specific teaching steps, which may make students curious but also a little confused.

Suggestions for explanation:
Students, the problem we are going to discuss today is the eagerness to know the final position of an element during the sorting process. You can imagine that you are shopping online and want to know where a certain item ranks in the search results, but you don't want to wait for the entire list to load. So how do we implement such a function?
Thinking guidance: Think about it, if we want to find the position of an element after all sorting is completed, do we have to know the entire sequence?
Algorithm Challenge: In a scenario where space complexity is limited and all elements cannot be loaded at once, how can we efficiently complete such a requirement?
Now, please discuss in groups and propose possible solutions based on the questions just given.

Continue to ask:
Stop generating copy
Close Assistant

Figure 3: The lesson-planning interface for *LessonPlanner*. The text in parentheses serves as a correction to the mistranslated output.

Table 1: Nine events of Gagné’s instructional theory and the activities that our participants in the formative study suggest to use in each event. Participants also share their opinions on whether (Yes or No) LLMs are able to help them in each activity. In LessonPlanner, we have preset template prompts to LLMs for most of the suggested activities.

Gagné’s Event	Common activities or resources in this event	LLMs facilitated	Implemented
Gain attention	Pose open-ended questions or case studies	Y	Y
Inform learners of objectives	Create ordered lists of knowledge points Display table of contents in slide	Y Y	Y Y
Stimulate recall of prior learning	Compile prerequisite knowledge list Provide prerequisite knowledge examples	Y Y	Y Y
Present stimulus	Provide the definition Provide algorithms Provide source code Provide equations Provide a example sentence	Y Y Y Y Y	Y Y Y Y N
Provide learner guidance	Explain examples in detailed Design animations in PowerPoint Play videos	Y N N	Y - -
Elicit performance	Construct multiple choice or fill-in-the-blank questions Propose open-ended questions Construct group discussion topics	Y Y Y	Y Y Y
Provide feedback	Offer problem solutions	Y	Y
Assess performance	Assign homework	Y	Y
Enhance Retention and Transfer	Assign Projects as homework. Select topics for writing papers	Y Y	Y N

4.1 Goal-Setting Page

As shown in Figure 2A, on the goal-setting page, users need to input the course name (e.g., Data Structures), the topic of the planned lesson (e.g., Quick Sort), and the specific stage of the lesson tailored for (e.g., Sophomore (2nd-year undergraduate)). This meta information is utilized by *LessonPlanner* in all predefined prompts to facilitate content generation.

Users can input their lesson goals, or, in a more convenient way, click the “Generate Lesson Goals” button to first check the generated goals conditioned on Bloom’s Taxonomy. They can then modify the generated goals, e.g., remove item 6 and add requirements like “programming using python” (Figure 2: B), and click the “Generate Lesson Goals” again to iterate the design goals with LLM. Once users are satisfied with the lesson goals, they can click on the “Generate outline” button, which will initialize a lesson plan outline conditioned on Gagné’s Nine Events. Users now can proceed to the Lesson-Planning page (Figure 3), as described in the next subsection.

an h3 heading (e.g., 5 minutes) indicating the planned time for this event. In each section, there are related teaching materials and strategies, e.g., **Related application cases** “Do you want to know in advance where a specific element will be ranked in the entire sequence before the sorting is completed?”. As a lesson may not necessarily cover all of the suggested sections and may include other sections the users need, users have the options to ignore or specify needed sections, delete the section (C3), insert a new block, and specify the title of a new section at any location (C5). Users can also regenerate the lesson plan outline (A), e.g., if they are not satisfied with the current one.

Users can click the content area (Figure 3: C1) of each block to invoke the editing mode of this section (Figure 4: E) or click the “Edit using Markdown editor” button to invoke an advanced Markdown editor of all content in the lesson plan (Figure 4: H). The editor provides features like “bold”, “headings”, and “lists” that common editors have. Upon finishing the edition of the lesson plan, users can click the “Download Lesson Plan” button (not shown in Figure 3) at the bottom of the Lesson Planning page to download the lesson plan as a markdown file.

4.2 Lesson-Planning Page

4.2.1 Interactive block-based lesson plan outline. The generated lesson plan outline is displayed in the form of block-based interactive markdown text (Figure 3: C). In each block, the editable h1 heading, e.g., “Open question: special requirements in sorting”, is a summary of the content in one section in the planned lesson. Each section contains one or more suggested events (C4) that teachers can plan in this section. The events are sourced from Gagné’s Nine Events and separated by editable h2 headings (e.g., “attract attention”) with

4.2.2 LLM assistant. In the markdown editor of each blocked section (Figure 4: E), users can click the “Open the lesson plan assistant” button (Figure 4: E1) to invoke the LLM assistant in the sidebar (Figure 3: D) that helps users refine the content in this section. The assistant includes a Core Action set (D1), Contextual Action set (D2), and Output Area (D3). The Core Action set contains a collection of preset buttons designed to call functions that regenerate the content in this section, evaluate the content give instructional suggestions,

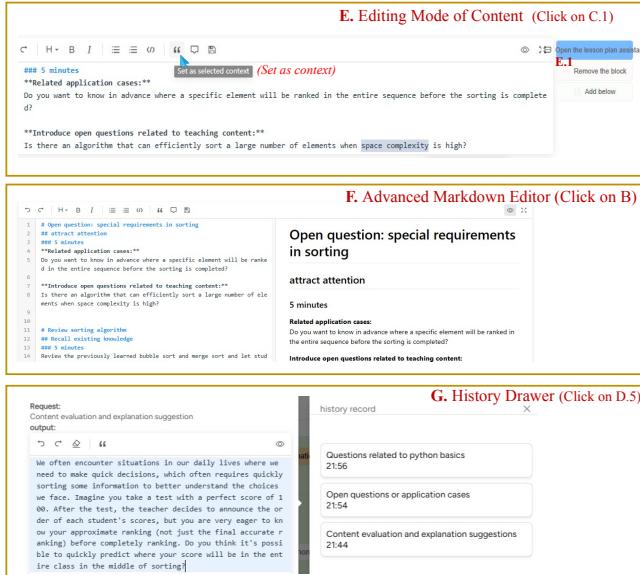


Figure 4: The subsequent changes or pop-up windows triggered by clicking a button on Figure 3.

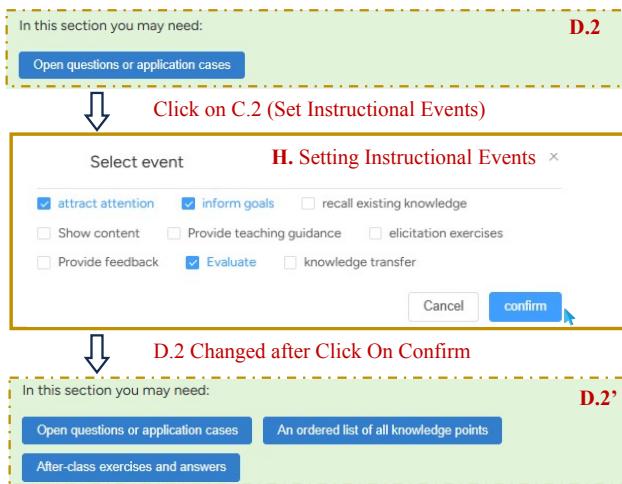


Figure 5: The process of setting instructional events by users in the lesson-planning interface.

and advice on presenting it, and give suggestions on making a slide based on it. The Contextual Action set consists of buttons that list all the preset activities of events (C4) under this section. Table 1 displays the implemented activities for each event in our system, derived from the results of the User Study, excluding activities that were difficult to implement because of the needed various prompts in different subjects. If clicking on "Set Instructional Events" (Figure 3: C2) to change events in the current section (Figure 5: F), the activities in the Contextual Action set will also change (Figure 5: D2 change into D2'). If users prefer to freely inquire about LLM, they can enable the "I need" button and question the LLM. The

Output Area contains a Markdown editor (Figure 3: D3) that displays the LLM outputs after users click any button or freely query LLM in the Action sets. It also has a text area under D3 for users to conduct multiple rounds of interaction with LLM on the content in the markdown editor. When interacting with LLM via either preset buttons or text area, users can select any text in the editor and click the "set as context" button (Figure 4: E) to use the selected text as context to prompt LLM. For example, if there are four key knowledge points in the current section, clicking "Generate Multiple Choice Questions" is highly likely to result in the generated questions that include a range of knowledge points. If users select the text of one knowledge point as context, the generated questions will be likely to be specific to that knowledge point. Users can select any content in the LLM outputs¹ and click the "Copy" button at the bottom of the assistant panel to copy to the pasteboard and then paste it in any position of the editor of this current section. Selecting the Trash icon (Figure 3: D4) will delete all content in the Output Area while clicking the "Stop Generating" button can halt the LLM outputs. Clicking the "Close Assistant" button will hide the assistant. Additionally, clicking the "history record" button (Figure 3: D5) allows users to access previously generated content (Figure 4: G).

4.3 Implementation Workflow

LessonPlanner incorporates a workflow that chains all prompts and system functions with flexible user control.

4.3.1 Metadata and generated objectives as slots in prompt templates. In the Goal-setting page Figure 2, teachers are able to enter courses' metadata (e.g., course name, course topic), which are used as slots in the prompt templates, as shown below.

- {course name}: e.g., Data Structures and Algorithms
- {lesson topic}: e.g., Quick sort
- {students stage}: e.g., Sophomore
- {lesson goals}: the content input by the user or LLM in the Lesson Goals area in the Goal-Setting Page

When designing Prompts for Core or Context Actions set, we always place this paragraph at the beginning of them:

I will instruct the course of {lesson name} - {lesson topic} for students in {students stage}. Here are my lesson goals: {lesson goals}.

This setting helps impose strict restrictions on all generated content. Sometimes, due to the lack of key information, LLMs are unable to infer accurate information about the course from the content of the current section or selected context, resulting in output that is unrelated to the course. This also applies to the prompts designed for "I need" feature or for having continuous conversations with LLM. Before engaging in a conversation, we always place this prompt at the beginning of the user's first request.

4.3.2 Leveraging the nine events to ensure control over generated content. As described in DG1, *LessonPlanner* should provide effective teaching strategies at each stage (i.e., outline generation, interacting

¹If nothing is selected, clicking the "Copy" button will copy all the content in the output text area to the pasteboard.

with LLMs) rather than merely offering educational resources. Additionally, displaying the outline in a block-based format for each section and ensuring that LLMs are aware of the corresponding teaching strategies requires a precisely defined workflow.

Combine all the instructional events and metadata when generating outlines. As seen in Figure 3 C4, recommended instructional events are set once a section of the outline is generated. The preset prompt for generating the outline is provided in supplementary material 1. Developers have confirmed that GPT-4 possesses background knowledge about Gagné's Nine Events in their trials on the preset prompt. Therefore, only the Nine Events and corresponding activities in Table 1 are included in the prompts of each action. LLMs have demonstrated superior reading comprehension and the ability to analyze lengthy texts [64]. We have fully leveraged this capability to create outlines that are aligned with the course content and incorporate the Nine Events effectively.

To ensure that the format of the contents generated by the LLMs meets the requirement and minimizes errors, the formatting instructions are set as simple as possible. We defined that a new section should start with a single "#", and instructional events in every section should be specified with "##". After generation, a *Formatter* then splits and maps the content into several blocks in the user interface. Also, the whole templates and instructional events in the section can be defined as slots, which will be used in some templates.

- {outline}: generated outline of the lesson plan.
- {current section}[events]: the events in current section.

Emphasize solely the instructional events involved in the current section. As for interacting with LLMs within a certain section, if the same context or even section content is assigned to different teaching events, we hope to click on the activities in the Core Actions set or use "I need" feature to bring results that match the current teaching events. For example, when clicking on "evaluate the content and give instructional suggestions" in the section with the event "Stimulate recall of prior learning", we expect a brief explanation. In contrast, in the section with "Provision of Learner Guidance", we anticipate detailed information to be provided about the knowledge. We dynamically insert the current section's instructional events and their corresponding definitions into the prompts through programming. It helps to remind the LLM to generate more relevant content based on these events. An example is shown below.

The educational theories involved in this section are as follows. Be sure to construct the lesson plan around the following events. Events that are not mentioned cannot be covered in this section.
({current section}[events] and their definitions, respectively.)

Based on our trials, it is not recommended to include all instructional events and their corresponding definitions and add the current events after that, which will make the generated content extensively cover each event and cause information redundancy.

4.3.3 Chaining the prompts with examples of input and output.

During the process of iteratively designing prompts, we found that

relying on descriptive statements frequently results in unstable output. To improve the robustness of the output, we design an example at the end of every prompt to ensure that the system generates results that align with expectations every time. For example, for the activity "Generate definition", the example

Input:

Context: Quick Sort

Output: **Quick Sort * *:

-Definition: Quick Sort is a divide-and-conquer algorithm. It works by selecting a 'pivot' element from the array and partitioning the other elements into two sub-arrays, according to whether they are less than or greater than the pivot.²

was added to the end of the prompt, forcing LLMs not to generate more detailed explanations to further explain the definition.

5 EVALUATION

To explore the effectiveness and user experience of *LessonPlanner* for assisting novice teachers in lesson planning, we conducted two user studies. First, we conduct a within-subjects study with 12 university students who have little or no teaching experience to use *LessonPlanner* and a baseline tool to prepare lesson plans for a course that they have learned before. Second, we conduct an expert interview with 6 teachers from different educational levels and subjects. Our research questions (RQs) are:

RQ1. How would *LessonPlanner* affect the lesson planning outcome?

RQ2. How would *LessonPlanner* affect the lesson planning process?

RQ3. How would users perceive *LessonPlanner* for preparing lesson plans?

5.1 Within-subjects study

5.1.1 Participants. We recruit 12 students (P1-P12, two females, ten males; age: $Mean = 23.42$, $SD = 1.98$) via word-of-mouth from a university in mainland China (Table 2). All participants, including two 4th-year undergraduate students and ten graduate students, major in artificial intelligence and are familiar with the course Data Structures and Algorithms. The experimental setup controls the lessons that participants need to prepare for, which leads to unbalances of majors and genders of our participants. We will discuss the limitations of this setting in the Discussion section. Nine of the participants have experience working as a teaching assistant (TA) in the university or as a home tutor for high-school students. All participants are quite interested in having a trial on utilizing artificial intelligence tools or online resources for lesson planning ($Mean = 4.33$, $SD = 0.47$; 1 - no interest at all, 5 - a large amount of interest). Nine participants are daily or weekly users of large language models (LLMs, e.g., ChatGPT and Claude), and the other three participants are infrequent users.

5.1.2 Experiment Setup. The goal of the within-subjects study is to quantitatively evaluate the *LessonPlanner*'s effectiveness and user experience compared to a baseline system.

²<https://en.wikipedia.org/wiki/Quicksort>

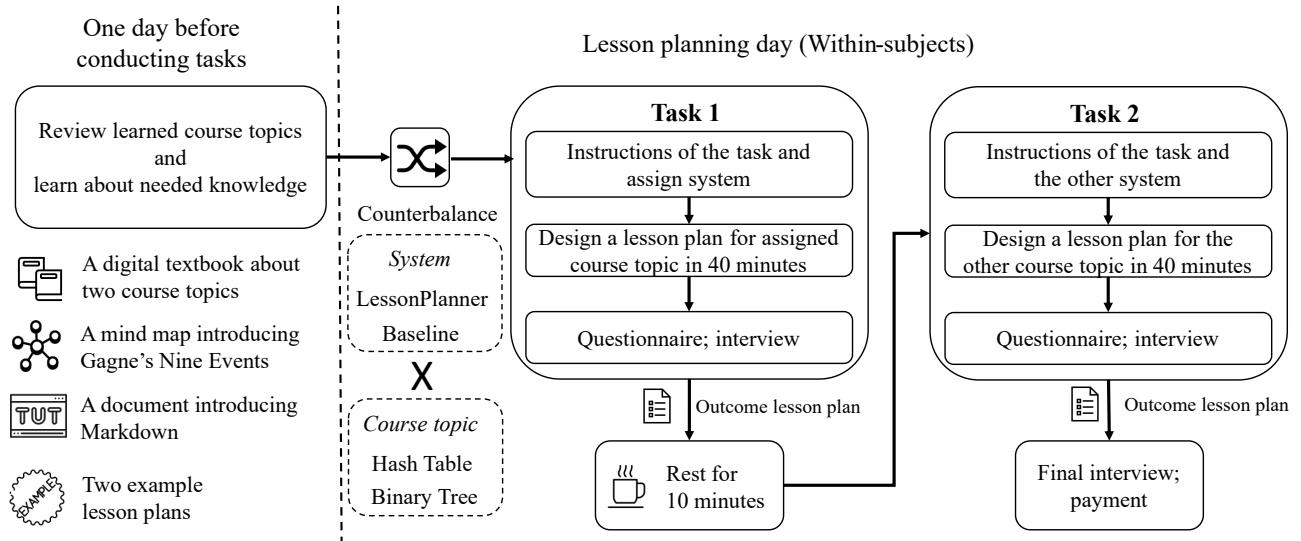


Figure 6: The procedure of the within-subject study.

Table 2: Participants involved in the within-subjects study.

ID	Gender	Age	Year	Experiences	Freq. of LLMs Usage
P1	M	22	Graduate	TA & Home Tutor	Daily
P2	M	23	Graduate	TA & Home Tutor	Weekly
P3	M	29	Graduate	N/A	Have Tried
P4	M	22	Graduate	N/A	Weekly
P5	M	25	Graduate	TA & Home Tutor	Weekly
P6	M	24	Graduate	TA	Infrequently
P7	M	22	Undergraduate	N/A	Weekly
P8	M	24	Graduate	TA	Daily
P9	F	23	Graduate	TA	Infrequently
P10	M	23	Graduate	TA	Weekly
P11	F	21	Undergraduate	TA	Daily
P12	M	23	Graduate	TA	Daily

Baseline condition. The baseline system is the ChatGPT web app that uses the GPT-4 model as used in *LessonPlanner*. In the baseline condition, participants can also use the Bing search engine in their web browsers and a markdown editor³ that is identical to the markdown component embedded in *LessonPlanner*.

LessonPlanner condition. In the *LessonPlanner* condition, participants are permitted to use Bing and *LessonPlanner*, but are not allowed to visit the ChatGPT web app or another markdown editor. In either condition, participants could freely choose whether, when, and how to use the provided tools.

Task-system assignment. In the recruitment survey, we ask participants to choose the course topics of Data Structures they are familiar with. We select the two topics that all participants indicate their familiarity with, *i.e.*, “Hash Table” and “Introduction and Traversal of Binary Trees”. We counter-balance the order of lesson planning tasks and use a system using Latin Square, with three participants in each of the four assignments:

- *LessonPlanner* - Hash Table, Baseline - Introduction and Traversal of Binary Trees

- Baseline - Introduction and Traversal of Binary Trees, *LessonPlanner* - Hash Table
- *LessonPlanner* - Introduction and Traversal of Binary Trees, Baseline - Hash Table
- Baseline - Hash Table, *LessonPlanner* - Introduction and Traversal of Binary Trees

5.1.3 Tasks and Procedure.

Each participants have two lesson-planning tasks. The prompt for each task is:

The lesson plan is a teacher’s detailed description of the course of instruction. Now, you are instructed to act as a teacher who will deliver a lesson about Data Structures - [Topic Name in this task] to second-year university students who are new to this subject. Your class is composed of 70 students whose academic abilities follow a normal distribution. The lesson is scheduled to last for 90 minutes, and you have 40 minutes available to develop your lesson plan.

The whole procedure is illustrated in Figure 6. Before the day of conducting lesson planning tasks, we sent participants the following materials, which they should spend 15-30 minutes reviewing before the tasks.

- Sections on Hash Tables and Introduction and Traversal of Binary Trees in a digital textbook of Data Structures and Algorithms
- A mind map that introduces Gagne’s Nine Events.
- A document that introduces the basics of Markdown operations.
- Two example lesson plans of course topics different from those in Data Structure and Algorithm.

On the day of conducting lesson planning tasks, each three participants that are in the same task-system assigned group come to our lab. We briefly introduce Gagne’s Nine Events and the considerations for a good lesson plan using the materials sent yesterday.

³<https://github.com/code-farmer-i/vue-markdown-editor>

To motivate them to put forth their best effort in the task, we inform them that an additional 80RMB reward will be given to the best lesson plan among others rated by a seasonal lecturer of Data Structures and Algorithms.

In each task, we first introduce the task and demonstrate the assigned system to participants. We then help participants to set up the environments of study in our provided computers or their laptops, *i.e.*, opening webpages of the baseline system or *LessonPlanner*, Bing, and digital textbook of the assigned course topic. Each participant then independently works on the lesson plan preparation task. We allocate 40 minutes for each task and inform them that they can finish earlier or have a few more minutes to complete the task. After each task, we ask participants to fill out a questionnaire about their experience in this lesson planning process and perceptions of the used system. Subsequently, we interview them for 5 minutes. There is a 10-minute break between two tasks for participants to take a rest. Upon completion of two tasks, we conduct a final semi-structured interview with the participants that focuses on their preferences on the used systems, perceptions on the generated content, opinions on the features of *LessonPlanner*, and suggestions for improving it. In total, each participant spends around 120 minutes in our experiment and receives 120RMB for compensation.

5.1.4 Measurements. We employ a standard 7-point Likert scale (1 - strongly disagree, 7 - strongly agree) to measure the quality of outcome lesson plans, experience in the lesson planning process, and perceptions towards the used system.

RQ1: Lesson planning outcome. To assess the quality of the outcomes, we invite a teacher with three years of experience teaching Data Structures and algorithms to score all outcome lesson plans in a shuffle order. The aspects of evaluation are adapted from Aziz et al. [5], based on the CIPP model, and the lesson plans are assessed on a 7-point Likert scale to indicate the degree to which each criterion is met. For each lesson plan, the teacher assesses it from five key aspects adapted from the CIPP lecture evaluation model [5] include the alignment of teaching content with learning objectives, facilitation of students' skill acquisition, integration of effective teaching materials, a balance between theoretical and practical activities, and the employment of effective teaching strategies. Sufficient detail has also been added as an aspect to further evaluate the lesson plans.

RQ2: Lesson Planning Process. Drawing from the NASA-TLX survey [30], we pose six questions to measure the workload during the lesson planning process, including the aspects of Mental Demand, Physical Demand, Temporal Demand, Performance, Effort, and Frustration. In the interviews, we encourage users to share how the system has increased or decreased their workload.

RQ3: Perception of LessonPlanner. First, the ten questions from the System Usability Scale (SUS) [13] are set in our questionnaire. We divide SUS into and analyze it from three levels: Effectiveness & Learnability, Use Efficiency, and Satisfaction. Then we adapted five questions from Jian's Trust Scale [34] and investigated how users trust the system by inquiring about the system's vigilance, potential negative impact on teaching, their trust in the system's ethical standards, commitment to users, and the reliability of outputs. In the interviews, we encourage them to share their

experiences with and preferences for each tool (*e.g.*, search engine; ChatGPT web app; *LessonPlanner*) and each component (*e.g.*, different pre-set activities and the editor in *LessonPlanner*). We further inquire about their reasons for trusting or distrusting the system.

5.2 Expert interviews

Following the within-subjects study, to gather additional feedback to enhance the generalization of our findings, we conducted think-aloud studies and semi-structured interviews with six teachers.

5.2.1 Participants. The six teachers (female=2, male=4) include five novice teachers with teaching experience ranging from 1 to 3 years ($M = 2$), and one experienced teacher with 15 years of teaching experience. Participants E1-E3, who have been involved in the formative study, teach at the university level. E4 and E5 are primary school teachers, and E6 teaches at a senior high school. Detailed information about their teaching subjects in the current year (the same as what topics they choose to test on our system) are presented in Table 3.

5.2.2 Method. We conduct a 60-minute interview for every expert. Initially, we introduce the participants to the background and goal of our research. For E4-E6, we present Gagne's Nine Events of Instruction and confirm their full understanding of the concept. Subsequently, we give them a 10-minute tutorial on how to use *LessonPlanner*. They were given 30 minutes to complete the following tasks, being asked to think aloud. The examples of expert's interactions with *LessonPlanner* is provided in supplementary material 3.

- **Small Task 1.** Generate lesson goals and the outline and set up instructional events.
- **Small Task 2.** Refine a section. Add an example of a question generated by LLMs.
- **Exploration.** Explore and refine any areas that they are interested in.

Finally, we conduct a 15-minute semi-structured interview, the fixed questions are shown below.

RQ1: Lesson planning outcome.

- Please rate the quality of lesson plans created by *LessonPlanner*, compared to those you prepare during your usual planning routine.
- Can *LessonPlanner* help you manage the teaching process effectively?

RQ2: Lesson Planning Process.

- Does our system impose additional cognitive load on you, such as memory or thought burden, compared to preparing lesson plans without it?

RQ3: Perception of LessonPlanner.

- Is *LessonPlanner* easy to handle? Can you use *LessonPlanner* smoothly without any technical support?
- Which feature (including goal-setting, outline generating, pre-set activities, and output refinement) provided by *LessonPlanner* help you most?
- Please provide some suggestions for *LessonPlanner*.

Table 3: Experts involved in user study.

ID	Gender	Educational Career length	Educational Level	Subject(s)
E1	F	15 years	Senior High School	English
E2	M	3 years	Elementary School	Mathematics
E3	M	1 year	Elementary School	Science
E4	F	3 years	University	Korean Intensive Reading
E5	M	2 years	University	Natural Language Processing
E6	M	2 years	University	Convex Optimization & Computer Network

6 ANALYSES AND RESULTS

In this section, we present the quantitative and qualitative results for each research question (RQ). As for the items measured on a 7-point Likert scale about the lesson planning outcome (RQ1), lesson planning process (RQ2), and perceptions with the system (RQ3), we employ the Wilcoxon signed-rank tests [68] compare the differences between the *LessonPlanner* and baseline condition in the within-subjects study. For the qualitative data in the within-subjects study and expert interviews, two authors transcribe the audio into text scripts and conduct a thematic analysis on these scripts. They first familiarize themselves by reviewing all the text scripts independently. After several rounds of coding with comparison and discussion, they finalize the codes of all the interview data. We count the occurrences of codes and incorporate these qualitative findings in the following presentation of our results.

6.1 Lesson Planning Outcome (RQ1)

Figure 7 shows the quality of outcome lesson plans from the within-subjects study, and the lesson plan designed by P5 is provided in supplementary material 2. Overall, the lesson plans created using *LessonPlanner* ($Mean = 32.25, SD = 5.43$) are rated significantly better than those developed with the baseline system ($M = 26.33, SD = 5.588$) in terms of the total scores; $W = 10.5, p = 0.02$. Regarding each aspect of the outcome lesson plans, *LessonPlanner* ($M = 5.42, SD = 1.037$) does not show a significant advantage over the baseline system ($M = 4.92, SD = 1.256$) regarding aligning the teaching content to the objectives; $W = 15.5, p = 0.40$. However, compared with the baseline condition, the lesson plans from the *LessonPlanner* conditions have a tendency to perform better in facilitating students' skill acquisition (*LessonPlanner*: $M = 5.42, SD = 1.04$; baseline: $M = 4.58, SD = 0.86$; $W = 7, p = 0.058$) and incorporating effective teaching materials (*LessonPlanner*: $M = 5.33, SD = 1.03$; baseline: $M = 4.50, SD = 1.26$; $W = 13.50, p = 0.070$). Lesson plans constructed with *LessonPlanner* are significantly better in terms of balancing the theoretical and practical content (*LessonPlanner*: $M = 5.08, SD = 0.86$; baseline: $M = 4.08, SD = 0.86$; $W = 8, p = 0.039$), using effective teaching strategies (*LessonPlanner*: $M = 5.58, SD = 0.86$; baseline: $M = 4.42, SD = 1.04$; $W = 7, p = 0.034$), and providing sufficient details on the teaching materials (*LessonPlanner*: $M = 5.42, SD = 1.19$; baseline: $M = 3.83, SD = 1.14$; $W = 3, p = 0.007$). Our participants in the within-subjects study and teachers in the expert interview share how *LessonPlanner* affect their lesson planning outcome, which is summarized below.

6.1.1 The Impact of Structured Outlines on Overall Lesson Plan Quality. Four participants (P1, P2, P6, P12) mentioned that the structured outlines helped improve the quality of their lesson plans. As the generated outlines take into account various lesson goals and Gagne's Nine Events, "*LessonPlanner made my lesson plan more standardized*" (P6), and "*it has a more reasonable instructional structure, compared to most resources I found on the Internet*" (P12). Furthermore, the block-based structure strengthens users' grasp of the overall lesson plans. "*It (LessonPlanner) makes me more clear about what should I do in the different stages of this lesson*" (E5). However, E3 offered a different perspective on the impact of the structured outline. "*When I prepare lessons, I already have a clear framework of this lesson (elementary science) in mind. It is synthesized from many lesson plans I've read. Compared to my own framework, I personally find the generated one less effective*" (E3). It reveals that the generated lesson plan outline in *LessonPlanner*, which is based on Gagne's Nine Events, may be less effective in specialized subject domains.

6.1.2 Experts' Opinions on the Impact of Event Hints on the Teaching Content. The teachers in the expert interviews confirmed that the tags that give hints of Gagne's Nine Events in *LessonPlanner* can guide them to prepare a lesson plan with more effective teaching content. E6 mentioned, "*without the system, I'd always forget to put some important content in the lesson plans. The hints act like reminders*". E1 also affirmed this point, "*(Without LessonPlanner,) I wouldn't keep so many tags (instructional events) in my mind. These constantly visible tags help me think and explore the possibilities of conducting my class from various perspectives*". However, not all experts think the Events Hints generated in the outlines are useful. E2 and E3 complained that some hints were useless for them. E2 said, "*Elementary students have limited comprehension, and a single math lesson is impossible to encompass all nine events. I have to adjust the events in each part to fit my need*".

In all, we find that *LessonPlanner* significantly improves participants' quality of lesson plans in the within-subjects study. Participants and teachers generally value the structured lesson plan outline and hints of Gagne's Nine Events. Nevertheless, these events may not be always necessary if teachers have their own lesson plan structures and if they want to focus on specific events in a lesson.

6.2 Lesson Planning Process (RQ2)

Figure 8 shows the statistical results of the participants' perceived task workload in the lesson planning process with *LessonPlanner* and the baseline system. On average, participants perceived significantly less mental demand (*LessonPlanner*: $M = 2.75, SD =$

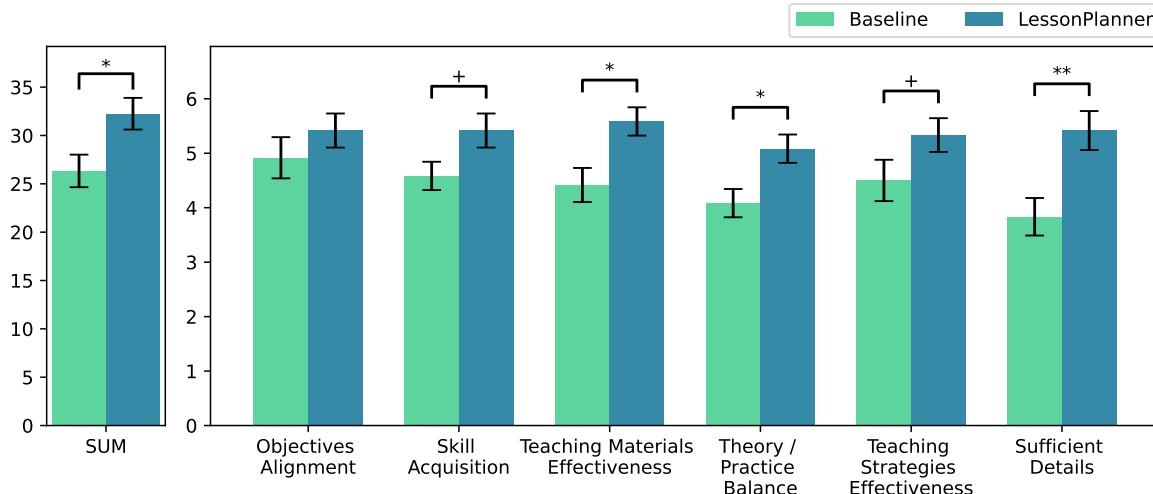


Figure 7: RQ1 results regarding the lesson plan outcome evaluated by experienced teachers in six different aspects and total scores. ***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$, +: $p < 0.1$.

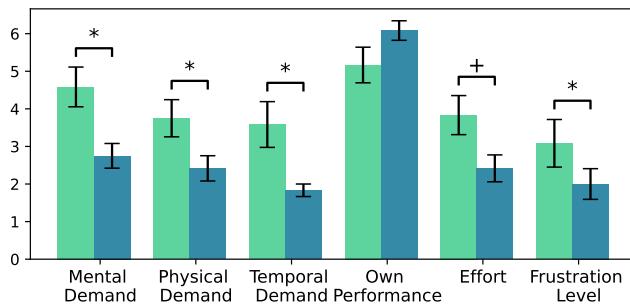


Figure 8: RQ2 results regarding the test of task load. ***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$, +: $p < 0.1$.

1.09; baseline: $M = 4.58, SD = 1.75; W = 7.00, p = 0.035$), physical demand (*LessonPlanner*: $M = 2.42, SD = 1.11$; baseline: $M = 3.75, SD = 1.64; W = 2, p = 0.024$), temporal demand (*LessonPlanner*: $M = 1.83, SD = 0.55$; baseline: $M = 3.58, SD = 2.02; W = 1, p = 0.027$), and frustration (*LessonPlanner*: $M = 2, SD = 1.35$; baseline: $M = 3.08, SD = 2.10; W = 6, p = 0.046$) when they prepare lesson plans with *LessonPlanner* than with the baseline system. Besides, participants tend to perceive less effort spent in the lesson planning process when they are with *LessonPlanner* ($M = 2.42, SD = 1.19$) than with the baseline system ($M = 3.83, SD = 1.72$); $W = 12.5, p = 0.066$. There is no significant difference regarding their perceived task performance in lesson planning between the *LessonPlanner* ($M = 6.08, SD = 0.86$) and baseline conditions ($M = 5.17, SD = 1.57$); $W = 14.5, p = 0.175$. Nine participants believe that utilizing our system requires less workload compared to the baseline in the interview after two tasks. For example, P12 said “Using ChatGPT can be quite skill-intensive, and often the first response you receive may not align with your expectations.

LessonPlanner’s first response is of high quality, which significantly eases the mental and physical demand on me.”

6.2.1 Experts’ Opinions on the Workload of Preparing Lesson Plans with LessonPlanner. Our within-subjects user study showcases that *LessonPlanner* can reduce task workload compared to the baseline interface. In the expert interviews, we are more concerned about what brings workload to teachers when planning lessons with *LessonPlanner*. We found that the workload mainly arises from the selection, refinement, and verification of LLMs’ outputs. First, E4 and E6 indicated that the process of selecting output imposed an additional burden. Interestingly, they both reported that this burden was beneficial, as reported by E6, “The system offers varied content, allowing me to select anything I am satisfied with. This process naturally demands more effort. Before, I could only stick to one basic idea”. Second, E3 and E4 noted that they invested the majority of their time in ensuring that the system accurately produced a small segment of the content, such as formulas, a suitable example, and so on. E3 complained that “When I realized that the results provided by this generative model were not what I was looking for, I kept engaging in conversing with it and iterating the generated results. Clearly, it takes up more time”. Third, the experts emphasized that checking the accuracy of the content (E4, E5) and making sure it was appropriate for the student’s current learning stage (E2, E3) also contributed to their workload. For example, E2 stated, “In this example, it (*LessonPlanner*) uses the area calculation formula for triangles as an introduction, asking students to think about the formula for the area of parallelograms. Yet, the students don’t know how to calculate the area of triangles because it is the content in the next chapter... Having to check every output makes me drained.”.

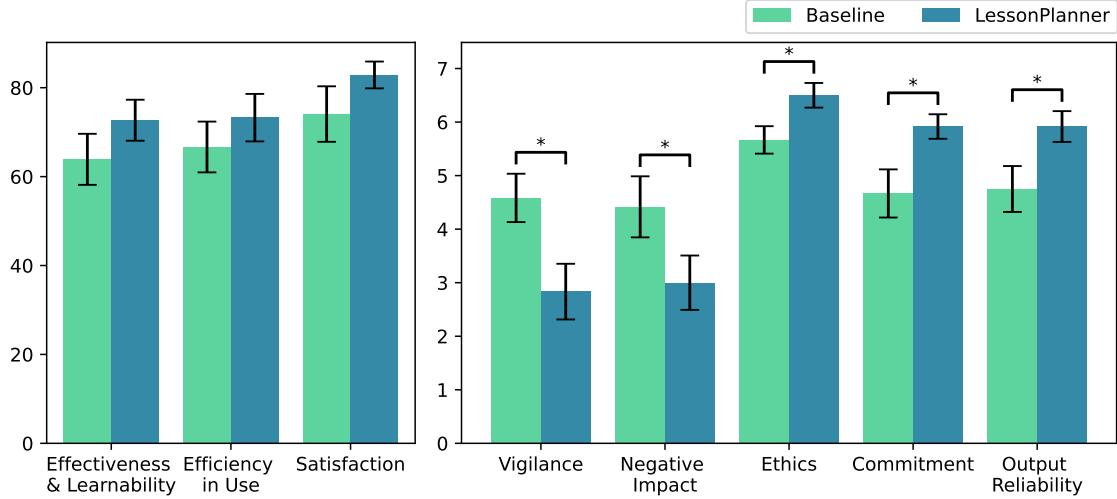


Figure 9: RQ3 results regarding the test of system usability and participants' trust in *LessonPlanner*. *: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$, +: $p < 0.1$.**

6.3 Perception of *LessonPlanner* (RQ3)

Figure 9 shows the participants' ratings on the usability of and their trust on *LessonPlanner* and the baseline system in the within-subjects user study. Overall, participants gave a higher score for *LessonPlanner* than that for the baseline system regarding the system's Effectiveness & Learnability (*LessonPlanner*: $M = 72.70$, $SD = 15.97$; baseline: $M = 63.90$, $SD = 19.90$; $W = 21$, $p = 0.286$), Efficiency in Use (*LessonPlanner*: $M = 73.30$, $SD = 18.50$; baseline: $M = 66.70$, $SD = 19.78$; $W = 27$, $p = 0.380$), and Satisfaction (*LessonPlanner*: $M = 82.90$, $SD = 10.45$; baseline: $M = 74.10$, $SD = 21.62$; $W = 20$, $p = 0.246$), but the differences are not significant. As for the trust in the system, compared to the baseline system, participants generally have significantly fewer concerns regarding the *LessonPlanner*'s vigilance (*LessonPlanner*: $M = 2.83$, $SD = 1.72$; baseline: $M = 4.58$, $SD = 1.50$; $W = 4$, $p = 0.028$) and potential negative impact on teaching (*LessonPlanner*: $M = 3.00$, $SD = 1.68$; baseline: $M = 4.42$, $SD = 1.90$; $W = 4$, $p = 0.048$). Participants also have a significantly higher level of trust on *LessonPlanner*'s ethical standards (*LessonPlanner*: $M = 6.50$, $SD = 0.76$; baseline: $M = 5.67$, $SD = 0.85$; $W = 3.5$, $p = 0.020$), commitment to user (*LessonPlanner*: $M = 5.92$, $SD = 0.76$; baseline: $M = 4.67$, $SD = 1.49$; $W = 3.5$, $p = 0.012$), and reliability of outputs (*LessonPlanner*: $M = 5.91$, $SD = 0.95$; baseline: $M = 4.75$, $SD = 1.42$; $W = 2$, $p = 0.024$). Participants and teachers actively comment on the usability of *LessonPlanner* and their trust or concerns about its generated content in the interviews, which we summarize below.

6.3.1 Experts' Gained Insights from *LessonPlanner*. Our teachers in the expert interviews appreciated that *LessonPlanner* can provide inspiring content for constructing lesson plans. Specifically, all teachers agreed that the "Generate an Example in Detailed" feature is well-designed and useful, as it provides them with new perspectives for lesson design. For instance, E2 mentioned, "Sometimes I

come up with an example but I am uncertain if it's appropriate. I am also often not sure from which angle I should present the example to the students. I can get other examples from the system as a reference to my example and get guidance on how to present it". Moreover, when using "Construct multiple choice or fill-in-the-blank questions" or "Construct group discussion topics" in their small tasks of lesson planning, four experts expressed that the system output was "very inspiring". For example, E6 stated, "I had never considered incorporating discussion questions into the (Computer Networks) course, but now it has generated a great topic I think. Discussing this topic is beneficial for students, actually". These qualitative findings once again confirm that LLMs are competent tools for promoting users' creativity [21]. Novice teachers greatly benefit from the insights generated by LLMs and use them to broaden their thinking.

6.3.2 Diverse Attitudes to the Reliability of *LessonPlanner*. Eight participants in the user study indicated that they were more confident with the reliability of the generated content in *LessonPlanner* compared to that in the baseline. For example, P9 and P12 accounted for their increased confidence in the preset prompts and structured output, which make *LessonPlanner* seem professional and reliable. In contrast, one participant, P3, doubted the preset prompts, "I would only trust the system if I can access all the details of the prompts".

Teachers in the expert interviews have different opinions on the reliability of the LLM's output. For example, the most frequently debated issue was the credibility of the "Generate definition" activity. E5, who teaches Computer Network and Convex Optimization at the university level, believed that the generated definitions were accurate and could be directly utilized in his lessons. Conversely, E2 and E3, who are instructors in elementary school, contended that the definitions did not correspond with the student's current stage of learning, making them nearly impracticable for the classroom. This conflicting view could be due to that our *LessonPlanner* was

primarily designed for supporting teachers in university in our study, but it reveals the need to tailor the generated content to the knowledge level of targeted students, which we will discuss in the Discussion section.

Another interesting finding is that users' trust in *LessonPlanner* could be affected by the level of their familiarity with LLMs. Participants who interact with LLMs daily or weekly ($M = 78.0, SD = 17.20$) generally show a higher degree of trust in the system than those who use it less frequently ($M = 68.3, SD = 14.55$). Additionally, in the expert interviews, teachers (E2, E4) who do not know how LLMs work and rarely used LLMs before, expressed less satisfaction with *LessonPlanner*, which could be due to their *inappropriate trust* [54]. For instance, E4 stated, "*The system needs to give clear sources for its generated examples, such as from newspaper. I would not count on an example without credible sources, because it might be misleading for students.*"

6.3.3 Room for Improving *LessonPlanner*'s Usability. Eight participants in the user study suggested that our system had room for improved usability. First, the embedded editor was considered "not so intelligent" by four participants (P2, P4, P8, and P10). For example, P2 mentioned, "*I usually use shortcuts, but here, I have to use the mouse to click. It is inconvenient.*" Second, *LessonPlanner* interaction design may not match the routine way to use LLM. P1 and P9, who are familiar with the ChatGPT web app, struggled with our system's conversation design. "*I am used to scrolling up to see what it (LLM) responded to before, but it does not work in this system. I miss that the way I do all the time with ChatGPT*" (P1).

In the expert interviews, we further inquired about the learnability of *LessonPlanner*. All teachers believed that the system was easy to use, but they reminded that it should have a clear tutorial to walk through the system before using it. Without a tutorial, they might be confused about the system's functions. For example, E2 and E5 claimed that they were confused by the labels on buttons without detailed descriptions of their functions, input, and output. E5 suggested that "*when a user enters this system for the first time, providing a new user tutorial or a product tour could be helpful, as many websites do*".

7 DISCUSSION

In this work, we design and develop *LessonPlanner*, an interactive system that facilitates novice teachers to prepare lesson plans with large language models (LLMs). Our within-subjects study with 12 participants compared to a ChatGPT interface and our expert interviews with six teachers demonstrate the usefulness of *LessonPlanner* in improving the outcome and reducing the task load of preparing lesson plans. Our user study findings also highlight the imitations, and opportunities of *LessonPlanner*, which we summarize as design considerations for future interactive systems that support teachers with LLMs.

7.1 Design Considerations

7.1.1 Base the generated content on reliable knowledge databases. Both participants and experts acknowledged the remarkable generative capabilities of LLMs for supporting lesson-planning tasks. *LessonPlanner* can assist teachers in producing various types of insightful and creative teaching materials and suggesting strategies to deliver these materials. Nevertheless, the reliability of the LLM used in *LessonPlanner* was still questioned by some participants and experts (E2-E5), primarily because of the hallucination of LLMs [74] and the limitation of not being able to automatically access web resources. Furthermore, P12 and E3 expressed a continued preference for utilizing search engines to search for materials, though LLMs are equipped in *LessonPlanner*. On the one hand, the generated content may not match different educational levels, for example, including knowledge that elementary students have not yet grasped in previous lessons (E2). This problem stems from teachers providing only a brief description of prior knowledge in a single line on the Goal-setting page (Figure 2), which is insufficient. To improve the alignment of the generated content with the students' educational levels, future work could construct a detailed knowledge graph of courses at different levels and recommend prior knowledge of current course topics to guide the *LessonPlanner*. On the other hand, the preference stemmed from their belief that despite the unstructured results returned from search engines, the online materials published by other teachers are more professional, compared to those generated by LLMs. Hence, content generated by LLMs should not constitute the entire lesson plan but rather serve as a supplementary resource to complement the lesson plan. This principle is fundamental in our motivation, design, and implementation of *LessonPlanner*. To maximize the potential of traditional resources, web resources, and generated content within the system, we suggest that future lesson planning systems should integrate reliable knowledge databases, either prepared by teachers or curated from web resources, into the prompts to LLMs. There are already many ways that help us do this, such as LangChain⁴, Microsoft Copilot, and the recent advances in Retrieval-Augmented Generation techniques [14, 41]. With the assistance of external resources, teachers can enhance their confidence in the system and decrease the workload they spend on verifying the generated content.

7.1.2 Allow flexible customization of prompts to LLMs. When designing and developing *LessonPlanner*, we spend a lot of effort in providing flexible user control (DG4) to the generated content, e.g., users are able to alter or cooperate with LLMs to refine the lesson goals in the Goal-Setting page, and users can freely modify all generated content of the lesson plan in the Lesson-Planning page. However, users hold elevated expectations regarding the customization of the interactive features with LLMs. For example, participants expressed a desire to see and adjust our built-in prompts of the default buttons, which would make the generated content align more closely with their expectations. Teachers in the expert interviews further highlighted that Instructional Events used in the LLM prompt for initializing the lesson plan may need to be adjustable by users as well. Potential solutions to address these expectations are incorporating additional predefined teaching scenarios (e.g., a lesson after midterm exams) offered by developers and enabling users

⁴<https://github.com/langchain-ai/langchain>

to customize the system's predefined prompts. Another approach is to automate the prompting process [43] by utilizing course metadata to provide different predefined functionalities tailored to users.

7.1.3 Integrate multi-modal generated content. The findings of our study demonstrate that planning with *LessonPlanner* leads to a notable enhancement in the quality of lesson plans and a reduction in teachers' workload, which can be attributed to the abundant textual content produced and structured by the LLM. An observation identified in the expert interviews is that *LessonPlanner* demonstrates its interest in using multi-modal materials to teach particular knowledge points. For instance, it attempts to draw a binary tree through a combination of text and symbols or directs the teacher to present a video on specific topics or display images in the classroom. The aforementioned phenomenon reminds us that integrating other multi-modal models, such as text-to-image generation models [73] and text-to-video generation models [20], with LLMs might make the content of lesson plans richer and save time by reducing the need to gather supplementary materials from the web. These two points were further confirmed by the experts in the interviews, where they hoped to insert generated images into the lesson plans.

7.2 Generality

In the within-subjects study, *LessonPlanner* demonstrated strong performance, as no participants raised concerns about the suitability of the content delivered by LLM for students at their current stage. The teachers of different educational levels in the expert interviews have expressed overall satisfaction with the system's performance, despite that the examples in the prompt templates of *LessonPlanner* are about the subjects of computer science for university students. This suggests that our system has the potential to be adapted to various educational levels and subjects. However, our teachers reported that they occasionally got generated definitions or examples unaligned with the students' knowledge background. To mitigate this issue, two feasible solutions can be considered. First, the simplest and most practical approach could be encouraging users to customize examples or definitions as templates and embed them into the original prompts. This will guide LLMs to generate content that is aligned with the provided templates. Second, the Chain-of-Thought (CoT) [66] technique could be employed in prompt engineering, instead of providing fixed examples (e.g., data structures) which might lead to content that is inappropriate for the educational level or subject. For example, the prompt shown in subsubsection 4.3.3 could be edited as below.

- Q1: I will instruct the course of {lesson name} for students in {students' educational level}. Please provide the names of three key concepts that students may need to learn for this course.
- Q2 (after receiving the response to Q1 from the LLMs): Provide the specific definition of the first concept that is suitable for the {students' educational level}. The response must conform to the following format.

Input:

Context: the name of the concept

Output:**the name of the concept ** -Definition: The definition of the concept.

Moreover, *LessonPlanner* serves not only as an assistant in preparing lesson plans for courses but also as a platform helping novice teachers promote their critical thinking and delve deeper into the subjects. It assists educators in gaining the knowledge of developing a lesson based on educational theory in a systematic way, thus improving their pedagogical abilities.

7.3 Limitations and Future Work

Our work has several limitations that call for future work. First, all participants in our within-subjects study majored in subjects related to computer science, which helps to control the topics of planned lessons in the study. While we qualitatively evaluate *LessonPlanner* with six experts teaching different subjects at various educational levels, we lack quantitative findings on *LessonPlanner*'s effectiveness in facilitating users with diverse backgrounds in planning lessons on other subjects. Second, *LessonPlanner*, which includes detailed instructional events and delivery methods, is presently oriented towards assisting novice teachers. In the future, we will evaluate and customize *LessonPlanner* with teachers with diverse backgrounds and teaching experience. Third, while the output markdown file of the lesson plans from *LessonPlanner* can serve as the intermediate materials to deliver a course, teachers usually need to prepare other digital materials (e.g., slides) upon the lesson plans in their teaching practices. Future researchers can explore the application of deep learning techniques to generate slides [26, 75] based on the output lesson plans from *LessonPlanner* or support users to prepare a lesson in Powerpoint or Google slides embedded with LLMs. Fourth, we invite a teacher who is experienced with the course topics to evaluate the outcome of lesson plans in the within-subjects study. However, a more ideal way to assess the effectiveness of a lesson plan would be to enact the plan in a real-world lesson and collect feedback from students and teachers. In the future, we plan to invite teachers to use *LessonPlanner* to prepare for the lessons they are going to teach and evaluate *LessonPlanner*'s effectiveness after the outcome lesson plans are enacted in a course. Recordings of the lessons and post-lesson interviews with the teachers will also be collected and analyzed to ensure *LessonPlanner*'s practical usability and benefits for educators. Last but not least, to generate an initial lesson plan, we included all nine events in the prompt. While nine events are generally helpful, not all events are applicable to one specific lesson. Future iteration of *LessonPlanner* should allow the users to specify the events they would like to incorporate into their course before prompting the LLMs to generate a lesson plan.

8 CONCLUSION

In this paper, we designed and developed an interactive system, *LessonPlanner*, to support novice teachers in interactively constructing lesson plans using LLM-generated content grounded on Gagné's nine events. *LessonPlanner* is capable of initializing lesson goals and outlines, generating activities and materials that adapt to planned teaching events, and offering flexible user control to customize the lesson plan and query the LLM. The within-subjects experiment

with 12 participants demonstrates that *LessonPlanner* leads to an enhancement in the quality of the lesson plan outcome and eases their workload during the preparation process. We further conducted expert interviews with six teachers who highlighted the potential of utilizing it to support novice teachers at various educational levels and in diverse subjects. We discuss design considerations and insights derived from the user study for leveraging generative models to support lesson-planning tasks.

9 ACKNOWLEDGEMENT

This work is supported by the Young Scientists Fund of the National Natural Science Foundation of China (NSFC) with Grant No.: 62202509, NSFC Grant No.: U22B2060, and the General Projects Fund of the Natural Science Foundation of Guangdong Province in China with Grant No. 2024A1515012226.

REFERENCES

- [1] R Abdelghani, YH Wang, X Yuan, T Wang, H Sauzéon, and PY Oudeyer. 2022. GPT-3-driven pedagogical agents for training children's curious question-asking skills. *ArXiv preprint arXiv 2211 (2022)*.
- [2] Mahmoud Abdulwahed and Zoltan K Nagy. 2009. Applying Kolb's experiential learning cycle for laboratory education. *Journal of engineering education* 98, 3 (2009), 283–294.
- [3] Ashraf Alam. 2023. Intelligence unleashed: An argument for AI-enabled learning ecologies with real world examples of today and a peek into the future. In *AIP Conference Proceedings*, Vol. 2717. AIP Publishing.
- [4] Alages Andre. 2011. The iLessonPlan: a lesson planning tool for the 21 st century. In *Proceedings of the Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education, ASCILITE 2011*. 93–105.
- [5] Shamsa Aziz, Munazza Mahmood, and Zahra Rehman. 2018. Implementation of CIIPP Model for Quality Evaluation at School Level: A Case Study. *Journal of Education and Educational Development* 5, 1 (2018), 189–206.
- [6] Anna L Ball, Neil A Knobloch, and Sue Hoop. 2007. The instructional planning experiences of beginning teachers. *Journal of Agricultural Education* 48, 2 (2007), 56–65.
- [7] Jan Philip Bernius, Stephan Krusche, and Bernd Bruegge. 2022. Machine learning based feedback on textual student answers in large courses. *Computers and Education: Artificial Intelligence* 3 (2022), 100081.
- [8] Shravya Bhat, Huy A Nguyen, Steven Moore, John Stamper, Majd Sakr, and Eric Nyberg. 2022. Towards automated generation and evaluation of questions in educational domains. In *Proceedings of the 15th international conference on educational data mining*, Vol. 701.
- [9] Mary Kalantzis Bill Cope and Duane Searsmith. 2021. Artificial intelligence for education: Knowledge and its assessment in AI-enabled learning ecologies. *Educational Philosophy and Theory* 53, 12 (2021), 1229–1245. <https://doi.org/10.1080/00131857.2020.1728732>
- [10] Sairavi Kiran Biri, Subir Kumar, Muralidhar Panigrahi, Shaikat Mondal, Jishil Kumar Behera, Himel Mondal, and Jishil K Behera IV. 2023. Assessing the Utilization of Large Language Models in Medical Education: Insights From Undergraduate Medical Students. *Cureus* 15, 10 (2023).
- [11] Euan Bonner, Ryan Lege, and Erin Frazier. 2023. Large Language Model-Based Artificial Intelligence in the Language Classroom: Practical Ideas for Teaching. *Teaching English with Technology* 23, 1 (2023), 23–41.
- [12] Virginia Braun and Victoria Clarke. 2012. *Thematic analysis*. American Psychological Association.
- [13] John Brooke. 2013. SUS: a retrospective. *Journal of usability studies* 8, 2 (2013), 29–40.
- [14] Deng Cai, Yan Wang, Lemao Liu, and Shuming Shi. 2022. Recent advances in retrieval-augmented text generation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*. 3417–3419.
- [15] Brendan Calandra, Laurie Brantley-Dias, and Kezia McNeal. 2007. An electronic system to support novice teachers' reflective lesson design. *Multicultural Education & Technology Journal* 1, 2 (2007), 100–111.
- [16] Jerry CK Chan, Yaowei Wang, Qing Li, George Baciu, Jiannong Cao, Xiao Huang, Richard Chen Li, and Peter HF Ng. 2022. Intelligent instructional design via interactive knowledge graph editing. In *International Conference on Web-Based Learning*. Springer, 41–52.
- [17] Hung Chau, Jordan Barria-Pineda, and Peter Brusilovsky. 2017. Content wizard: concept-based recommender system for instructors of programming courses. In *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*. 135–140.
- [18] Qiaoyi Chen, Siyu Liu, Kaihui Huang, Xingbo Wang, Xiaojuan Ma, Junkai Zhu, and Zhenhui Peng. 2024. RetAssist: Facilitating Vocabulary Learners with Generative Images in Story Retelling Practices. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference*. 2019–2036.
- [19] Chun-Wei Chiang, Zhuoran Lu, Zhuoyan Li, and Ming Yin. 2024. Enhancing AI-Assisted Group Decision Making through LLM-Powered Devil's Advocate. (2024).
- [20] Joseph Cho, Fachrina Dewi Puspitasari, Sheng Zheng, Jingyao Zheng, Lik-Hang Lee, Tae-Ho Kim, Choong Seon Hong, and Chaoning Zhang. 2024. Sora as an AGI World Model? A Complete Survey on Text-to-Video Generation. *arXiv preprint arXiv:2403.05131* (2024).
- [21] Geoff Davis, Mick Grierson, et al. 2022. Investigating attitudes of professional writers to GPT text generation: AI based creative support tools. (2022).
- [22] Paul Denny, Sumit Gulwani, Neil T Heffernan, Tanja Käser, Steven Moore, Anna N Rafferty, and Adish Singla. 2024. Generative AI for Education (GAIED): Advances, Opportunities, and Challenges. *arXiv preprint arXiv:2402.01580* (2024).
- [23] Samantha L Dias-Lacy and Ruth V Guirguis. 2017. Challenges for New Teachers and Ways of Coping with Them. *Journal of Education and Learning* 6, 3 (2017), 265–272.
- [24] Christopher B Divito, Bryan M Katchikian, Jenna E Gruenwald, and Jennifer M Burgoon. 2024. The tools of the future are the challenges of today: The use of ChatGPT in problem-based learning medical education. *Medical Teacher* 46, 3 (2024), 320–322.
- [25] Mary Forehand. 2010. Bloom's taxonomy. *Emerging perspectives on learning, teaching, and technology* 41, 4 (2010), 47–56.
- [26] Tsu-Jui Fu, William Yang Wang, Daniel McDuff, and Yale Song. 2022. Doc2ppt: Automatic presentation slides generation from scientific documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 634–642.
- [27] Ebrahim Gabajiwala, Priyat Mehta, Ritik Singh, and Reeta Koshy. 2022. Quiz maker: Automatic quiz generation from text using NLP. In *Futuristic Trends in Networks and Computing Technologies: Select Proceedings of Fourth International Conference on FTNCT 2021*. Springer, 523–533.
- [28] Robert Gagne. 1985. The conditions of learning and theory of instruction Robert Gagné. *New York, NY: Holt, Rinehart ja Winston* (1985).
- [29] Qiushi Han, Haitong Chen, Haoxiang Fan, and Zhenhui Peng. 2023. ReDBot: Exploring Conversational Recommendation for Decision-Making Support in Group Chats. In *Proceedings of the Eleventh International Symposium of Chinese CHI*. 73–80.
- [30] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.
- [31] Kenneth H Hoover and Paul M Hollingsworth. 1970. Learning and teaching in the elementary school. (*No Title*) (1970).
- [32] Christina L Jacobs, Sonya N Martin, and Tracey C Otieno. 2008. A science lesson plan analysis instrument for formative and summative program evaluation of a teacher education program. *Science education* 92, 6 (2008), 1096–1126.
- [33] Maurice Jakesch, Advait Bhat, Daniel Buschek, Lior Zalmanson, and Mor Naaman. 2023. Co-writing with opinionated language models affects users' views. In *Proceedings of the 2023 CHI conference on human factors in computing systems*. 1–15.
- [34] Jun-Yin Jian, Ann M Bisantz, and Colin G Drury. 2000. Foundations for an empirically determined scale of trust in automated systems. *International journal of cognitive ergonomics* 4, 1 (2000), 53–71.
- [35] Peiling Jiang, Jude Rayan, Steven P Dow, and Haijun Xia. 2023. Graphologue: Exploring large language model responses with interactive diagrams. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–20.
- [36] Sushma N Jogan. 2019. An Effective 5 E Lesson Plan in Teaching Prose: A Model. *Online Submission* 6, 50 (2019), 11999–12009.
- [37] Kayvan Khadjooi, Kamran Rostami, and Sauid Ishaq. 2011. How to use Gagne's model of instructional design in teaching psychomotor skills. *Gastroenterology and hepatology from bed to bench* 4, 3 (2011), 116.
- [38] Osama Koraishi. 2023. Teaching English in the age of AI: Embracing ChatGPT to optimize EFL materials and assessment. *Language Education and Technology* 3, 1 (2023).
- [39] Tiffany H Kung, Morgan Cheatham, Arielle Medenilla, Czarina Sillos, Lorie De Leon, Camille Elepaño, Maria Madriaga, Rimel Aggabao, Giezel Diaz-Candido, James Maningo, et al. 2023. Performance of ChatGPT on USMLE: potential for AI-assisted medical education using large language models. *PLoS digital health* 2, 2 (2023), e0000198.
- [40] Mina Lee, Percy Liang, and Qian Yang. 2022. Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities. In *Proceedings of the 2022 CHI conference on human factors in computing systems*. 1–19.
- [41] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel,

- et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.
- [42] Michael Xieyang Liu, Advait Sarkar, Carina Negreanu, Benjamin Zorn, Jack Williams, Neil Toronto, and Andrew D Gordon. 2023. “What it wants me to say”: Bridging the abstraction gap between end-user programmers and code-generating large language models. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–31.
- [43] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Comput. Surveys* 55, 9 (2023), 1–35.
- [44] Piotr Mirowski, Kory W Mathewson, Jaylen Pittman, and Richard Evans. 2023. Co-writing screenplays and theatre scripts with language models: Evaluation by industry professionals. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–34.
- [45] Chanchikar Mitra, Mihran Miroyan, Rishi Jain, Vedant Kumud, Gireeja Ronade, and Narges Norouzi. 2024. RetLLM-E: Retrieval-Prompt Strategy for Question-Answering on Student Discussion Forums. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 23215–23223.
- [46] Reza Hadi Mogavi, Chao Deng, Justin Juho Kim, Pengyuan Zhou, Young D Kwon, Ahmed Hosny Saleh Metwally, Ahmed Tlili, Simone Bassanelli, Antonio Bucciarone, Sujit Gujar, et al. 2023. Exploring user perspectives on chatgpt: Applications, perceptions, and implications for ai-integrated education. *arXiv preprint arXiv:2305.13114* (2023).
- [47] Reza Hadi Mogavi, Chao Deng, Justin Juho Kim, Pengyuan Zhou, Young D Kwon, Ahmed Hosny Saleh Metwally, Ahmed Tlili, Simone Bassanelli, Antonio Bucciarone, Sujit Gujar, et al. 2024. ChatGPT in education: A blessing or a curse? A qualitative study exploring early adopters’ utilization and perceptions. *Computers in Human Behavior: Artificial Humans* 2, 1 (2024), 100027.
- [48] Ethan R Mollick and Lilach Mollick. 2023. Using AI to implement effective teaching strategies in classrooms: Five strategies, including prompts. *Including Prompts* (March 17, 2023) (2023).
- [49] Ali Jamali Nesari and Mina Heidari. 2014. The important role of lesson plan on educational achievement of Iranian EFL teachers’ attitudes. *International Journal of Foreign Language Teaching & Research* 3, 5 (2014), 25–31.
- [50] Hanna-Liisa Pender, Lennart Bohl, Marius Schönberger, and Julia Knopf. 2022. An AI-based lesson planning software to support competence-based learning. In *8th International Conference on Higher Education Advances (HEAD'22)*. Editorial Universitat Politècnica de València, 1033–1041.
- [51] Sida Peng, Eirini Kalliamvakou, Peter Cihon, and Mert Demirer. 2023. The impact of ai on developer productivity: Evidence from github copilot. *arXiv preprint arXiv:2302.06590* (2023).
- [52] Savvas Petridis, Nicholas Diakopoulos, Kevin Crowston, Mark Hansen, Keren Henderson, Stan Jastrzebski, Jeffrey V Nickerson, and Lydia B Chilton. 2023. Anglekindling: Supporting journalistic angle ideation with large language models. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–16.
- [53] I Gede Widiartana Putra, I Wayan Sukra Warpala, I Gde Wawan Sudatha, and Ni Putu Rara Stianing Utami. 2022. Developing a Gagne Theory-Based Learning Video for Thematic Subject in Elementary School. *International Research Journal of Management, IT and Social Sciences* 9, 4 (2022), 666–675.
- [54] Crystal Qian and James Wexler. 2024. Take It, Leave It, or Fix It: Measuring Productivity and Trust in Human-AI Collaboration. *arXiv preprint arXiv:2402.18498* (2024).
- [55] Md Mostafizer Rahman and Yutaka Watanobe. 2023. ChatGPT for education and research: Opportunities, threats, and strategies. *Applied Sciences* 13, 9 (2023), 5783.
- [56] Rishabh Ranawat, Ashwin Venkataraman, and Lakshminarayanan Subramanian. 2021. Collectiveteach: a system to generate and sequence web-annotated lesson plans. In *ACM SIGCAS Conference on Computing and Sustainable Societies*. 1–13.
- [57] Nabil Y Razzouk, Jay N Razzouk, et al. 2008. Analysis In Teaching With Cases: A Revisit To Blooms Taxonomy Of Learning Objectives. *College Teaching Methods & Styles Journal (CTMS)* 4, 1 (2008), 49–56.
- [58] Aslina Saad and Christian Dawson. 2018. Requirement elicitation techniques for an improved case based lesson planning system. *Journal of Systems and Information Technology* 20, 1 (2018), 19–32.
- [59] Amanda G Sawyer and Joy Myers. 2018. Seeking comfort: How and why pre-service teachers use internet resources for lesson planning. *Journal of Early Childhood Teacher Education* 39, 1 (2018), 16–31.
- [60] Jianhao Shen, Yichun Yin, Lin Li, Lifeng Shang, Xin Jiang, Ming Zhang, and Qun Liu. 2021. Generate & rank: A multi-task framework for math word problems. *arXiv preprint arXiv:2109.03034* (2021).
- [61] Hana Stein, Irina Gurevich, and Dvora Gorev. 2020. Integration of technology by novice mathematics teachers—what facilitates such integration and what makes it difficult? *Education and Information Technologies* 25, 1 (2020), 141–161.
- [62] Sven Strickroth. 2019. PLATON: Developing a graphical lesson planning system for prospective teachers. *Education Sciences* 9, 4 (2019), 254.
- [63] Daniel L Stufflebeam. 2000. The CIPP model for evaluation. In *Evaluation models: Viewpoints on educational and human services evaluation*. Springer, 279–317.
- [64] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasamine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).
- [65] Bryan Wang, Gang Li, and Yang Li. 2023. Enabling conversational interaction with mobile ui using large language models. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–17.
- [66] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.
- [67] Judy W Wood and Jennifer W Miederhoff. 1988. Adapting lesson plans for the mainstreamed student. *The Clearing House* 61, 6 (1988), 269–276.
- [68] Robert F Woolson. 2007. Wilcoxon signed-rank test. *Wiley encyclopedia of clinical trials* (2007), 1–3.
- [69] Meng Xia, Qian Zhu, Xingbo Wang, Fei Nie, Huamin Qu, and Xiaojuan Ma. 2022. Persua: A Visual Interactive System to Enhance the Persuasiveness of Arguments in Online Discussion. *Proc. ACM Hum.-Comput. Interact.* 6, CSCW2, Article 319 (nov 2022), 30 pages. <https://doi.org/10.1145/3555210>
- [70] JIAYU YIN, CATHERINE GU, JENNY MAR, SYDNEY ZHANG, and STEVEN P DOW. 2024. Jamplate: Exploring LLM-Enhanced Templates for Idea Reflection. (2024).
- [71] Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. 2022. Wordcraft: story writing with large language models. In *27th International Conference on Intelligent User Interfaces*. 841–852.
- [72] Ismail Md Zain. 2017. The Collaborative Instructional Design System (CIDS): Visualizing the 21st Century Learning. *Universal Journal of Educational Research* 5, 12 (2017), 2259–2266.
- [73] Chenshuang Zhang, Chaoning Zhang, Mengchun Zhang, and In So Kweon. 2023. Text-to-image diffusion model in generative ai: A survey. *arXiv preprint arXiv:2303.07909* (2023).
- [74] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren’s song in the AI ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219* (2023).
- [75] Chengbo Zheng, Dakuo Wang, April Yi Wang, and Xiaojuan Ma. 2022. Telling stories from computational notebooks: Ai-assisted presentation slides creation for presenting data science work. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–20.

A INTERFACES OF LESSONPLANNER HIGH-FIDELITY PROTOTYPE IN FORMATIVE STUDY

The original version used for the actual formative study is in Chinese. We translate all the text in the interface into English completely, as shown in Figure 10- Figure 13.

创建您的课程

Lesson name
Data Structures and Algorithms

Lesson about
Data Structures and Algorithms is a fundamental course in computer science, designed to teach methods for organizing and storing data, as well as designing problem-solving algorithms. Students will learn about various data structures (such as arrays, linked lists, and trees) and algorithms (such as sorting, searching, and graph algorithms). This course develops the ability to analyze and solve real-world problems, forming the cornerstone of programming and software development.

Quick Sort

Please describe the title or main content of your lesson.

Lesson Goals Lesson Contents

Please specify the course objectives or the content you wish to include in the course (optional, keywords only).

Generate

Figure 10: The page of meta-data collection.

Course objective description

The goal of this lesson is to help students understand the core idea of divide and conquer, master the basic principles of quicksort, learn to analyze the complexity of the algorithm, and understand how to implement it in code (C++). Encourage students to apply the algorithm to various daily scenarios.

Generate outline

Outline

- Inform learners of objectives **List of Objectives and content** (Duration: short)
 - Stimulate recall of prior learning **Recall: Bubble sort and merge sort** (Duration: medium)
 - Present stimulus - Provide learner guidance **Divide and Conquer** (Duration: medium)
 - Present stimulus - Provide learner guidance **Merge Sort and Its Implementation** (Duration: medium)
 - Pseudocode Representation of Merge Sort Present the pseudocode for Merge Sort and provide an explanation.
 - Visualization of the Merge Sort Process Provide 2 example problems to visualize the Merge Sort process, making it easier for students to understand.
 - Elicit performance - Provide feedback **Time limited exercises** (Duration: medium)
- Generate the lesson plan**

Figure 11: The page of outline overview.

Quicksort Algorithm

- Question:** What is the basic idea of quicksort?
 - Bubble comparison
 - Divide and conquer
 - Insertion sort
 - Selection sort
- Answer:** b. Divide and conquer
- Question:** In the quicksort algorithm, the choice of pivot has a significant impact on the algorithm's performance. Which of the following descriptions is correct?
 - The choice of pivot does not affect the performance of quicksort
 - The best pivot choice is the first element of the array
 - A good pivot choice can improve sorting efficiency
 - The choice of pivot is only related to the size of the array
- Answer:** c. A good pivot choice can improve sorting efficiency
- Question:** In quicksort, what is the termination condition for the recursive steps?
 - The array has only one element
 - The array is already sorted
 - All elements in the array are the same
 - The array is empty
- Answer:** a. The array has only one element
- Question:** In the quicksort algorithm, the choice of pivot is crucial to sorting performance. Which of the following situations is most likely to lead to the worst-case time complexity of quicksort?
 - The pivot is always the middle element of the array
 - The pivot is always the largest element of the array
 - The pivot is always the first element of the array
 - The pivot is always a random element of the array

Please answer as soon as possible!

Type / evoke more

In the quicksort algorithm, the choice of the pivot is crucial for sorting performance. Which of the following scenarios is most likely to lead to the worst-case time complexity for quicksort?

- The pivot is always the middle element of the array.
- The pivot is always the largest element of the array.
- The pivot is always the first element of the array.
- The pivot is always a random element of the array.

Please respond as soon as possible.

Insert below Continue writing Rewrite and require longer

AI 对应助手快速响应 (prompts)

- provide me with an additional multiple-choice
- suitable for answering quickly
- Fill-in-the-blank question
- additional coding problem

Multiple choice question; suitable for answering quickly.

Figure 12: The editing page with LLM sidebar.

LLM assistant integrated within the editor

Application Cases

Imagine you are a data engineer responsible for sorting large datasets. Sometimes, these datasets can almost fill up all the memory. You need to ensure the space complexity is as low as possible, and in this case, merge sort may not meet the requirements.

Open-ended question

Have you ever needed to quickly determine the rank of a specific element during sorting? For instance, in sorting student grades, do you need to quickly find a particular student's rank? Consider the following:

- What practical applications exist for quickly obtaining an element's position in a sorting algorithm?
- What advantages does quicksort offer over other sorting algorithms for this purpose?

Student Grade Sorting:
Suppose you are a teacher, and student grades need to be sorted by different subjects. Sometimes, you may want to prioritize knowing the ranking of a particular student in the entire class to better understand their academic performance. In this case, the performance advantages of the quicksort algorithm become particularly important.
For example, if a class has the following student grade data:

```
EXPLAINTEXT>
Student A: 85
Student B: 92
Student C: 76
Student D: 88
Student E: 95
Student F: 79
```

LLM assistant

Replace Insert below Continue writing Rewrite and require longer

Figure 13: The editing page with embedded LLM assistant.