Int J Artif Intell Educ
https://doi.org/10.1007/s40593-024-00421-1

ARTICLE

LLM-Based Student Plan Generation for Adaptive Scaffolding in Game-Based Learning Environments

Alex Goslen[1] · **Yeo Jin Kim**[1] · **Jonathan Rowe**[1] · **James Lester**[1]

Accepted: 7 July 2024
© International Artificial Intelligence in Education Society 2024

Abstract

The development of large language models offers new possibilities for enhancing adaptive scaffolding of student learning in game-based learning environments. In this work, we present a novel framework for automatic plan generation that utilizes text-based representations of students' actions within a game-based learning environment, CRYSTAL ISLAND, to inform adaptive scaffolding of student goal setting and planning, which are critical elements of self-regulated learning. Plan generation is the task of automatically generating a set of low-level actions that contribute toward accomplishing a target goal given a sequence of student gameplay and their prior completed goals. We investigate the use of two pre-trained large language models, T5 and GPT-3.5, in the plan generation framework. The models utilize 144 middle school students gameplay data, encompassing a total of 11,610 event sequences, as input. The plans generated by the model are subsequently evaluated against plans crafted by students during gameplay utilizing an in-game planning support tool in CRYSTAL ISLAND. We compare automatically generated plans to students' manually generated in terms of the number of matching low-level actions, the number of actions that match when mapped to higher-level categories of actions, and the distribution of categories of actions within plans. Results indicate that automatically generated plans from both models largely align in terms of the high-level categories of actions that are included, but the generated plans feature fewer low-level actions than students' plans. Plans generated by T5 align more closely with student plans, whereas GPT-3.5, though not following student planning patterns, produces valid plans as well. These findings suggest that LLMs show significant promise for automatically generating plans that can be used to devise run-time adaptive scaffolding for student planning in game-based learning environments.

Keywords Goal setting and planning · Game-based learning environments · Language models · Self-regulated learning

✉ Alex Goslen
amgoslen@ncsu.edu

[1] Computer Science, North Carolina State University, Raleigh 27606, NC, USA

Published online: 17 July 2024

Introduction

Winne and Hadwin's model of self-regulated learning (SRL) refers to goal-driven learning that encompasses the formulation of objectives, the development of plans, and the continuous monitoring and adaptation of goals and plans (Winne & Hadwin, 1998, 2008; Winne, 2018). Providing support for students' SRL processes can greatly assist them in navigating challenging learning tasks, such as solving scientific problems (Dever et al., 2022). Game-based learning environments can engage learners in self-regulated learning processes by facilitating goal setting and planning (Boekaerts & Pekrun, 2015). They can also adapt to individual learner strategies in real time as students solve problems in the game environment. Game-based learning environments that are adaptive in game difficulty have demonstrated positive impact on student motivation and situational interest (Koskinen et al., 2023). Trace data produced from logs of student interactions with game-based learning environments provides a rich source of input for such machine learning models. This data has the potential to enhance student learning experiences by informing adaptive scaffolding tailored to each student's individual needs. However, to construct such machine learning models using classical or deep learning techniques, the input and output of models must be in a numerical format, leading to challenges in interpreting the resulting plans generated from such systems. Additionally, constructing robust models that can be implemented in real-time often requires data from thousands of gameplay sessions, which can be difficult to acquire in educational settings.

Recent developments in large language models (LLMs) have created new opportunities for analyzing student log data by using natural language representations of student learning and problem solving. In general, LLMs are pre-trained on vast quantities of text data, enabling the model to develop a broad understanding of natural language. Subsequent fine-tuning with domain-specific data equips LLMs to address specific, downstream tasks within a target domain. This capability introduces opportunities in adaptive learning, such as generating practice problems, detailed solutions, and comprehensive explanations (Kasneci et al., 2023). In our research, we explore how we can leverage LLMs to generate plans aimed at supporting student SRL processes and, thereby, enhance their ability to solve scientific problems.

The work presented in this paper examines the performance of two large pre-trained language models, T5 and GPT-3.5, in automatically generating problem-solving plans in the context of a game-based learning environment for middle school science education called CRYSTAL ISLAND. We define a plan as a set of actions that contribute toward accomplishing a target goal and that can be enacted in the game environment. We refer to the actions that can be directly enacted in gameplay as low-level actions. Our plan generation framework leverages textual representations of student gameplay data, as well as students' previous completed goals and their current target goal as input. As output, the framework produces a set of low-level actions that form a recommended plan to achieve the target goal. Generated plans from this framework have the potential to inform the creation of adaptive scaffolding to support students' learning strategies and SRL behaviors, by providing assistance to students when constructing plans and helping to facilitate their gameplay. We present an empirical analysis of problem-solving plans that were automatically generated by T5- and GPT-based mod-

els by comparing them to plans students manually created using an in-game planning tool in CRYSTAL ISLAND.

We aim to answer the following research questions in this work:

RQ1: How does the performance of a fully-supervised fine-tuned LLM compare with a prompt-engineered few-shot learning approach for the task of student plan generation?

RQ2: Does the LLM-based student plan generation framework produce valid problem-solving plans that can be enacted in CRYSTAL ISLAND and achieve the target goal set by the student?

RQ3: How effectively do LLM-based plan generation models produce plans that can inform real-time adaptive scaffolding in a game-based learning environment?

To answer these research questions, we conducted a quantitative and qualitative comparison of LLM-generated plans and manually generated plans created by students using the planning support tool in CRYSTAL ISLAND. We examine the alignment of the different types of plans, as well as the relationship between different high-level categories of goals and actions that are included in the generated plans. Lastly, we present examples of generated plans that highlight potential issues in automatic plan generation, and we discuss directions for addressing these issues to improve their suitability for driving adaptive scaffolding in game-based learning environments.

Background

SRL in Online Learning

Planning and goal-setting are crucial elements of SRL in many theories (Pintrich, 2000; Winne, 2018; Winne & Hadwin, 1998, 2008; Zimmerman, 2013). Game-based learning, in particular, requires effective goal setting and plan construction, as students need to understand tasks and construct strategies to navigate the game environment (Plass et al., 2020). However, research has shown that students struggle to employ SRL strategies during challenging learning tasks (Azevedo et al., 2016). Additionally, little work has been done to explore explicit goal setting and planning in online learning contexts (Winne, 2018).

Despite this gap, research has shown that when supported in goal setting and planning, students' learning experiences are improved in online environments. Azevedo et. al (2022) demonstrated enhanced learning outcomes when students were aided in goal-setting through pedagogical agents (Azevedo et al., 2022). Another intelligent logic tutor created to provide strategies to students when developing logic proofs demonstrated performance improvement in students that adopted the given strategies (Shabrina et al., 2023).

In our work, we aim to support students similarly in their goal setting and planning in-real time by employing LLMs. However, it is important to note that in SRL theory, there is no singular approach to effective planning. Prior work examining plans students' created using the planning tool in CRYSTAL ISLAND identified a wide range of behaviors in how many plans students created and the types of plans (Goslen et al.,

2024). Hence, a real-time plan generation system requires the flexibility of generating many different planning strategies to be adaptive to individual students. We aim to analyze LLMs capabilities for this task in our work.

Plan Generation

Classical AI planning traditionally revolves around generating sequences of actions within an environment, a concept we extend to plan generation in CRYSTAL ISLAND (Ghallab et al., 2004). The field of automated planning has explored various techniques for plan generation over many years (Blum & Furst, 1997; Hoffmann & Nebel, 2001; Bercher et al., 2019). Recent approaches include hierarchical planning (Barták et al., 2021) and self-supervised learning for narrative plan generation (Polceanu et al., 2021). These methods have been applied in educational contexts with finite, small learning environments. For example, online learning environments have demonstrated improvement in student planning and new capabilities for creating study plans using graph networks (Segedy et al., 2015; Leung & Li, 2003). However, plan generation tasks in open-world game-based learning environments pose a unique challenge due to the scale of these environments.

For real-time plan generation in learning environments, it is essential to have an understanding of students' underlying problem-solving strategies. Students' gameplay within game-based learning environments such as CRYSTAL ISLAND is highly individualistic and characterized by exploratory behavior. Previous research with CRYSTAL ISLAND has delved into student goal and plan recognition, which is formalized as a classification task focused on recognizing students' goals and plans given a sequence of their in-game actions (Min et al., 2016, 2017). Long short-term memory (LSTM) networks were found to be effective in addressing student goal and plan recognition when treated as a multi-task multi-label classification problem with a binary vector representation of students' problem-solving actions (Goslen et al., 2022a, b). Prior work has also examined the use of LLMs for student goal recognition and found that T5 significantly outperforms competing machine learning-based methods using textual representations of student gameplay data as input (Kim et al., 2023). To date, LLMs have not been previously used to encode students' gameplay actions for the purpose of automatic student plan generation. By utilizing textual representations of student game log data, as well as the generation of textual output for executable in-game plans, language models hold significant promise for the creation of plans that are not only human-interpretable but also adaptable to various learning scenarios.

Large Language Models

The development of LLMs has been a significant recent advancement in the field of natural language processing (NLP). These models have proven successful in many NLP tasks like text summarization (e.g., BART (Lewis et al., 2020)), code assistance (e.g., Copilot (Chen et al., 2021)), and dialogue generation (e.g., GPT-3 (Brown et al., 2020), LaMDA (Thoppilan et al., 2022)). In our research, we aim to explore the effectiveness of LLM-based plan generation within game-based learning environments.

This work compares the validity of plans generated from two language models, T5 (Raffel et al., 2020) and GPT-3.5 (Brown et al., 2020). T5 is a transformer-based language modeling framework, adopting a consistent approach where all NLP tasks are reconfigured into a unified text-to-text format (Raffel et al., 2020). T5's architecture is an extension of the original transformer architecture (Vaswani et al., 2017), incorporating three key modifications. These modifications involve eliminating the layer norm bias, repositioning the layer normalization, and adopting an alternative position embedding scheme. GPT-3.5 is an autoregressive language model trained using 175 billion parameters. It has shown great potential when compared to other language models for NLP tasks like text-completion, question answering, and translation (Brown et al., 2020).

There is growing recognition of the potential of LLMs across a broad range of educational applications, including lesson planning, assessment, and providing scaffolding (Kasneci et al., 2023). In educational settings, T5 has been explored for the purpose of automatic question generation (Bulathwela et al., 2023; Jiao et al., 2023) and team communication analysis (Pande et al., 2023). The ability to fine-tune LLMs, such as T5, has yielded promising results for such tasks. GPT-3 is among the most widely used LLMs in recent years with applications like generating code explanations and improving automated evaluation of student text responses (Kasneci et al., 2023; MacNeil et al., 2022; Cochran et al., 2023). However, to protect student data without transmitting it to external servers, it is essential to train and operate the model locally on a machine. In this sense, the T5-small (60M parameters) offers a notable advantage due to its relatively smaller size compared to larger language models like Llama-2 (7B-70B; Touvron et al. (2023)) and GPT-3.5 (135B). This makes it feasible to train and deploy the model using the limited computing resources typically available in educational settings. While the smallest Llama-7B/13B model can be run on a single/double GPU, respectively, its zero-shot and few-shot performance does not match that of GPT-3.5, limiting its ability to achieve satisfactory performance. Additionally, fine-tuning Llama-2 also requires a much higher computing cost compared to T5-small.

Regarding more recent and advanced LLMs, GPT-4 (Achiam et al., 2023) has 1.76 trillion parameters, approximately 10 times more than GPT-3.5's 175 billion parameters, while GPT-4o (Hello GPT-4o, n.d.)ha s a parameter count similar to GPT-3.5 but the performance often matches to GPT-4. Recent work in adaptive support has utilized GPT-4 to generate levels of a science game-based learning environment in real-time through natural language prompts (Kumaran et al., 2024). Although both GPT-4 and GPT-4o surpass GPT-3.5 in many aspects including factual accuracy, data analysis, multimodal capabilities, and reduced bias, not all their advanced features equally benefit plan generation. In our scenario, the most critical aspect is the ability to handle highly technical and domain-specific content with more complex data analysis. These enhanced capabilities in GPT-4 and GPT-4o could be potentially advantageous for our plan generation task that analyzes extended sequences of student actions in educational games. Despite the technical advantages of GPT-4 and GPT-4o, their API costs pose a significant barrier in non-commercial and educational domains. GPT-4 and GPT-4o incur costs that are 60 times and 10 times higher per input token compared to GPT-3.5, respectively. Consideration is needed because the cost of LLM API services

could potentially restrict educational opportunities that should ideally be accessible to everyone equally.

Our work explores how we can utilize LLMs to support student learning and problem-solving by automatically generating plans that can be used to inform adaptive scaffolding for student SRL. We introduce a novel plan generation framework that utilizes student gameplay data as input to our LLMs and evaluate problem-solving plans generated by T5- and GPT-based plan generation models against student constructed plans. We discuss the pros and cons of each language model for this task, as well as how these models might be integrated into real-time adaptive learning environments to enhance student learning.

Game-Based Learning Environment

Our work is conducted with a game-based learning environment called CRYSTAL ISLAND (Rowe et al., 2011; Taub et al., 2020). Designed for eighth-grade microbiology education, it immerses students in a mystery narrative where they must investigate the origins of an illness that has spread across a remote island research station (Fig. 1). Students can visit different locations on the island, interact with non-player characters, read educational texts via books and posters, and run tests in a virtual laboratory. A planning support tool was incorporated into the game to assist students in orienting themselves to the learning environment and to support their goal setting and planning processes (Goslen et al., 2022a). Students were prompted throughout their gameplay to set goals and formulate plans using a drag-and-drop interface (Fig. 2). In the tool, a plan is defined as a target goal (green) that contains one or more low-level actions (blue) that contribute toward accomplishing the goals. Students can choose from 20

Fig. 1 CRYSTAL ISLAND game-based learning environment

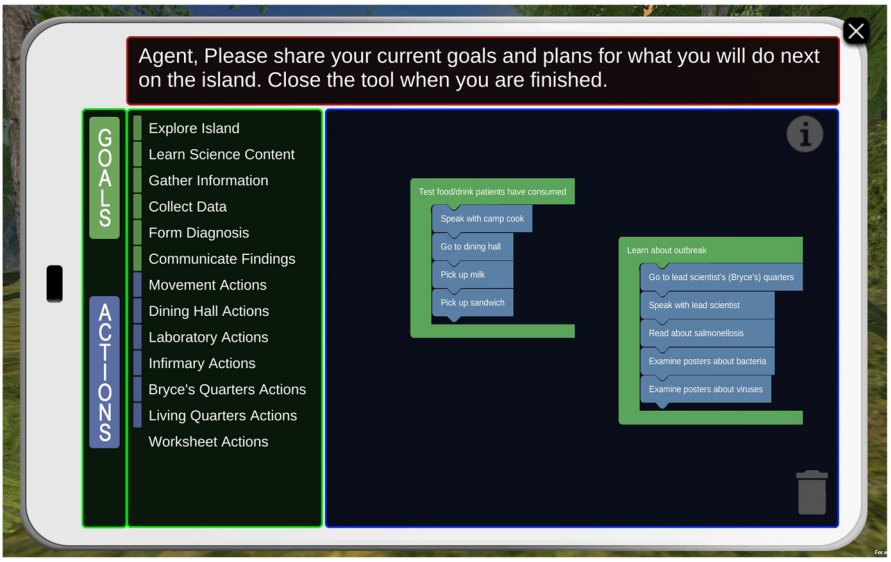

Fig. 2 Examples of goals and plans in the planning support tool

low-level goals and 55 low-level actions in the planning tool. Based on the design of the planning tool, the goals and actions listed in the planning support tool can be categorized in terms of 5 high-level goal categories (collect data, communicate findings, form diagnosis, learn science content, gather information) and 6 high-level action categories (explore, speak with characters, read science content, gather and scan items, examine poster, evaluate hypothesis).

The design of this tool is rooted in Winne and Hadwin's model of SRL (Winne & Hadwin, 1998, 2008), by allowing students to understand tasks in the game, externalize goals and plans, and then monitor and adapt those plans through the planning support tool. The low- and high-level goals and actions can be thought of as a hierarchy, with the lowest level being low-level actions, as those are direct actions that can be played in CRYSTAL ISLAND. High-level actions represent categories of actions that can be taken in the game, with each high-level action mapping to a set of low-level actions. The highest level in the hierarchy is high-level goals that represent goal categories that map to sets of low-level goals. Thus, plans students create in the tool are combinations of low-level goals and actions and are directly traceable with gameplay actions. High-level goals and actions demonstrate more general strategies students can take in the game.

Plan-Enhanced Adaptive Game-Based Learning Environment

The small size and nature of educational datasets often limit the performance of certain machine learning methods, causing prediction tasks to be abstracted to categories or fewer classification tasks. Additionally, classical machine learning methods require the conversion numerical input and output, limiting the interpretability of the results of

such models. With the introduction of pre-trained LLMs, the task of plan generation has the potential for significant predictive performance improvement, by allowing for more fine-grained predictions and interpretable results. Through real-time plan generation, we can support *how* students plan to achieve their goals. Figure 3 shows a framework diagram of an adaptive learning environment that incorporates plan generation models in real-time.

As students interact with the game-based learning environment, they once again engage in SRL processes, by exploring the environment to understand tasks, setting goals based on the game progression, enacting plans to achieve those goals, and monitoring and adapting those plans based on how well they worked for the given goal. Without the planning tool, students would engage in these SRL behaviors metacognitively. Through the planning support tool, students are able to externalize their plans at several points throughout the game. These gameplay interactions, as well as their planning activities, are automatically logged by the system. The data logged from the system is passed to the student plan generation framework, particularly a sequence of gameplay events, a target goal selected by the student and a set of previously completed goals. From this data, the plan generation framework outputs a set of low-level actions that are playable in CRYSTAL ISLAND. Based on the outputs of these models, the system can deliver adaptive interventions aimed at improving students' planning activities. For example, the plan generation framework can provide benefits in three aspects: student planning support, plan modification during gameplay, and game difficulty adjustment. 1) If a student is struggling during planning, it generates standardized plans that align with the presented goal, assisting the student's planning process in real-time. 2) If a student encounters difficulties while executing a plan, it suggests alternative plans, helping the student modify and strengthen the plan to achieve the goal more efficiently. 3) For students with high abilities and engagement in the game, it enhances learning experience and engagement by proposing advanced plans that offer a more diverse and in-depth experience.

Below are a few examples of interventions based on how a learning environment could adapt to such behaviors using the plan generation framework. Consider a student that kept the same goal in their planning tool throughout gameplay, and only

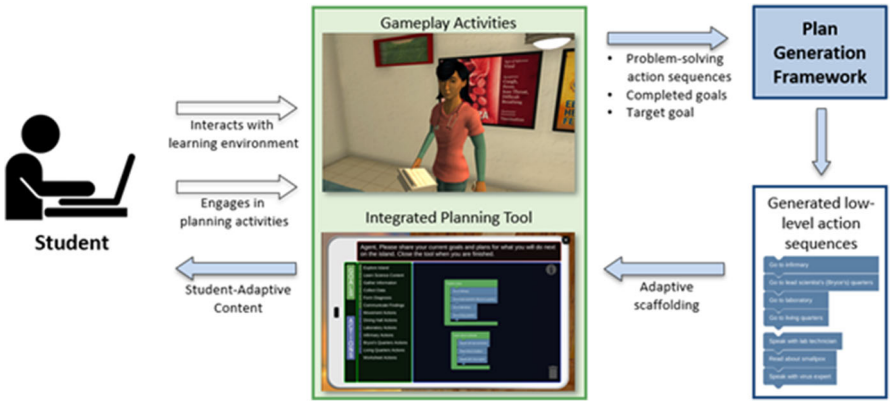

Fig. 3 Example game-based learning environment that incorporates a plan generation framework

changed the actions within that plan. This type of behavior is indicative of a student not externalizing their plans fully in the planning support tool because multiple goals are required to solve the mystery. A plan generation framework could show the student generated plans similar to other students to help them formulate new plans, prompting the student to achieve multiple goals. Consider a student that has the same plan persist throughout gameplay, without achieving the corresponding goal for a long time. Through tracelogs, an adaptive learning environment could identify if each low-level action in the plan had been enacted in gameplay. If a plan was not already completed, additional hints or alternative plans generated by the student plan generation framework could be provided to the student, potentially avoiding game abandonment. The framework could also provide more efficient plans, allowing the student to reach milestones in the game quicker, potentially increasing engagement in the game. Finally, consider a student that has a high level engagement with the planning support tool, by voluntarily accessing the planning tool to view and update plans, potentially demonstrating a high SRL skill-level. The plan generation framework could automatically provide more detailed hints of alternative actions to achieve their goals. Alternatively, an adaptive learning environment could adjust the game difficulty by incorporating new educational concepts based on how quickly the student achieves their goals or navigates the game's milestones, preventing the student from getting bored with the game. The plan generation framework could then automatically generate plans for the new content and help the student navigate further challenges.

Methods

This sections introduces the methods for constructing a novel plan generation framework that utilizes sequences of students' gameplay interactions, their completed goals and a current target goal as input. Through the use of an LLM, the framework produces a viable set of low-level actions that can be played in the game to achieve the corresponding target goal. To evaluate the framework, we compared the LLM-generated plans to plans students manually constructed in the planning support tool in CRYSTAL ISLAND.

In this study, we harness the capabilities of pre-trained LLMs to analyze textual game trace logs generated by student interactions with the CRYSTAL ISLAND game-based learning environment and produce corresponding plans to attain specific target objectives. The process is divided into two main phases. The first phase aims to transform the input to the framework, which are students' gameplay sequences. We utilize an encoder to transform the input because the event sequences consist of lists of game actions and locations, information contextually specific to CRYSTAL ISLAND. The second phase involves decoding the output of the framework, the generated plans. We use a sequence-to-sequence model that incorporates both an encoder and decoder because the output of the framework encompasses generated plans of varying lengths. Instead of a traditional classification task, this challenge falls within the realm of language generation. As a result, a decoder that is proficient in language generation becomes a

requisite component. In essence, our task necessitates the use of incorporating both an encoder to encode the gameplay input and a decoder for the plan output (Sutskever et al., 2014).

Many seq2seq language models have been explored, including well-known examples like the transformer model (Vaswani et al., 2017) and BART (Lewis et al., 2020). For our task, we utilize T5-small (Raffel et al., 2020), one of the transformer-based models, in order to examine the benefits of fine-tuning pre-trained language models for plan generation, and we utilize GPT-3.5 (Brown et al., 2020) to examine the impact of prompt engineering with a large language model for plan generation.

T5 Model Construction

We pre-trained the T5 model using a pre-processed English language corpus of approximately 700GB in size called the Colossal Clean Crawled Corpus (C4). Among the various sizes of available T5 models, we selected T5-small, which has a size of 600MB and comprises 60 million parameters, for fine-tuning our plan generation models. This choice was made with careful consideration of efficient model training and the potential for deploying the model to facilitate real-time plan generation in educational settings, where compute resources are frequently constrained.

To analyze student log data using LLMs, we considered a series of interactions by a student with CRYSTAL ISLAND as an event sequence. Student interactions with CRYSTAL ISLAND were automatically recorded as they navigated the environment. To construct our input, we extracted three attributes from each game event: event type (nine possible in-game activities), event argument (additional context for the event type), and event location (24 locations in the game). To illustrate, two event types in the game were reading a book and conversing with a character. Corresponding event arguments would be the title of the book being read or the name of the character being spoken to. The final attribute would be the location either of these events occurred. For example, one such event would be denoted as $e = (Conversation, Kim, Infirmary)$. Note this feature extraction aligns with prior work done in CRYSTAL ISLAND (Goslen et al., 2022a; Min et al., 2017).

Once we constructed a 3-tuple for each event, we discretized each students' event sequences based on the student's utilization of the planning support tool. A typical interaction within the context of CRYSTAL ISLAND involved a series of gameplay events (E_1), followed by an interaction with the planning tool where students created one or more plans that contain various target goals ($\mathcal{G}_t = \{Explore\ island\}$). Subsequently, there would be another series of events (E_2), followed by another planning tool interaction with updated or new plans ($\mathcal{G}_t = \{Find\ sick\ individuals\}$), and so on, until they complete their gameplay session. These planning tool interactions highlight key moments where students are using SRL processes and may need support. Thus, we segmented event sequences by planning tool interaction to align with real-time prediction. In other words, our model's input consisted of the actions taken by students up to the point of prediction (a planning tool interaction). Formally, we denote the input as $I : [E = (e_i, e_{i+1}, ..., e_{i+k}), Completed : \mathcal{G}_c, Goals : \mathcal{G}_t]$ where E is the series of game events prior to planning, $\mathcal{G}_c$ represents a set of goals that a student

has completed, and $\mathcal{G}_t$ denotes the set of target goals the student constructed during planning. Note, that $\mathcal{G}_c$ could be empty if a student had not completed any of their goals in the planning tool.

In the planning tool interactions, a student's plan consisted of one goal with a set of related low-level actions. Students were prompted to plan at various milestones in the game, allowing us to use the plans made in these interactions as reference points for predicting the next plan a student is likely to formulate in real-time. Consequently, we utilized the sequence of student gameplay actions up to the moment of the student opening the planning tool as input to our model, and the goals students selected during the planning tool were used as the target goal.

As an illustration, consider the interaction with CRYSTAL ISLAND described previously. In this context, the initial input row would be represented as $I = [E_1, \mathcal{G}_c = \{\}, \mathcal{G}_t = \{\textit{Explore island}\}]$ because the student had not completed goals at that point. For a goal to be considered completed, a student would need to execute all the low-level actions outlined in their plan prior to opening the planning tool. Thus, if the student in this example completed each the low-level action from in plan 1 during E_2, then we would consider plan 1 to be completed and the following input row would be $I = [E_2, \mathcal{G}_c = \{\textit{Explore island}\}, \mathcal{G}_t = \{\textit{Find sick individuals}\}]$. For the T5 model, we utilized a cumulative representation of event sequences and set a maximum event sequence length of 30 (Goslen et al., 2022a).

GPT-3.5 Model Construction

To provide a comparison to the fully-supervised T5 results, we also generated plans using a few-shot learning approach with OpenAI's GPT-3.5 model. We utilized the gpt-3.5-turbo model and generated plans using the chat completions API. We selected this model because it shows robust performance across a range of tasks, has lower resource requirements and is lower cost than GPT-4. Because we are able to leverage student planning activities for evaluation and training, we opted for a few-shot learning approach. In this request prompt, we included a description of CRYSTAL ISLAND and the overarching goal of the game, a list of all goals included in the planning tool, a list of all actions included in the planning tool, and two example inputs with their corresponding set of planned actions. These example inputs were randomly selected for each goal category, since not every goal within a category was used more than two times. This meant that every target goal within the same goal category would contain the same examples in the prompt, and these examples were excluded from the analysis. The end of the prompt consisted of the input event sequence (E), the set of completed goals ($\mathcal{G}_c$) and the target goal ($\mathcal{G}_t$). Because the GPT-3.5 API has a token limit per call-response, we limited the number of examples used for few-shot learning approach to two plans in the prompt.

Table 1 shows an example prompt used in the GPT-3.5 plan generation model. The event sequences were of length 30 so as to be consistent with the T5 input. The example event sequences shown in the table resemble T5 input as well.

Table 1 An example GPT-3.5 prompt

For each given event sequence, completed goals and target goal, generate a set of actions for a player to enact in a game based learning environment. The setting for the learning environment is a remote island called CRYSTAL ISLAND where some members of a research team have recently started to become sick. The player must determine what disease is spreading amongst the researchers, what food item it is spreading through, and what the correct treatment or prevention plan is based on the disease that's spreading. Players can take the following actions to solve the mystery:

1) speak to virtual characters to learn about their symptoms, the mystery, or microbiology knowledge

2) read informational texts about types of diseases or other microbiology knowledge

3) collect food items in the world

4) scan food items to see if they contain bacteria, viruses, or carcinogens

5) view posters to learn about the symptoms and treatments of various diseases

A target goal is selected from the following set: [listed 20 goals from the planning tool]. The set of actions can be chosen from the following: [listed 55 actions from the planning tool].

Plans can contain as many actions as the player wants. An event sequence represents the set of actions already completed by the player. The completed goals represent the set of target goals the player already accomplished.

Given an event sequence, completed goals and the player's next target goal, please generate a set of actions that a player can enact to accomplish the target goal.

Here is an example:

Event sequence:

Movement LabStairs

Movement Lab

Scanner Bread Lab

PlotPoint TestContaminatedObject Lab

Worksheet Lab

Completed Goals: test objects patients have touched

Target Goal:test food drink patients have consumed

Actions: speak with camp nurse, speak with patients, examine poster about salmonellosis

Generate a set of actions for the following:

Event sequence:

BooksAndArticles Scientific Method Infirmary

ConceptMatrices Scientific Method Infirmary

BooksAndArticles How do diseases spread? Infirmary

ConceptMatrices How do diseases spread? Infirmary

Movement InfirmaryStairs

Completed Goals:

Target Goal:test food drink patients have consumed

Actions:

Note: event sequences and example plans have been reduced for space. Actual prompts included event sequences of length 30 and two example plans

Evaluation

Dataset

The dataset used in this work was collected during the COVID-19 pandemic from a group of 144 middle school students. Students played the game remotely during asynchronous science class time. The average age of students was 13.2 years, with 60% being female. Students played the game for 94.7 minutes on average (SD=47.7) and completed pre- and post-tests. Students' gameplay was logged automatically, yielding game log data which included students' in-game planning activities. Students' plans were distributed across five categories of goals as follows: collect data: 22%, communicate findings: 4%, form diagnosis: 13%, learn science content: 22%, and gather information: 40% (Goslen et al., 2022a).

Experiment setting

For T5, we performed a 5-fold cross-validation of fine-tuning the models to prevent the memorization of input plans and to generate plans for all the student data. The input also allowed multiple goals. A total of 510 plans were included for the analysis, with 11,610 cumulative event sequences. We selected the hyperparameters from preliminary explorations with the search space of learning rate [0.0001, 0.0003, 0.0005] and batch size [4, 8, 16] and fixed the following hyperparameters for T5: a learning rate of 0.0003, a batch size of 4, an input max length of 1024, a weight decay of 0.01, and a warmup setting of 1000.

For GPT-3.5, we conducted prompt engineering through few-shot learning, described in the Method section, omitting the use of cross-validation. Unlike the T5 model, each API call consisted of one single goal, meaning if an input had more than one target goal associated with it, we would make separate API calls for each target goal and provide the same input to the model. This was done to ensure the interpretability of the GPT-3.5 response. We also did not construct event sequences cumulatively for this model because API calls were limited to a maximum of three per minute, increasing our runtime considerably. After removing the provided examples from the dataset, we had 497 generated plans from the GPT-3.5 model.

Quantitative and Qualitative Analysis

To evaluate both language models' ability to generate plans and to answer our research questions, we developed evaluation methods to qualitatively and quantitatively assess the generated plans. Because currently there is not a validated approach to evaluate plans in CRYSTAL ISLAND, these evaluation methods are meant to be exploratory in nature and provide a range of techniques for plan evaluation. Our quantitative metrics (RQ1) focus on the percentage of actions within the generated plan (GP) that match actions in the corresponding student plan (SP). To calculate this percentage, we counted how many actions in the GP matched the actions in the SP and divided by the size of the GP. We calculated the percentage of actions in the GP that matched actions in the corresponding SP at the low-level (p_L) and also using the high-level action categories

(p_H). Additionally, we examined the percent matched across three cases (1) where the size of SP and GP were equal ($SP = GP$), (2) where the GP was larger than the SP ($SP < GP$), and (3) where the GP was smaller than the SP ($SP > GP$). This analysis provided more insights into how each model performed compared to student plans and when it might be useful to use generated plans in adaptive systems.

For our qualitative analysis (RQ2 and RQ3), we compared the contents of generated plans from each model with students' plans using high-level goal and action categories. For T5, GPT-3.5, and student plans, we separated plans into groups by mapping the target goal of the plan to its corresponding goal category. There were five total goal categories: gather information, learn science content, form diagnosis, collect data, and communicate findings. Then, we mapped the low-level actions in each of these groups to their high-level action categories and summed each action category. This provided a way to understand which types of actions were being used for each goal category. In our analysis, we examined the high-level action category distributions of both T5 and GPT-3.5 generated plans and compared them to student high-level action category distributions for each goal category. Finally, we extracted several example plans from the language models, as well as students-created plans, to present a comparison of the quality of plans. Examples were chosen from each of the target goal categories and were meant to showcase a wide variety of behavior and how our system could provide intervention and support with these generated plans. These examples contribute to the overall discussion on how language models could be leveraged for adaptivity in learning environments.

Results

RQ1: How Does the Performance of a Fully-Supervised Fine-Tuned LLM Compare with a Prompt-Engineered Few-Shot Learning Approach for the Task of Student Plan Generation?

To compare the fully-supervised and few-shot learning approaches, we assessed the T5 and GPT-3.5 generated plans based on the size of generated plans and how closely they matched the student plans. Figure 4 illustrates the distribution three variables considered in this evaluation: (1) the differences in plan size between student-created plans and those generated by the T5 model (left), (2) low-level match percentages (center), and (3) high-level match percentages (right).

The plan size difference had a median of 0.0 (mean = 0.0, SD = 3.4). In this case, a negative difference indicated the generated plans contained more low-level actions than the student plans. The maximum observed difference in plan size was 26, signifying that the T5 model was larger than the corresponding student plan by 26 low-level actions.

The match percentage for low-level actions (p_L) had a median of 0.0 (mean = 25.2, SD = 32.3), as depicted in the center boxplot. Among the 11,610 generated plan instances, 8.1% generated all low-level actions present in the corresponding students' plans. We also observed that 51.2% of generated plans had no low-level actions matching those of the student plan.

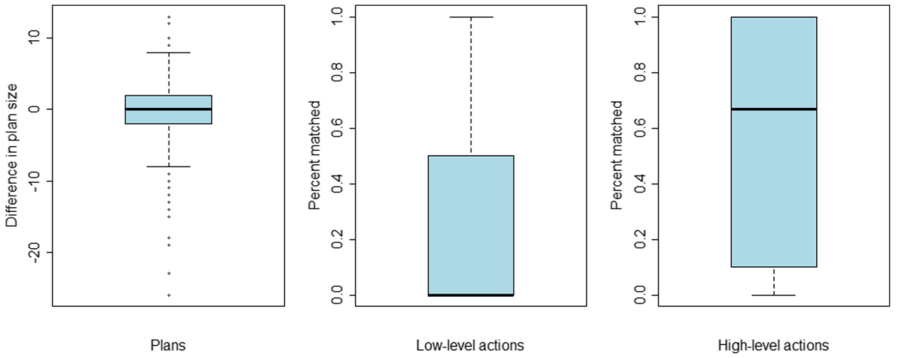

Fig. 4 Distribution metrics for T5 generated plans

The match percentage for high-level actions (p_H) had a median of 66.7 (mean = 59.8, SD = 41.9). Notably, there was an increase in exact matches for high-level actions to 44.4% and a decrease in the amount of plans with no high-level action match to 24.8%. Although a small percentage of T5 generated plans closely matched students' low-level actions, the increase in high-level match percentage indicates that the model generated low-level actions within the same high-level categories. These generated plans could prove useful for adaptive support in that it could show students a wide variety of plans that would be similar to their own plan, allowing them to reflect on their plan.

Figure 5 shows the distributions of the plans generated from the GPT-3.5 model. The boxplot on the left shows the median plan size difference was -1 (mean=-0.49, SD=2.44). The maximum and minimum plan size difference were both 10, meaning at most the GPT model included 10 additional low-level actions in its plan and at least the student included 10 more low-level actions in their plan. This difference is a bit smaller than the T5 generated plans, possibly due to token limitations with the API. For GPT plans, low-level action match percentage (p_L) had a median of 0.0 (mean=19.0, SD=28.7). Out of 497 GPT generated plans, 6.0% had plans where all

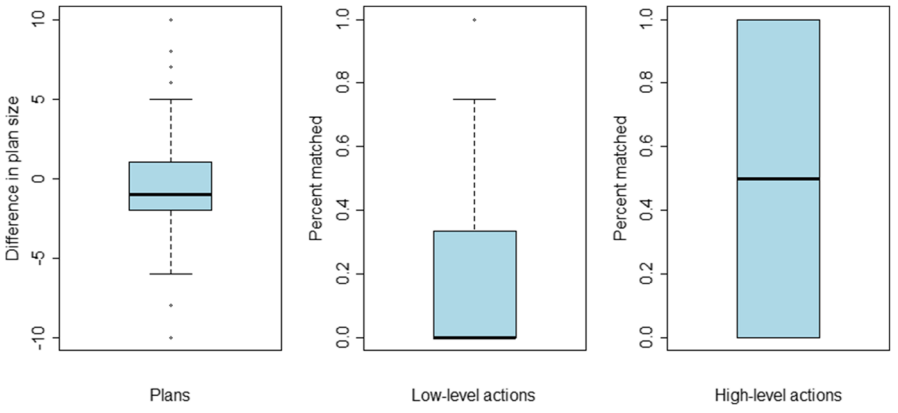

Fig. 5 Distribution metrics for GPT generated plans

low-level actions matched that of the students' plans and 58.9% contained no low-level action matches to the corresponding student plan. For GPT-3.5, the median p_H was 50.0% (mean=50.3% , SD=41.5%). Similarly to T5, we saw an increase in generated plans where all high-level actions were present in the students' plan to 33.6% and a decrease in no matches to 28.9%. These results were lower than the T5 match rates at the high-level, indicating T5 more closely aligned with students' plans.

A comparison of match rate between LLMs is shown in Table 2. For both models, we observed a match percentage increase between low-level and high-level match rates. For T5 generated plans, the case SP = GP performed best in terms of low-level matching, whereas for GPT-3.5 the case SP > GP contained the highest p_L. This indicates that fully-supervised fine-tuning methods (T5) performed best in terms of generating plans very similar to students, and few-shot learning with prompt engineering (GPT-3.5) potentially generated subsets of students plans. For both LLMs, we noticed the largest increase in matches when the generated plan was smaller than the student plan (SP > GP). We believe these could be reflective of potentially more efficient plans, because the generated plans contained fewer low-level actions and these actions were within the same high-level category. These types of generated plans demonstrate a potential use-case for real-time systems that could help the student access more of the game in the time allotted. The case of SP < GP performed worse for both models, which indicates that when the models generate larger plans, they are more distinct from the student plans. Overall, we found that for both cases of match rate GPT-3.5 performed worse than T5, but followed similar trends as T5.

Furthermore, there was a significant difference in average plan size difference, low-level matching and high-level matching between T5 and GPT-3.5 generated plans. Results from a two-sided t-test showed a significant difference in the average difference in plan size between T5 and GPT-3.5 generated plans ($p < 0.0001$). Additionally, results from two Wilcoxon rank sum tests showed a significant difference in the average percent matched for both low-level actions ($p < 0.0001$) and high-level actions ($p < 0.0001$) between T5 and GPT-3.5 generated plans.

RQ2: Does the LLM-Based Student Plan Generation Framework Produce Valid Problem-Solving Plans that can be Enacted in CRYSTAL ISLAND and Achieve the Target Goal Set by the Student?

Table 3 shows the high-level action distribution for students, T5 and GPT-3.5. For goal categories "gather information", "learn science content" and "collect data", T5 had

Table 2 The percents of action matches between T5 and GPT-3.5 generated plans and students' plans at the low-level (p_L) and high-level (p_H)

	T5			GPT-3.5		
	N dist (%)	p_L (%)	p_H (%)	N dist (%)	p_L (%)	p_H (%)
$SP = GP$	21.4	34.7	66.0	19.3	28.9	47.9
$SP < GP$	42.5	16.1	46.8	52.7	10.4	39.1
$SP > GP$	36.1	30.2	71.2	27.9	32.9	71.3
Overall	100.0	25.2	59.8	100.0	21.9	51.2

Table 3 High-level action category distribution for each high-level goal across student, T5, and GPT-3.5 plans

		Collect data (%)	Communicate findings (%)	Form diagnosis (%)	Learn science content (%)	Gather info. (%)
T5	Explore	5.94	13.74	0.00	23.07	57.47
	Speak with characters	19.44	24.29	6.57	34.06	20.73
	Read science content	8.78	29.53	1.63	21.01	9.45
	Gather and scan items	55.34	12.04	48.68	11.28	6.41
	Examine poster	7.53	11.26	3.26	9.62	4.97
	Evaluate hypothesis	2.96	9.14	39.86	0.96	0.96
GPT-3.5	Explore	0.26	0.00	4.35	24.60	94.33
	Speak with characters	26.04	22.22	41.30	20.27	4.08
	Read science content	7.55	19.44	17.39	22.55	0.35
	Gather and scan items	40.89	0.00	4.35	0.46	0.71
	Examine poster	24.74	22.20	28.26	31.44	0.53
	Evaluate hypothesis	0.52	36.11	4.35	0.68	0.00
Student	Explore	5.99	8.33	0.00	14.92	58.97
	Speak with characters	20.39	16.67	0.00	33.97	17.76
	Read science content	15.08	8.33	71.62	22.22	6.57
	Gather and scan items	50.78	8.33	8.11	11.11	9.77
	Examine poster	3.10	41.67	14.86	14.60	5.86
	Evaluate hypothesis	4.67	16.67	5.41	3.17	1.07

very similar distributions to students' plans. This is expected as these are the most used goal categories and T5 had more training data than GPT-3.5. Interestingly, students used primarily "read science content" actions for the goal of "form diagnosis", but T5 generated plans that mainly consisted of "gather and scan items" and "evaluate hypothesis" for the same goal. This demonstrates a case where the generated plan might benefit students, as evaluating a hypothesis helps to form a diagnosis in the game. Therefore, suggesting this action as part of the planning process could aid in the students' overall comprehension and learning efficiency.

The GPT-3.5 generated plans varied in comparison to the student plans. For example, the GPT-3.5 "gather information" high-level goal almost exclusively consisted of "explore" actions. "Gather information" goals were the most used by students and are the most general of the goals presented in the planning tool, meaning there are several ways to complete the goals. This implies there should be diversity in the high-level actions for this high-level goal, so the GPT-3.5 model is not ideal in this case. Notably, the "communicate findings" category had distinct distributions for all three sets of plans. This goal category is the most specific goal category out of the group, with the game explicitly describing how to communicate findings and win the game. To do this, a player must fill in the diagnosis and speak with the camp nurse. GPT-3.5 generated plans were the only plans that were composed of the two corresponding high-level actions to accomplish this goal.

To further understand the performance of the few-shot learning technique used with GPT-3.5, we compared the GPT-3.5 plans with the examples provided in the prompt. We found that 5.8% of GPT-3.5 generated plans contained all actions present in the examples provided in the prompt, and 27.1% contained at least half of the actions provided. Alternatively, 52.5% of the GPT-3.5 plans contained no actions present in the examples but were generated based on the given context information, shown in Table 1. One interesting case was where GPT-3.5 combined both provided examples as its generated plan. Overall, while the GPT-3.5 generated plans did not align with students' plans, it was generally able to generate valid plans beyond the provided examples.

RQ3: How Effectively do LLM-Based Plan Generation Models Produce Plans that can Inform Real-Time Adaptive Scaffolding in a Game-Based Learning Environment?

To answer this question, we extracted example plans that showcase the nuances of plan validity analysis. Table 4 shows an example student, T5, and GPT-3.5 plans for each listed target goal. Each distinct target goal shown in the table was selected from a different goal category to provide more comprehensive examples. There was a wide variety of plans for each goal across all sets of plans (student created and LLM-generated). This variance is evident in the data, and we observed similar patterns between LLM-generated plans. For each goal category, we extracted interesting cases of generated plans that have implications for how we might implement such models in adaptive systems.

The generated plans for the "test objects patients have touched" goal (a) demonstrated a case where the plan is very long, and in this case exceeds the amount of items a player could pick up at a time. This could be beneficial to students to understand their options and finetune their hypothesis, but it could also contribute to students trying to game the system by testing all objects. We did notice that most generated plans for this goal included the action "use scanner and test objects", which is crucial to achieving the goal. This highlights that LLMs have the capability of providing contextually important information.

"Report evidence-based diagnosis" (b) was the least used goal, and to achieve it students are instructed to fill in information on their diagnosis worksheet and speak with the camp nurse about the diagnosis. None of the plans included the speak action, but they did include several "fill in" actions. Interestingly, the T5 model generated a plan identical to a different students' plan. We noticed T5 did this several times, which could be beneficial in real-time settings. This type of cross-student hints could help foster collaboration or competition in the game. For example, if the plan generated by the LLM based on student A's game event sequence is similar to student B's plan, it implies that the student B is going through a similar situation as student A. Thus, it can encourage collaboration with these two students, allowing them to share the reasoning and execution methods behind their plans, there by increasing performance and engagement in the game.

Table 4 Example student-created plans and LLM-generated plans for a target goal in each goal category

Target goal	(a) Test objects patients have touched
Target goal	(a) Test objects patients have touched
Student plan	speak with lead scientist, pick up lettuce, speak with virus expert, speak with bacteria expert, read about viruses, pick up peanuts, use scanner to test objects
T5 plan	pick up raw chicken, pick up cheese, pick up egg, pick up water, pick up lettuce, pick up yogurt
GPT-3.5 plan	pick up water, pick up apple, pick up bread, pick up cheese, pick up egg, pick up sandwich, pick up lettuce, pick up jelly, pick up peanuts, pick up raw chicken, pick up bananas, use scanner to test objects
Target goal	(b) Report evidence-based diagnosis
Student plan	fill in symptoms
T5 plan	read about storing food
GPT-3.5 plan	examine poster about salmonellosis, speak with lab technician, fill in symptoms, fill in tested object, fill in test results, fill in final diagnosis
Target goal	(c) Evaluate hypothesis anthrax
Student plan	read about viruses, read about carcinogens, read about mutagens, read about microbes
T5 plan	fill in symptoms, fill in prediction about anthrax, fill in prediction about smallpox
GPT-3.5 plan	speak with lead scientist, read about smallpox, speak with lab technician, examine poster about smallpox
Target goal	(d) Learn about disease
Student plan	go to infirmary, speak with camp nurse, speak with patients, read about microbes, examine poster about influenza, examine poster about ebola, examine poster about botulism
T5 plan	go to laboratory, go to lead scientist's quarters, speak with lab technician, read about how diseases spread
GPT-3.5 plan	go to dining hall, speak with camp cook, go to laboratory, speak with lab technician
Target goal	(e) Explore camp
Student plan	go to infirmary, go to lead scientist s (bryce s) quarters, go to laboratory, go to living quarters
T5 plan	go to infirmary, go to lead scientist's quarters, go to laboratory, go to living quarters, speak with lab technician, read about smallpox, speak with virus expert
GPT-3.5 plan	go to beach, go to dining hall, go to forest

For the target goal "evaluate hypothesis anthrax" (c), both models generated plans that included an action having to do with smallpox. This could be due to smallpox actions being enacted by several students, however in a real-time system, we would want to ensure suggested planned actions align with the target goal. Additionally, anthrax is not a potential solution for the game, so suggesting a different disease might give away the mystery and prevent the student from learning.

For the target goal "learn about disease" (d), GPT-3.5 seemed to interpret this goal as learning about the specific diseases presented in the game, while the game offers more general educational content about diseases. We can tell this because the actions it generated are to go to a location and speak with a character. The purpose of these characters are to explain the situation on the island, particularly with the camp cook. More general information about disease is found in other locations, which would be

represented in the game trace logs. Additionally, there is another goal in the same category called "learn about outbreak" that is meant to be more game specific. This might indicate that we should define each goal for the model for better results. The T5 model plans generated several "go to" actions where science content is located, but it is unclear if the model was aware of that connection. T5 plans also included several "read" and "speak" actions, which are appropriate for this target goal.

GPT-3.5 included many "go to beach" and "go to forest" actions for the "explore camp" (e) goal, which is a bit off-task for the game. In contrast, the dining hall does offer useful information for solving the mystery, so the plan is not entirely invalid. In contrast, T5 plans contained several meaningful locations as well as "speak" actions, which could benefit the student in getting oriented in the game.

Overall, these patterns demonstrate the potential for LLMs to inform real-time systems. Generally, these models generated coherent plans, with no duplicates and generated actions aligning with students' actions. We did identify instances of hallucinations in both models, but they still aligned with actions that could be completed in the game. For example, one T5 generated plan included the action "pick up yogurt", which is not in the planning tool but is still present in the game. GPT-3.5 had several more general, natural language actions like "scan objects to test for bacteria and viruses". While this does not align with the planning tool, it provides additional reasoning to the action, which could help students in their understanding. We will discuss further implications of these findings in 8.2.

Discussion

Training Approach Comparison Implications

Our results demonstrate the promise of leveraging language models for plan generation in game-based learning environments. We found T5-based generated plans are very closely aligned with student plans and GPT-3.5-based plans are more summative in nature. This aligns with our understanding of the methodology, as the T5 model was fully supervised using cross validation and the GPT-3.5 model utilized a few shot learning approach with the game context and two examples provided for every input. This distinction is important in educational settings, as we typically have small quantities of data to train on.

The T5 model did take longer to run as the fine-tuning of the language model requires considerable computational power. The model seemed to work particularly well and generally generated valid plans. On the other hand, the GPT-3.5 model had different types of limitations. The token limit to the API proved to be a challenge as to how to concisely explain the task and provide the input for each API request. Additionally, unlike ChatGPT there is no memory allocated for API calls, so the same information has to be provided several times, limiting us to one event sequence per call. Additionally, the API is not free, even for GPT-3.5, which is a significant disadvantage relative to open-source models. These factors informed our decision to limit the number of generated plans. Thus, we did not utilize a cumulative representation of the event sequences like we did for the T5 model. Additionally, API calls were limited to three

calls per minute, making the runtime much longer. Despite these limitations, the GPT-3.5 generated plans followed similar trends to the T5 models and demonstrated promise in generating plans in a game-based learning environment context.

In a practical implementation of this plan generation framework, it might prove more beneficial to use a language model like T5 that can be fined-tuned with no financial cost. The computational effort could be completed before the classroom implementation, and then the generative model could be run in real-time as students are interacting with the learning environment. Currently, using the GPT-3.5 API might not be practical for real-time classroom use because of cost and run-time limitations. Another drawback of using OpenAI GPT services is that data shared with a third-party and potentially can be used as training data by OpenAI. Depending on the type of data collected in a classroom setting, this type of data sharing could raise issues of consent and data usage.

Implications for Adaptive Learning Environments

Our results demonstrated that over half of plans generated by both language models aligned with the corresponding high-level action category from student plans. This suggests that even in situations where there were few low-level action matches, the plans still incorporated similar high-level actions to students. Based on the design of the planning tool, low-level actions within a high-level action category share a common narrative purpose within the game. For example, the “explore" high-level action mainly maps to “go to" low-level actions, all of which involve exploring the game environment. As an illustration, consider a student’s plan that included the following actions: “go to infirmary”, “speak with camp nurse”. The T5 predicted plan in this instance was: “speak with lead scientist”, “go to infirmary”, "go to lead scientist’s quarters." All actions align with the given goal when mapped to their high-level action category, event though the “speak” action differs. Generating different planned actions from students’ ones is desirable because only if there exist some differences between LLM generated plans and students’ plan, LLM can provide meaningful advice to students, although more work could be done to understand which plans would be beneficial in a real-time scenario. For example, the action “speak with camp nurse” is a distinction from the student’s plan. Early in the game, it is beneficial for a player to engage with the camp nurse, so prompting a student to speak with her could enhance their gameplay and overall experience in the game. Incorporating a temporal element to the plan generation might prove helpful to improving the models. We also noticed that some sets of low-level actions generated for a given student were present in other students’ plans targeting the same goals. We believe this is due to the nature of language models generating most probable plans for the situation the student is in rather than fully personalized to individual students. In a real-time scenario, these predictions could help improve students’ planning abilities through demonstrating more efficient plans. There is also a potential for the game to adapt to individual students’ strategies by providing instructive hints with these plans.

A limitation of this research pertains to the concept of optimal planning activities. To our knowledge, there does not exist a ground truth for optimal plans in an open-

world game-based learning environment such as CRYSTAL ISLAND. In this research, we rely on plans created by students as our reference point, but it is important to note that there was a considerable range in the planning activities exhibited by students. The extent to which students utilized the planning support tool varied significantly, both in terms of how frequently they accessed it and how many plans they formulated during gameplay. The analysis of the high-level action category distributions between student and language model plans further demonstrated the ability of language models to generate plans resembling those of students. Although our language models appear to generate plans resembling those of students, the absence of a clear measure for plan quality makes it challenging to determine whether the generated plans are superior to those created by students. Furthermore, educational games present two key strategies for students: winning the game and learning the educational material. Depending on a student's primary objective, an adaptive system may need to tailor its prompts accordingly. Achievement goal orientations have been shown to provide insights into students' approach to these strategies, as well as other metacognitive behaviors like motivation (Cloude et al., 2019). Incorporating such metrics into the language models could help improve generated plans in adaptive systems.

We also noticed that the language models occasionally generated low-level actions that were not available for students to choose for their plans using the planning support tool but were playable within the game. For instance, students have the ability to pick up items and test then them for various diseases in the game. One low-level action we observed in a generated plan was "pick up coconut." Players can pick up a coconut in the game, but cannot select that as a planned action in the planning tool. The planning tool was designed to limit the amount of low-level actions offered to students to reduce cognitive load. Consequently, "pick up coconut" was omitted, as it is not essential for solving the mystery. One potential approach to address this problem is to introduce action constraints because relying solely on the general game context and students' plans is likely to result in suboptimal plans. Nevertheless, it's noteworthy that the language model generated valid low-level actions in this context, as this indicates potential for LLMs to generate content for such support tools. This could help with designing similar SRL-based tools or extending the planning support tool to other game-based learning concepts.

Although there is no correct answer to the goal-setting and planning that students strive to achieve, there can be various strategies to make plans, such as being more efficient or seeking more knowledge. Our approach differs significantly from existing work in that it provides opportunities for students to review candidate plans learned from the collective intelligence of their peers', enabling them to reflect on and adjust their own plans as necessary, which is another key component of SRL. Most importantly, this approach is not specific to the game domain, as the generated plans are learned from data rather than manually created by experts. Thus, our framework has generality beyond CRYSTAL ISLAND to other educational games.

Overall, we find that the utilization of LLMs demonstrates promise for the task of plan generation and has the potential to enhance online learning environments. In challenging learning scenarios, like CRYSTAL ISLAND, students can become confused about next steps, then frustrated and then give up on the game entirely. The planning support tool helps orient them in the game and can help facilitate engaging in SRL

processes. However, an LLM-driven plan generation system would help improve this experience even further and potentially prevent negative emotions from occurring. For example, if a student is constructing and enacting plans, but is not getting closer to solving the mystery of the game. The existing version of CRYSTAL ISLAND could not assist, while the plan generation framework could provide specific guidelines adaptive to the student's learning situation. Furthermore, a classical machine-learning based plan generation method would train gameplay data in a numerical format and, given the limited size of the dataset, output only numerical categories of high-level actions. This is because there are 6 high-level actions versus 55 low-level actions, and there is not enough data to train on all 55 low-level actions using classical machine learning models like Random Forest or LSTM. LLMs provide the capability of pre-training on large language corpora creating more robust models. Thus, an LLM would allow for natural language versions of the gameplay data (which are more interpretable) to be input and can output any number of low-level actions for the plan generation task. This widens the capabilities of plan generation models in general. In our case, the LLM-based plan generation framework could generate several potential options for the student and help facilitate them continuing in the game.

Conclusion and Future Work

In this work, we introduce a plan generation framework that leverages LLMs for the task of plan generating sets of problem-solving actions within a game-based learning environment by utilizing textual representations of student gameplay. This framework is evaluated by presenting a quantitative and qualitative analysis of both T5 and GPT-3.5 generated plans, as well as plans created by students using a planning support tool in the game. Results show that both LLMs have the ability to generate valid problem-solving plans, with T5 plans aligning more closely to student plans due to its fully-supervised fine-tuned training approach. T5 also shows similar patterns in the types of actions generated for a given goal category. GPT-3.5 had more variability in its plans, likely because of the few-shot learning approach; it typically included different actions for each goal category than did student-generated plans. Despite their differences, both models demonstrated the potential of utilizing LLMs to provide adaptive, real-time assistance to students through the generation of problem-solving plans.

These findings point towards several areas for future research. Exploring the performance of more recent releases of ChatGPT, like GPT-4 and GPT-4o, could prove useful to understanding the potential benefit of these models in educational settings. Applying these plan generation techniques to other online educational environments could further demonstrate the generalizability of the methods and the potential of LLMs for the task of plan generation. Metrics for plan evaluation are still needed for game-based learning environments in the context of science problem-solving. Additional work is needed to assess the quality of plans constructed by students and model-generated plans. Incorporating both sequentiality of goals and other high-level strategy metrics, like achievement goal orientation, into the models, may help improve the framework for real-time adaptive support. Exploring how such methods could be applied to gener-

ating other types of SRL scaffolding or enhancing current tools would provide insight into the generalizability of these techniques. Additionally, exploring how natural language representations of the input data impacts performance is a promising area for future work. Finally, it will be important to investigate how incorporating LLM-based plan generation models into game-based learning environments can drive adaptive scaffolding to create more effective learning experiences for students.

Acknowledgements This research was supported by funding from the National Science Foundation under grants DUE-1761178 and DRL-2112635. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

Author Contributions All authors contributed to the study conception and design. Data analysis was performed by Alex Goslen and Yeojin Kim. The first draft of the manuscript was written by Alex Goslen, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding This research was supported by funding from the National Science Foundation (NSF) under grants DUE-1761178 and DRL-2112635.

Availability of data and material Data and materials created for this research are available upon request. Please direct all inquiries to the corresponding author.

Code availability Code created for this research is available upon request. Please direct all inquiries to the corresponding author.

Declarations

Conflicts of interest/Competing interests No potential conflicts of interest.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. (2023). Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

Azevedo, R., Martin, S. A., Taub, M., Mudrick, N. V., Millar, G. C., & Grafsgaard, J. F. (2016). Are pedagogical agents' external regulation effective in fostering learning with intelligent tutoring systems? In: Intelligent Tutoring Systems: 13th International Conference, ITS 2016, Zagreb, Croatia, June 7-10, 2016. Proceedings 13, pp. 197–207. Springer

Azevedo, R., Bouchet, F., Duffy, M., Harley, J., Taub, M., Trevors, G., Cloude, E., Dever, D., Wiedbusch, M., Wortha, F., et al. (2022). Lessons learned and future directions of metatutor: Leveraging multichannel data to scaffold self-regulated learning with an intelligent tutoring system. *Frontiers in Psychology., 13*, 813632.

Barták, R., Ondrčková, S., Behnke, G., & Bercher, P. (2021). Correcting hierarchical plans by action deletion. *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning, 18*, 99–109.

Bercher, P., Alford, R., & Höller, D. (2019). A survey on hierarchical planning-one abstract idea, many concrete realizations. In: IJCAI, pp. 6267–6275.

Blum, A. L., & Furst, M. L. (1997). Fast planning through planning graph analysis. *Artificial intelligence., 90*(1–2), 281–300.

Boekaerts, M., & Pekrun, R. (2015). Emotions and emotion regulation in academic settings. *Handbook of Educational Psychology* (pp. 90–104). New York, NY: Routledge.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems., 33*, 1877–1901.

Bulathwela, S., Muse, H., & Yilmaz, E. (2023). Scalable educational question generation with pre-trained language models. In: International Conference on Artificial Intelligence in Education, pp. 327–339. Springer

Chen, M., Tworek, J., Jun, H., et al. (2021). Evaluating large language models trained on code. CoRR. arXiv:2107.03374

Cloude, E. B., Taub, M., Lester, J., & Azevedo, R. (2019). The role of achievement goal orientation on metacognitive process use in game-based learning. In: Artificial Intelligence in Education: 20th International Conference, AIED 2019, Chicago, IL, USA, June 25-29, 2019, Proceedings, Part II 20, pp. 36–40 . Springer

Cochran, K., Cohn, C., Rouet, J. F., & Hastings, P. (2023). Improving automated evaluation of student text responses using gpt-3.5 for text data augmentation. In: International Conference on Artificial Intelligence in Education, pp. 217–228. Springer

Dever, D. A., Amon, M. J., Vrzakova, H., Wiedbusch, M. D., Cloude, E. B., & Azevedo, R. (2022). Capturing sequences of learners' self-regulatory interactions with instructional material during game-based learning using auto-recurrence quantification analysis. *Frontiers in Psychology., 13*, 813677.

Ghallab, M., Nau, D., & Traverso, P. (2004). *Automated Planning: Theory and Practice*. Amsterdam: Elsevier.

Goslen, A., Carpenter, D., Rowe, J., Azevedo, R., & Lester, J. (2022). Robust Player Plan Recognition in Digital Games with Multi-Task Multi-Label Learning. In: Proceedings of the 18th AAAI Conference on AIIDE, pp. 105–112. AAAI Press, Pomona, CA, USA.

Goslen, A., Carpenter, D., Rowe, J., Henderson, N., Azevedo, R., & Lester, J. (2022). Leveraging Student Goal Setting for Real-Time Plan Recognition in Game-Based Learning. In: Proceedings of the Twenty-Third International Conference on Artificial Intelligence in Education (AIED-22), pp. 78–89. Springer, Durham, UK.

Goslen, A., Taub, M., Carpenter, D., Azevedo, R., Rowe, J., & Lester, J. (2024). Leveraging student planning in game-based learning environments for self-regulated learning analytics. *Journal of Educational Psychology*.

Hello GPT-4o. https://openai.com/index/hello-gpt-4o/

Hoffmann, J., & Nebel, B. (2001). The ff planning system: Fast plan generation through heuristic search. *J. Artif. Int. Res., 14*(1), 253–302.

Jiao, Y., Shridhar, K., Cui, P., Zhou, W., & Sachan, M. (2023). Automatic educational question generation with difficulty level controls. In: International Conference on Artificial Intelligence in Education, pp. 476–488. Springer

Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., et al. (2023). Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences., 103*, 102274.

Kim, Y.J., Goslen, A., Rowe, J., Mott, B., & Lester, J. (2023). Language model-based player goal recognition in open world digital games. In: Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE-23).

Koskinen, A., McMullen, J., Hannula-Sormunen, M., Ninaus, M., & Kiili, K. (2023). The strength and direction of the difficulty adaptation affect situational interest in game-based learning. *Computers & Education., 194*, 104694.

Kumaran, V., Carpenter, D., Rowe, J., Mott, B., & Lester, J. (2024). Procedural level generation in educational games from natural language instruction. *IEEE Transactions on Games*.

Leung, E. W. C., & Li, Q. (2003). A dynamic conceptual network mechanism for personalized study plan generation. In: Advances in Web-Based Learning-ICWL 2003: Second International Conference, Melbourne, Australia, August 18-20, 2003. Proceedings 2, pp. 69–80. Springer

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7871–7880. Association for Computational Linguistics, Online. https://doi.org/10.18653/v1/2020.acl-main.703 . https://aclanthology.org/2020.acl-main.703

MacNeil, S., Tran, A., Mogil, D., Bernstein, S., Ross, E., & Huang, Z. (2022). Generating diverse code explanations using the gpt-3 large language model. In: Proceedings of the 2022 ACM Conference on International Computing Education Research-Volume 2, pp. 37–39.

Min, W., Mott, B., Rowe, J., Liu, B., & Lester, J. (2016). Player Goal Recognition in Open-World Digital Games with Long Short-Term Memory Networks. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence(IJCAI-16), pp. 2590–2596. , New York.

Min, W., Mott, B., Rowe, J., Taylor, R., Wiebe, E., Boyer, K., & Lester, J. (2017). Multimodal goal recognition in open-world digital games. In: Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE-17), pp. 80–86. https://doi.org/10.1609/aiide.v13i1.12939

Pande, J., Min, W., Spain, R. D., Saville, J. D., & Lester, J. (2023). Robust team communication analytics with transformer-based dialogue modeling. In: International Conference on Artificial Intelligence in Education, pp. 639–650. Springer

Pintrich, P. R. (2000). The role of goal orientation in self-regulated learning. In: Handbook of Self-regulation, pp. 451–502. Elsevier, ???.

Plass, J. L., Mayer, R. E., & Homer, B. D. (2020). *Handbook of Game-based Learning*. Mit Press, ???

Polceanu, M., Porteous, J., Lindsay, A., & Cavazza, M. (2021). Narrative Plan Generation with Self-Supervised Learning. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI-21), pp. 5984–5992. AAAI Press, Virtual.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *JMLR*, *4*.

Rowe, J. P., Shores, L. R., Mott, B. W., & Lester, J. C. (2011). Integrating learning, problem solving, and engagement in narrative-centered learning environments. *International Journal of Artificial Intelligence in Education., 21*(1–2), 115–133.

Segedy, J. R., Kinnebrew, J. S., & Biswas, G. (2015). Using coherence analysis to characterize self-regulated learning behaviours in open-ended learning environments. *Journal of Learning Analytics., 2*(1), 13–48.

Shabrina, P., Mostafavi, B., Chi, M., & Barnes, T. (2023). Impact of learning a subgoal-directed problem-solving strategy within an intelligent logic tutor. In: International Conference on Artificial Intelligence in Education, pp. 389–400. Springer

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. CoRR. arXiv:1409.3215

Taub, M., Sawyer, R., Lester, J., & Azevedo, R. (2020). The impact of contextualized emotions on self-regulated learning and scientific reasoning during learning with a game-based learning environment. *International Journal of Artificial Intelligence in Education., 30*, 97–120.

Thoppilan, R., Freitas, D.D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H., Jin, A., et al. (2022). Lamda: Language models for dialog applications. CoRR. arXiv:2201.08239

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. arXiv:2307.09288.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In: Proceedings of the 31th Conference on Neural Information Processing SystemsNeurIPS

Winne, P., & Hadwin, A. (1998). *Studying as self-regulated learning* (pp. 291-318). Routledge

Winne, P. H. (2018). Theorizing and researching levels of processing in self-regulated learning. *British Journal of Educational Psychology., 88*(1), 9–20.

Winne, P., & Hadwin, A. (2008). The weave of motivation and self-regulated learning. In D. H. Schunk & B. J. Zimmerman (Eds.), *Motivation and self-regulated learning: Theory, research, and application.* New York, NY: Routledge.

Zimmerman, B. J. (2013). From cognitive modeling to self-regulation: A social cognitive career path. *Educational psychologist., 48*(3), 135–147.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.