ABSTRACT

GOSLEN, ALEX. Student Plan Recognition and Generation for Adaptive Support in Science Game-Based Learning Environments. (Under the direction of Dr. James Lester).

A key aim for science education is the improvement of scientific reasoning through inquiry-based learning which asks students to "think and act like scientists". This requires the regulation of specific skills and abilities such as identifying problems, generating hypotheses and evidence, and drawing conclusions. Goal setting and planning can help learners regulate their learning as they engage with scientific inquiry, especially within investigative exploration. Contemporary work on scaffolds for science inquiry-learning requires we (1) understand how students set goals and plans, (2) measure the quality of goals and plans, and (3) develop adaptive intelligent goals and planning scaffolds. The process of setting goals and creating plans is crucial for self-regulated learning (SRL), yet students often struggle to construct efficient plans and establish goals. Scaffolding incorporated into game-based learning environments creates an opportunity to collect data that captures students' goal setting and planning behaviors during challenging science inquiry learning scenarios that require effective SRL to achieve success. Additionally, the development of large language models offers new possibilities for enabling adaptive scaffolding of student learning in game-based learning environments.

The dissertation presents a data-driven analysis of students' explicit planning activities in two science game-based learning environments that can enable adaptive support for student planning, as well as their learning. In both games, students are provided with a planning support tool which aids them in externalizing their science-related goals and plans before putting them into action. By leveraging the trace data from these game-based learning environments, we present two plan scaffolding methodologies: (1) a student plan recognition framework and (2) a student plan generation framework. The student plan recognition framework aims to predict students' goals and planned action sequences given student gameplay sequences as input and student interactions with the planning tool as labels for both prediction tasks, which is an indication of *what* students are planning. The plan generation framework aims to automatically generate a set of low-level actions that contribute toward accomplishing a target goal given a sequence of prior gameplay events and prior completed goals, which is an indication of *how* students plan.

The preliminary research presented in this dissertation constructs and evaluates the student plan recognition framework in the science game-based learning environment called CRYSTAL ISLAND. The work has evaluated the student plan recognition framework using six supervised machine learning models and has also investigated additional machine-learning architectures and a technique for detecting when students enact all steps in their plans as methods for improving the framework. This dissertation research builds on preliminary results by testing the student plan recognition framework on another science game-based learning environment, FUTURE WORLDS.

Additionally, the dissertation presents a novel plan generation framework that leverages text-based representations of students' interactions within game-based learning environments to inform adaptive scaffolding of student goal setting and planning. We investigate the use of two pre-trained large language models, T5 and GPT-3.5, in the plan generation framework. We examine the generalizability of the plan generation framework, by comparing performance of models for both CRYSTAL ISLAND and FUTURE WORLDS. Finally, we present novel plan evaluation metrics and an empirical analysis of the plans generated by the models when evaluated against plans crafted by students during gameplay. This dissertation presents insights into how machine learning can be best leveraged to devise run-time adaptive support for student goal setting and planning in game-based learning environments.

Student Plan Recognition and Generation for Adaptive Support in Science Game-Based Learning Environments

by

Alexis Marie Goslen

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Computer Science

Raleigh, North Carolina
2024

APPROVED BY:

_____	_____
Dr. James Lester II Committee Chair	Dr. Wookhee Min
_____	_____
Dr. Thomas Price	Dr. Krista Glazewski

BIOGRAPHY

Alex Goslen was born and raised in Winston Salem, North Carolina. She attended Elon University and obtained two Bachelors of Science degrees in Mathematics and Computer Science. While at Elon, she participated in the undergraduate research program Elon College Fellows, under the mentorship of Dr. Shannon Duvall. It was through her undergraduate research that she first developed an interest in the intersection of education and technology.

After graduating from Elon in 2017, Alex moved to New York City to become a software developer for Credit Suisse. She mainly worked as an end-to-end developer for a data visualization web-based application for the credit e-trading team. At Credit Suisse, Alex was able to build her mentorship skills through leading internship teams and coordinating the winter internship program for sophomore women from CUNY schools. Alex deeply enjoyed this aspect of her job and was fueled by helping foster interest in computer science for the next generation of women in STEM and Finance.

In 2020, Alex relocated back to North Carolina to begin her Ph.D. in Computer Science at North Carolina State University. She began as a graduate research assistant in 2021 for the IntelliMedia Lab under the supervision of Dr. James Lester. During her time in graduate school, Alex was granted the opportunity to work on several projects related to student planning and reflection, student affect, and AI education, while helping to develop games, tools, and activities for students.

Alex has accepted a Tenure Track appointment to serve as Assistant Professor of Computer Science at Elon University, where she will combine her knowledge and experience to teach and mentor undergraduate students.

ACKNOWLEDGEMENTS

I would first like to thank my advisor, Dr. James Lester, for all his support and guidance during my Ph.D. program. He has modelled exemplary research collaboration and management and has provided much needed encouragement throughout my time at NC State. In addition, I would like to thank my other doctoral committee members, Dr. Wookhee Min, Thomas Price, and Krista Glazewski, for their insights, questions, and feedback while completing this dissertation.

This work would not have been possible without our collaborators at the University of Central Florida. A special thank you to Michelle Taub, Megan Wiedbusch, Daryn Dever, and Roger Azevedo for your guidance through the fields of educational psychology and metacognition. Another special thank you goes out to the collaborators on other projects I have participated in during my graduate school career. In particular, thank you to Jaclyn Ocumpaugh, Stephen Hutt, Nidhi Nasiar, Andres Felipe Zambrano, Veronica Catete, and Danielle Boulden.

The IntelliMedia Group provided an exciting and creative workplace, where there was no shortage of laughter or millipedes. I am deeply grateful to have had the opportunity to work with this group. In particular, I would like to thank Dr. Bradford Mott and Dr. Dan Carpenter for being a sounding board for my ideas, whims, and emotions throughout this process. To all the students old and new in the IntelliMedia Group, thank you for your time, effort, and support for one another. Additionally, a million thanks go to Dr. Jessica Vandenberg, who, with her sense of humor, empathy, and humanity, helped me grow as a person and researcher.

Finally, I would like to thank all of my friends and family who have provided an endless amount of support through this journey. Thank you to my parents, Mark and Robyn, and my little sister, Gigi, for all the check-in phone calls. Thank you to Ally Limke, Lauren Alvarez, Derek Martin, and Anisha Gupta for lending a listening ear, being a work buddy, and providing encouragement to me over the years. Finally, I would like to thank my partner, Rickey. Words cannot describe how much your support means to me. Thank you for not only all of the meal prep, but also your patience and endless support (even if you say it's the bare minimum).

TABLE OF CONTENTS

List of Tables vii
List of Figures ix
Chapter 1. Introduction 1
1.1. Thesis Statement and Hypotheses 4
1.2. Contributions 5
1.2. Organization 6
Chapter 2. Background and Related Work 7
2.1. Self-Regulated Learning 7
2.2. SRL and Game-Based Learning Environments 8
2.3. Scientific Reasoning 9
2.4. Predicting Plans in Online Environments 11
2.5. Plan Generation 12
2.6. Large Language Models 13
Chapter 3. Enabling Adaptive Learning Environments with Student Plan Recognition and Generation 15
3.1. Operationalization of Planning 15
3.2. Student Planning Scenarios 16
3.3. Plan-Enhanced Game-Based Learning Environment 17
3.4. Student Planning Scenarios with an Adaptive Learning Environment 20
Chapter 4. Crystal Island Data Corpora 22
4.1 Planning Support Tool in Crystal Island 22
4.2. Crystal Island Study Data 25
Chapter 5. Future Worlds Data Corpora 26
5.1. Planning Support Tool in Future Worlds 26
5.2. Future Worlds Study Procedure 27
Chapter 6. Student Plan Recognition 29
6.1. Developing a Baseline Student Plan Recognition Framework 29
6.1.1. Event Subsequence Partition Method 29
6.1.2. Prediction Tasks 31

6.1.3. Results 33
6.1.4. Discussion and Limitations 34
6.2. Improving the Student Plan Recognition Framework 36
6.2.1. Methods 36
6.2.2. Results 37
6.2.3. Discussion and Limitations 38
6.3. Applying Student Plan Recognition to Future Worlds 39
6.3.1. Methods 40
6.3.2. Results 42
6.3.3. Discussion and Limitations 44
Chapter 7. Student Plan Generation 46
7.1. Plan Generation Framework Construction 46
7.1.1. Methods 47
7.1.2. Results 51
7.2. Applying Student Plan Generation to Future Worlds 55
7.2.1. Methods 55
7.2.2. Results 57
7.3. Transfer Learning for Plan Generation in Future Worlds 60
7.3.1. Methods 61
7.3.2. Results 62
7.4. Examples of Generated Plans 64
7.4.1. Crystal Island Generated Plans 64
7.4.2. Future Worlds Generated Plans 67
7.5. Discussion 69
7.5.1. Training Approach Comparison Implications 69
7.5.2. Implications for Adaptive Learning Environments 70
Chapter 8. Plan Evaluation in Science Game-Based Learning 74
8.1. Plan Generation Framework Construction 74
8.1.1. Measuring Goal Setting and Planning Quality 74
8.1.2. Plan Complexity 75
8.1.3. Action-Goal Cohesion 76

8.1.4. Temporal Coherence 76
8.2. Applying Plan Evaluation Metrics to Crystal Island 77
8.2.1. Methods 78
8.2.2. Results 82
8.3. Applying Plan Evaluation Metrics to Future Worlds 84
8.3.1. Methods 85
8.3.2. Results 88
8.4. Discussion 91
Chapter 9. Implications of Adaptive Learning Environments with Student Plan Recognition and Generation 93
9.1. Student Planning Scenarios Revisited 93
9.1.1. Student A (Adam) 94
9.1.2. Student B (Barbara) 97
9.1.3. Student C (Carl) 101
9.2. Theoretical Implications 104
9.2.1. Implications for Learning Analytics 104
9.2.2. Implications for SRL Skill Intervention 106
Chapter 10. Conclusion 109
10.1. Hypotheses Revisited 109
10.2. Summary 113
10.3. Future Work 114
References 116

LIST OF TABLES

Table 2.1 Fischer et al.'s epistemic activities for scientific reasoning 10

Table 6.1 F-measure results for each predictive model and goal category in the goal recognition task. Results are averaged across the nested 5-fold cross validation, with distribution representing the test set. Bolded values indicate highest performance. 33

Table 6.2 F-measure results for each predictive model and planned action sequence cluster in the planned action sequence recognition task. Results are averaged across the nested 5-fold cross validation, with distribution representing the test set. Bolded values indicate highest performance. 34

Table 6.3 Goal recognition F-measure results comparing the single-task LSTM with the multi-task LSTM and the enhanced framework incorporating automatic plan completion detection. Results are averaged across the nested 5-fold cross validation, with distribution representing the test set. 37

Table 6.4 Planned action sequence recognition F-measure results comparing the single-task LSTM with the multi-task LSTM and the enhanced framework incorporating automatic plan completion detection. Results are averaged across the nested 5-fold cross validation, with distribution representing the test set. 38

Table 6.5 F-measure results for each predictive model and goal category in the goal recognition task in FUTURE WORLDS. Results are averaged across the nested 3-fold cross validation, with distribution representing the test set. Bolded values indicate highest performance. 42

Table 6.6 F-measure results for each predictive model and planned action sequence class in the planned action sequence recognition task for FUTURE WORLDS. Results are averaged across the nested 3-fold cross validation, with distribution representing the test set. Bolded values indicate highest performance. 43

Table 6.7 Goal recognition F-measure results comparing the single-task LSTM with the multi-task LSTM and the enhanced framework incorporating automatic plan completion detection for FUTURE WORLDS. Results are averaged across the nested 3-fold cross validation, with distribution representing the test set. 43

Table 6.8 Planned action sequence recognition F-measure results comparing the single-task LSTM with the multi-task LSTM and the enhanced framework incorporating automatic plan completion

detection for FUTURE WORLDS. Results are averaged across the nested 3-fold cross validation, with distribution representing the test set. 44
Table 7.1 An example GPT-3.5 prompt used for the CRYSTAL ISLAND learning environment implementation. Note: event sequences and example plans have been reduced for space. Actual prompts included event sequences of length 30 and two example plans. 50
Table 7.2. High-level action category distribution for each high-level goal in the CRYSTAL ISLAND planning tool across student, T5, and GPT-3.5 plans. 54
Table 7.3 An example GPT-3.5 prompt used for the FUTURE WORLDS learning environment implementation. Note: event sequences and example plans have been reduced for space. Actual prompts included event sequences of length 20 and two example plans. 56
Table 7.4. High-level action category distribution for each high-level goal in the FUTURE WORLDS planning tool across student, T5, and GPT-3.5 plans. 60
Table 7.5. High-level action category distribution for each high-level goal in the FUTURE WORLDS planning tool of transfer-learned GPT-3.5 generated plans (GPT-3.5-T) compared to student, T5, and GPT-3.5 plans. 63
Table 7.6. Example student-created plans and LLM-generated plans for a target goal for each high-level goal in the CRYSTAL ISLAND planning tool. 65
Table 7.7. Example student-created plans and LLM-generated plans for a target goal for each high-level goal in the FUTURE WORLDS planning tool. 68

LIST OF FIGURES

Figure 3.1 Overview of a game-based learning environment that incorporates a plan recognition framework. 18
Figure 3.2 Overview of a game-based learning environment that incorporates a plan generation framework. 19
Figure 4.1 (Left) Crystal Island game-based learning environment, (right) Conversation with an NPC in Crystal Island. 22
Figure 4.2. Example planning support tool interaction in Crystal Island. 24
Figure 5.1. An example level of the Future Worlds learning environment. 27
Figure 5.2. Example plans in the planning support tool. 28
Figure 6.1 Example single-task model architecture. 31
Figure 6.2 Multi-task model architecture. 37
Figure 7.1 Distribution metrics for T5 generated plans for CRYSTAL ISLAND. 51
Figure 7.2 Distribution metrics for GPT-3.5 generated plans for CRYSTAL ISLAND. 52
Figure 7.3 Distribution metrics for T5 generated plans for FUTURE WORLDS. 58
Figure 7.4 Distribution metrics for GPT-3.5 generated plans for FUTURE WORLDS. 59
Figure 7.5 Distribution metrics for transfer-learned GPT-3.5 generated plans for FUTURE WORLDS. 62
Figure 8.1 State model of Fischer et al. (2014)'s epistemic activities used in scientific reasoning and argumentation. 77
Figure 8.2 Analytical mapping matrix of actions (columns) and goals (rows). 78
Figure 8.3 Close up sub-section of the mapping matrix for examples of Goal category-goals (rows) and Action category-actions (Columns) to identify scientific reasoning epistemic activity (cell color) and plan relevancy (cell opacity). 79
Figure 8.4 State model of expected epistemic activities progression in CRYSTAL ISLAND. 81
Figure 8.5. Distribution of plan size for student, T5, and GPT-3.5 generated plans for CRYSTAL ISLAND. 82
Figure 8.6. Distribution of action-goal cohesion score for student, T5, and GPT-3.5 generated plans for CRYSTAL ISLAND. 83

Figure 8.7. Distribution of temporal coherence score for student, T5, and GPT-3.5 generated plans for CRYSTAL ISLAND. 84
Figure 8.8. Analytical mapping matrix of actions (columns) and goals (rows). 85
Figure 8.9. Close up sub-section of the mapping matrix for examples of Goal category-goals (rows) and Action category-actions (Columns) to identify scientific reasoning epistemic activity (cell color) and plan relevancy (cell opacity). 87
Figure 8.10. State model of expected epistemic activities progression in FUTURE WORLDS. 88
Figure 8.11. Distribution of plan size for student, T5, and GPT-3.5 generated plans for FUTURE WORLDS. 89
Figure 8.12 Distribution of action-goal cohesion score for student, T5, and GPT-3.5 generated plans for CRYSTAL ISLAND. 90
Figure 8.13 Distribution of temporal coherence score for student, T5, and GPT-3.5 generated plans for FUTURE WORLDS. 90
Figure 9.1 An example illustration of a planning instance with epistemic activity mapping and scoring (left), as well as the epistemic activity key (right). 94
Figure 9.2 Adam's planning progression with corresponding scores. 95
Figure 9.3 The T5-generated plans for Adam's planning progression with corresponding scores. 96
Figure 9.4 The GPT-3.5-generated plans for Adam's planning progression with corresponding scores. 96
Figure 9.5 Barbara's planning progression with corresponding scores. 98
Figure 9.6 The T5-generated plans for Barbara's planning progression with corresponding scores. 99
Figure 9.7 The GPT-3.5-generated plans for Barbara's planning progression with corresponding scores. 100
Figure 9.8 Carl's planning progression with corresponding scores. 101
Figure 9.9 The T5-generated plans for Barbara's planning progression with corresponding scores. 102
Figure 9.10 The GPT-3.5-generated plans for Carl's planning progression with corresponding scores. 103

CHAPTER 1

INTRODUCTION

Observing students' cognitive and metacognitive processes poses a challenge in educational settings, with inferences on learning mainly being based on changes in output, temporal duration of events, and observable behaviors (Winne, 2018). Game-based learning environments inherently provide means for measuring these factors through automatic trace data logging (Taub & Azevedo, 2019). Data from game-based learning environments also allow for predicting students' various cognitive and metacognitive states, enabling adaptation to individual students in real-time. With tools, hints and difficulty adjustments, game-based learning environments are able to customize support for students' individual problem-solving (McLaren & Nguyen, 2023). Recognizing student strategies and planning is a key component that drives adaptive scaffolding in game-based learning environments (Azevedo & Wiedbusch, 2023). However, prior work in recognizing such strategies often relies on multimodal data such as eye gaze and facial expression (D'Mello & Graesser, 2023). Little work has been done to explore the potential use of trace data alone to identify metacognitive behaviors during students' problem solving in game-based learning environments (Dever et al., 2023b).

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, 2018; Winne & Hadwin, 1998, 2008). Providing support for students' SRL processes can greatly assist them in navigating challenging learning tasks, such as solving scientific problems (Dever et al., 2022).

The application, practice, and development of scientific inquiry skills (e.g., identifying the problem, collecting evidence, generating hypotheses, making inferences, drawing conclusions) are fundamental to learning the nature of science and science content by allowing students to "think and act like scientists" (OECD, 2022; Omarchevska et al., 2022; Wallace & Coffey, 2019). The application of scientific inquiry skills requires students to use SRL to engage in the formulation of goals, develop plans to meet those goals, and monitor and adapt their (meta)cognition based on their progress towards achieving goals (Winne & Azevedo, 2022; Winne, 2018). However, many students lack the necessary SRL skills to do so effectively (Azevedo et al., 2017; Lester et al.,

2013). Contemporary work in SRL requires we: (1) understand how students set goals and plans, (2) measure the quality of goals and plans, (3) utilize trace data to capture the dynamics of plan enactment via SRL behaviors, and (4) develop adaptive intelligent planning scaffolds. These requirements are non-trivial as SRL is a temporally dynamic, contextual, and conditional process (Ben-Eliyahu & Bernacki, 2015).

For successful scientific reasoning to occur, students need to be effective self-regulators in which students can monitor their progress towards learning outcomes and change how they enact scientific reasoning processes to successfully achieve those outcomes (Greene et al., 2018; Omarchevska et al., 2022; Sinatra & Taasoobshirazi, 2017). However, scientific reasoning is a notoriously difficult process for students to engage in as they are required to continuously engage in information gathering, hypothesis generation, hypothesis testing, and hypothesis adaptation while increasing their scientific knowledge and modifying mental models of complex scientific topics (e.g., microbiology; Klahr & Dunbar, 1988; Van Mil et al., 2016; Woolley et al., 2018). Students' ability to engage in successful scientific reasoning is further hindered by students' inability to self-regulate either due to their inability to efficiently and effectively deploy SRL processes (e.g., planning, goal-setting) in addition to their lack of knowledge about SRL processes and strategies (Dever et al., 2022; Omarchevska et al., 2022; Dever et al., 2023a; Greene et al., 2024). Because of the compounding difficulties in engaging in both effective SRL and scientific reasoning processes, it is essential that students are externally supported by tools or scaffolds within the learning environment in which learning is taking place.

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. Student interactions with game-based learning environments generate a significant amount of student log data. The log data is essential for training machine learning models to inform adaptive scaffolding tailored to students' individual needs and further enhance student learning experiences. A novel example is when a planning support tool was incorporated into a science game-based learning environment for middle school (Goslen et al., 2022a, Goslen et al., 2022b). Through a block-based drag and drop tool, students were able to create their own plans by selecting goals and planned actions to achieve that goal. By leveraging the trace data from students' planning activities, a student plan recognition framework was constructed to predict students' goals and

planned actions at given points in gameplay, enabling adaptive support for students' planning. However, in this case and in most situations, only a subset of this data can be utilized by machine learning models and often has to be converted into a numerical format.

Recent developments in large language models (LLMs) have shown new possibilities for analyzing student log data 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 environments, such as generating practice problems, detailed step-by-step solutions, and comprehensive explanations (Kasneci et al., 2023).

This dissertation explores how we can predict and generate plans to support student SRL processes and, thereby, enhance their ability to solve scientific problems. In particular, it addresses two main goals: (1) examining the robustness of a plan recognition framework by comparing performance across two science learning environments and (2) constructing a framework that automatically generates plans within science game-based learning environments that can be used to inform adaptive scaffolding for student SRL. The dissertation also explores the development of theory-driven plan quality metrics to use for student and framework evaluation. The dissertation focuses on two science learning environments, CRYSTAL ISLAND and FUTURE WORLDS, that have a planning support tool incorporated into the game to allow students to externalize their planning throughout gameplay. We examine multi-label, multi-task learning and data pre-processing for the student plan recognition framework. With the student plan generation framework, we compare the performance of fully-supervised, few-shot, and transfer learning methodologies using LLMs. Both corpora are used to evaluate the student plan recognition framework and student plan generation framework to examine the generalizability of methods. Results from experiments will be discussed individually, followed by a description of student planning and how the student plan recognition and generation frameworks can enable adaptive learning environments to enhance student learning with SRL scaffolding.

1.1 Thesis Statement and Hypotheses

This dissertation investigates the following thesis statement:

> *Plan recognition and plan generation frameworks leveraging classical machine learning techniques and large language models, respectively, can be used to automatically predict student plans and generate valid plans to provide adaptive support and furthermore can be applied to multiple student populations and multiple science game-based learning environments.*

To investigate this thesis statement, we evaluate the following hypotheses:

- H.1. Applying the plan recognition framework methods to the FUTURE WORLDS dataset will yield similar trends in results compared to evaluations of the framework using CRYSTAL ISLAND data.
    - H.1.1. Because of the smaller size of the FUTURE WORLDS dataset, the traditional machine learning methods will be lower in predictive performance but will still have LSTMs outperform other models.
    - H.1.2. Incorporating both multi-task learning and automatic plan completion detection will improve predictive performance.

- H.2. Large language models will demonstrate potential for generating student plans given a set of game actions, completed goals and a target goal for both learning environments despite small datasets.
    - H.2.1. Given that T5 uses a fully-supervised methodology, it will successfully generate plans similar to students' plans for both learning environments.
    - H.2.2. Using a few-shot learning approach, GPT-3.5 will generate valid plans that are more distinct from student plans for both learning environments.

- H.3. The novel plan evaluation metrics that have been devised in collaboration with researchers in the learning sciences will provide further insight into the quality of both student and language model generated plans.
    - H.3.1. In terms of temporal coherence, students will achieve higher scores for their plans than the generated plans for both learning environments.

    - H.3.2. Action-goal cohesion performance will be similar for student plans and generated plans across learning environments.
    - H.3.3. Generated plans will be much longer than student plans, demonstrating more plan complexity.

- H.4. Applying the plan generation framework methods to the FUTURE WORLDS dataset will yield similar trends in results compared to the CRYSTAL ISLAND dataset.
    - H.4.1. Replicating plan generation methods in FUTURE WORLDS will produce similar results to CRYSTAL ISLAND in terms of T5 and GPT-3.5 performance.
    - H.4.2. LLMs will enable transfer learning approaches for the task of plan generation between science learning environments.

1.2 Contributions

This dissertation makes the following contributions:

- A novel student plan recognition framework that leverages students' externalized plans from a game-based learning environment (Goslen et al., 2024b; Goslen et al., 2022a). This includes an improvement on the student plan recognition framework through automatic detection of plan completion through trace data analysis, as well as an evaluation of multi-task learning against single-task learning models (Goslen et al., 2022b).

- An extension of the student plan recognition framework to FUTURE WORLDS, investigating the generalizability and robustness of the framework across science learning environments.

- A novel student plan generation framework that leverages textual representations of students' gameplay data as input to LLMs to predict a series of low-level actions that can be enacted in gameplay. This includes a comparison of LLM performance for the task of plan generation, specifically with a focus on the potential for adaptive scaffolding in game-based learning (Goslen et al., 2024a).

- An empirical investigation of the generalizability and robustness of the framework across science learning environments through the extension of the student plan generation

framework to FUTURE WORLDS. This work includes an empirical exploration of transfer learning methods utilizing LLMs across science game-based learning environments.

- The development of novel plan quality metrics aligned with scientific reasoning theory (Wiedbusch et al., 2024). Additionally, an empirical assessment of both student and machine learning model-generated plans for both CRYSTAL ISLAND and FUTURE WORLDS, with insights into the effectiveness of plan quality metrics for evaluating student and model-made plans, as well as implications for effective tool and game design.

1.3 Organization

The remainder of this dissertation is organized as follows. Chapter 2 provides background on SRL, scientific reasoning, plan recognition, plan generation, and large language models. While the discussion will focus mainly on game-based learning, it will also encompass broader research areas. Chapter 3 presents an overview of both the plan recognition and plan generation frameworks, which aim to predict students' goals and planned actions for that goal at various points of gameplay. The CRYSTAL ISLAND game-based learning environment, where the student plan recognition framework was created, is described in Chapter 4. We also include example scenarios of student gameplay to demonstrate how machine learning models can be derived from the trace data of game-based learning environments. Chapter 5 introduces the learning environment that will be utilized in proposed work to examine generalizability of both the plan recognition and plan generation frameworks, FUTURE WORLDS. Chapter 6 describes the methods and findings of constructing a student plan recognition framework in CRYSTAL ISLAND, as well as the results from applying those methods to the FUTURE WORLDS learning environment. Chapter 7 describes methods and findings from the LLM-based plan generation framework applied to both data corpora, including example plans generated for each learning environment and a discussion on the implications for real-time adaptive scaffolding. Novel plan evaluation metrics are presented in Chapter 8, with an empirical application of these metrics to student and LLM-generated plans included. Chapter 9 revisits the student planning scenarios presented in Chapter 3 to discuss the theoretical implications of incorporating the frameworks into learning environments. Chapter 10 addresses the hypotheses designed to support the thesis statements, summarizes the contributions of this work, and discusses possible directions for future work.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Self-Regulated Learning

Self-regulated learning (SRL) involves students actively engaging in their learning by deploying strategies for monitoring and controlling their emerging understanding of complex topics (Greene et al., 2024; Winne, 2018; Winne & Azevedo, 2022). SRL is a cyclical, recursive process that occurs throughout four phases during learning: task understanding, planning and goal setting, using learning strategies, and reflecting and making adaptations. According to the Conditions, Operations, Products, Evaluations, and Standards (COPES) model of SRL (Winne & Hadwin, 1998; Winne & Hadwin 2008), students select strategies based on how conditions, operations, products, evaluation, and standards affect their task performance. Students engage in several phases while completing a task. For example, a student formulates a plan for accomplishing a task and sets goals for executing those plans. While using the planned strategies, they monitor their understanding of the content by making a judgment-of-learning (metacognitive monitoring strategy) and deem that they do not understand the content. This leads them to revisit their original plan and make adaptations to it. Thus, students are engaging in SRL by constantly monitoring their emerging understanding and performance and controlling their subsequent actions. This cyclical process can occur several times during learning, making it important to be able to identify and keep track of such instances when examining SRL.

Scientific inquiry, in particular, requires students to have motivation for gaining knowledge and thus, requires effective planning strategies to enact scientific investigations (Graesser & McNamara, 2010). Supporting goal setting and planning in such learning has demonstrated student success and aided in positive emotions during learning (Boekaerts & Pekrun, 2015). However, there remain key questions about how students engage in such processes, assuming they already know how to effectively set goals, monitor their progress, and adapt their strategies to new information. Although student planning and goal setting are key components of SRL, there has been little research done on explicit goal setting and planning, particularly in online learning environments (Winne, 2018). Work has been done in biology classrooms to predict students who

are likely to perform poorly through trace data and improve their SRL through training materials (Cogliano et al., 2022). Additional work has been done in an intelligent logic tutor to trace students' learning and provide strategies to students in real-time (Shabrina et al., 2023). Our work aims to analyze students' explicit planning behaviors through trace logs, from which we can extract key features of student planning and make conclusions about how planning impacts their gameplay and learning

2.2 SRL and Game-Based Learning Environments

Game-based learning environments provide capabilities for tracking SRL processes, particularly in science-based narrative game environments that challenge students to use scientific reasoning skills. Planning and goal setting are key to SRL in game-based learning environments (Plaas et al., 2020) because they require the student to understand the given task and map out how to accomplish it. Despite their importance, little work has examined key aspects of planning and goal setting in environments where students can explore game features, express agency, and receive little scaffolding, which may impact learning (Taub et al., 2020). Trace data from game-based learning environments is apt to provide insights into students' SRL behaviors due to its temporal nature, potential for automatic validation, and capabilities for adaptivity, which are key features for analyzing SRL (Bernacki, 2017; Taub et al., 2020). This dissertation addresses vital theoretical, conceptual, and analytical questions about planning and goal-setting that must be systematically examined in game-based learning environments. More specifically, we address issues related to the quantity, quality, and temporal enactment of these processes during gameplay from trace data, their sequencing (with other self-regulatory processes), and their influence on learning within game-based learning environments.

This work analyzes middle school students' use of a novel planning support tool embedded into two science game-based learning environments, CRYSTAL ISLAND and FUTURE WORLDS. As students navigate through each learning environment, they are periodically prompted to externalize their problem-solving goals and plans through a drag-and-drop planning tool. This tool is grounded in Winne and Hadwin's COPES model of SRL (1998, 2008), which defines four phases of SRL: problem understanding and task definition, goal setting and plan generation, enactment of learning strategies, and reflection and adaptation. In addition, COPES impact how a learner self-regulates while engaging in a task. The trace data collected by the learning environment allow us

to capture three of these phases of students' SRL processes (Dever et al., 2022, 2023a): the externalized goals and plans represent the goal setting and plan-generation phase; the enactment of learning strategies is represented through students' problem-solving actions in the learning environment; and through students' planning instances with and without modification, we can also potentially identify the reflection and adaptation phase.

It can be challenging for researchers to measure learners' SRL during gameplay because games do not typically allow for overt measures of SRL processes and require inference making (see Azevedo & Taub, 2020). There are several advantages to collecting multimodal multichannel data (e.g., trace data, eye tracking, face videos, physiological sensors) during learning and gameplay, such as having unobtrusive, fine-grained measures of learning and self-regulated learning processes (Azevedo & Gasevic, 2019). Digital games typically collect behavioral logs of gameplay actions, which provide detailed traces of how students navigate through the game. While this generates a lot of data for analysis, it does require researchers to make inferences regarding their self-regulatory behaviors. Although other data channels allow for explicit externalization of metacognition, such as using think-aloud data (i.e., players are stating what they are doing), it is not feasible to use this kind of data in the classroom.

Therefore, by developing a set of measures or gameplay tools that describe students' learning and SRL processes during gameplay, we are fostering SRL (e.g., tools for planning and reflection) and science learning while also making fewer inferences about their actions. We have already developed a tool to foster students' reflections during gameplay (Carpenter et al., 2020, 2021; Cloude et al., 2021; Geden et al., 2021; Gupta et al., 2023) with CRYSTAL ISLAND. The planning tool we used for this study serves a similar role for planning—as a teaching tool to promote planning during gameplay, as well as a research tool for examining students' planning and goal-setting behaviors during gameplay.

2.3 Scientific Reasoning

Scientific reasoning and the skills commonly associated with it (e.g., making hypotheses, drawing conclusions) are often conceptualized within cycles of scientific inquiry. That is, the cycle one might go through to problem solve or develop new conceptual knowledge based on empirical testing. There have been several frameworks that model scientific reasoning and inquiry including Scientific Discovery as Dual Space (SDSS; Klahr & Dunbar, 1988), OPIRR (Orientation,

Preparation, Implementation, Reporting, and Reflection) model (Janssen et al., 2019), and Fischer et al.'s (2014) epistemic activities of scientific reasoning. These conceptualizations' descriptions refer to skill sets, linked domain-specific and domain general knowledge, and a cycle of stages that one goes through to generate and modify their knowledge production during complex learning (Janssen et al., 2019). Considering the complexity of many of these scientific reasoning tasks and activities, research has repeatedly shown that students struggle with efficiently and accurately engaging in scientific reasoning (Van Mil et al., 2016; Woolley et al., 2018).

Table 2.1. Fischer et al.'s epistemic activities for scientific reasoning.

Epistemic Activity	Definition
Problem Identification	A problem representation is constructed by the learner based on a study of a scenario in which a perceived discrepancy or weakness exists with respect to the explanation that is currently provided for a given problem.
Questioning	Based on the problem representation developed during problem identification, the learner identifies question(s) that may be refined later.
Hypothesis Generation	The learner derives possible answers to the questions from models, available theoretical frameworks, or empirical evidence they are aware of. If prior knowledge is not enough to allow for prediction, the question may be refined in questioning or there may be an exploratory approach of evidence generation.
Construction & Redesign of Artefacts	The learner constructs and artefact (e.g., a prototype or axiomatic system) typically based on current theoretical models to be tested in an authentic environment.
Evidence Generation	The learner collects evidence in various approaches including (1) conducting hypothetico-deductive experimental studies; (2) Induction using observations; (3) General Observations; (4) Deductive Reasoning; (5) Compiling fields of research. What counts as evidence differs among domains.
Evidence Evaluation	The learner assesses the degree to which evidence that has been generated supports a hypothesis/claim/theory.
Drawing Conclusions	The learner uses different pieces of evaluated evidence to either support or disprove a hypothesis. This can differ among domain and may refer to the compiling of multiple pieces of evidence for integration.
Communicating & Scrutinising	The learner interacts with others to share their findings and converse about the practice, methodology, results, conclusion, and implications.

The work presented in this dissertation focuses primarily on Fischer's framework of scientific reasoning activities (Fischer et al., 2014) but draws inspiration from the other models as well (such as the cyclic nature of the OPIRR model or the non-linearity of the SDSS model). According to this framework, there are 8 epistemic activities that one might engage in during scientific reasoning. These are defined in Table 2.1.

2.4 Predicting Plans in Online Environments

In the field of SRL, there has been limited effort in creating computational models to recognize and adapt to student plans in online learning environments. An exception to this trend is an intelligent physics tutoring system called Andes. This system employed Bayesian networks to construct models of student plans, enabling predictions of their actions during problem-solving (Conati et al, 2002). The work with Andes showcases successfully using plan recognition to facilitate tailored hints to individual students.

More substantial progress has occurred in the domain of player modeling, where data-driven methods have demonstrated potential in adapting digital games to players' intentions (Hooshyar et al., 2018). Recognizing players' strategies in digital games gives valuable insights into real-time adjustments to individual players' behaviors, performance, and interests (Duarte et al., 2020; Sukthankar et al., 2014). Theory of Mind, the ability to understand and predict intentions and cognitive attributes, has been used in player modeling applications (Shergadwala et al., 2021). Many graph or hierarchy-based methodologies have been utilized for such plan recognition models (Bisson et al., 2015; Rabkina et al., 2022; Rabkina et al., 2021). Theory of Mind is a crucial metacognitive process and has also been addressed in recent work in digital games to enhance plan recognition tasks (Boeda, 2021; Ware & Siler, 2021; Weerawardhana et al., 2021). While these approaches have demonstrated efficacy in digital games featuring predefined states, little exploration has been done into the potential of using plan recognition to adapt open-world game-based learning environments for individual students.

Open-world learning environments often require the application of SRL skills because they allow students to explore the environment at their own pace, by choosing their own goals and identifying paths to achieve them (Squire, 2008). These types of environments present a unique challenge in that student actions in the game are individualistic and highly exploratory. Prior work explored student goal recognition, a sub-area of plan recognition, in open-world game-based

learning environments and found Long Short-Term Memory networks (LSTMs) to perform best when compared to several baseline models (Min et al., 2016; Min et al., 2017). LSTMs, a type of recurrent neural network, can identify dependencies within a sequence of events (Hochreiter & Schmidhuber, 1997). LSTMs are adept at handling uncertain, probabilistic data. Additionally, the authors found that one-hot encoding vectors were effective for representing game trace data for the task of predicting student goals (Min et al., 2017).

2.5 Plan Generation

Classical Artificial Intelligence (AI) planning traditionally revolves around generating sequences of actions within a given environment, a concept we extend to the task of plan generation (Ghallab et al., 2004). The field of automated planning has explored various techniques for plan generation over many years (Blum, 1997; Bercher et al., 2019; Hoffman et al., 2001). Recent approaches to plan generation include hierarchical planning (Bartak et al., 2001) and narrative plan generation employing self-supervised learning (Polceanu et al., 2021). These techniques have been applied in educational settings where the environment is finite and relatively small. For instance, graph networks have been employed to evaluate and enhance student planning in systems like Betty's Brain (Segedy et al., 2015) and have also been used for generating study plans (Leung et al., 2003). However, plan generation tasks within game-based learning environments present a distinctive challenge due to the scale and open-world nature of the environment.

In order to generate viable plans for students in real-time, it is essential to have an understanding of students' underlying problem-solving strategies. Students' gameplay within game-based learning environments such as CRYSTAL ISLAND or FUTURE WORLDS is highly individualistic and characterized by exploratory behavior. To date, large language models have not been previously used to encode students' problem-solving 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.

2.6 Large Language Models

Recently, the field of natural language processing (NLP) has advanced significantly through the development of large language models. These models have proven successful in various NLP tasks such as 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). The proposed research aims to examine the effectiveness of large language models in the context of plan generation within game-based learning environments.

This dissertation work will leverage two language models for the task of plan generation, T5 (Raffel et al., 2020) and GPT-3.5 (Brown et al., 2020). T5 is a language model based on the transformer architecture, 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 (Cochran et al., 2023; Macneil et al., 2022; Kasneci 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 (OpenAi, 2024) has 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.

CHAPTER 3

ENABLING ADAPTIVE LEARNING ENVIRONMENTS WITH STUDENT PLAN RECOGNITION AND GENERATION

This chapter gives an overview of how plan recognition and generation models can enable adaptive learning environments. We first describe how planning is operationalized in game-based learning environments for this dissertation. Then, we provide example scenarios of students' planning in game-based learning to demonstrate what goals and plans look like in these environments without adaptive support. Next, we describe a proposed adaptive learning environment that incorporates plan recognition and generation to improve student learning. Finally, we discuss the student scenarios further to illustrate how an adaptive learning environment with a plan recognition and generation framework may improve the student experience.

3.1 Operationalization of Planning

In this dissertation work, we conceptualize planning from a SRL perspective. Based on theories of self-regulated learning (Pintrich, 2000; Winne, 2018; Winne & Hadwin, 1998, 2008; Zimmerman, 2000), planning is a key component of self-regulation and involves several components, such as goal setting and prior knowledge activation. However, as with self-regulated learning in general, there is no "correct" way to plan. Thus, students can choose to enact planning strategies in many ways. Although this provides students with agency (Taub et al., 2020) to plan, research demonstrates students achieve higher learning gains when setting appropriate subgoals with the support of a pedagogical agent during learning with a computer-based learning environment (Harley et al., 2018). Thus, we adopt this conceptualization of planning in the preliminary work by giving students some guidance towards planning with the planning tool in the game-based learning environment (e.g., a tool for fostering setting goals and subgoals as opposed to a pedagogical agent), however we do not require students to use the tool as a way to promote more agency than previous work. To demonstrate this, we provide three planning scenarios that demonstrate different types of planning with the planning tool in CRYSTAL ISLAND. These

illustrative scenarios are based on students' interactions with the planning tool and the names of students have been anonymized.

3.2 Student Planning Scenarios

In this first example scenario, we consider a student named Adam who accessed the planning tool five times throughout gameplay. In each of these planning tool interactions, Adam either created one plan or edited the existing plan. After completing the tutorial, they created a plan with the goal of "*learn about disease*" with two planned actions: "*speak with patients*" and "*speak with lead scientist*." Then, Adam immediately went to the infirmary and spoke with Kim and the patients. After a few minutes, Adam left the infirmary and wandered outside. They were then prompted to plan again because speaking with the camp nurse is a major plot point. During this planning instance Adam removed the action "speak with patients" from the plan and closed the tool. Immediately after this, they went to the lead scientist's house, read some informational text, and spoke with the lead scientist. Then, Adam began collecting items to test for viruses or bacteria. Once going to the laboratory, Adam got a positive scan result on an egg for pathogenic bacteria. Because this is a major plot point of the game, Adam was prompted to plan. In this planning tool instance, they updated their plan to remove "*speak with lead scientist*" and added "*use scanner to test objects,*" while keeping the same goal. After this planning tool interaction, Adam kept scanning other objects for viruses or bacteria. Once leaving the laboratory, Adam was prompted again to open the planning tool because they scanned a contaminated object. This time, Adam included "*read about how diseases spread*" in their plan and removed all other planned actions. Instead of following this plan, Adam then went to Kim and submitted an incorrect hypothesis for the cause of the illness and a treatment prevention plan. Soon after, at a 30-minute milestone of gameplay, Adam was prompted to plan, where they kept the same goal with the planned action of "*use scanner to test objects*." Adam continued to test objects and interact with the game environment for another hour, but did not interact with the planning tool during that time. They successfully solved the mystery after 96 minutes of gameplay.

Next, we consider another student named Barbara. Barbara also interacted with the planning tool five times throughout gameplay and did not successfully solve the mystery. They first planned after the tutorial, where they created one plan with the goal of "*explore camp*" and three planned actions: "*go to infirmary,*" "*speak with camp nurse,*" and "*speak with patients*."

Immediately after creating this plan, Barbara went to the infirmary and completed all actions in the plan. During the next planning tool interaction, Barbara kept this plan in the tool despite having completed it. They also added another plan for the goal "*learn about outbreak*" with nine planned actions attached to this goal. In every subsequent planning tool interaction, these plans persisted. Barabara played the game for 71 minutes and never solved the mystery.

For our last scenario, we consider a student named Carl. Like the other students, Carl planned five times during gameplay. Carl's first planning tool instance occurred after completing the tutorial and they created a plan with the goal of "*explore camp*" and five planned "*go to*" actions. After planning, Carl went to the infirmary and then the dormitory completing two out of the five actions. They were then prompted again, where they kept their "*explore camp*" plan the same and added another plan with the goal "*learn about outbreak*" and two planned actions: "*use scanner to test objects*" and "*speak with lab technician*." Shortly after, Carl opened the planning tool again voluntarily and removed the "*explore camp*" plan. After 30 minutes of gameplay, Carl was prompted to plan again. In this interaction, they added two plans with the goals "*test food/objects patients have consumed*" and "*test food/objects that can spread disease*." The first plan contained two "*pick up*" actions and the second plan contained one "*pick up*" action. Carl then proceeded to pick up several items from the dorm. Carl then opened the planning tool voluntarily and removed the "*test food/objects patients have consumed*" plan. Carl did not plan for the remainder of gameplay. They successfully solved the mystery and played in total for 73 minutes.

These scenarios were derived from gameplay alone, using the low-level actions shown in the trace data. These scenarios represent just a few examples of how students utilized planning while navigating CRYSTAL ISLAND, demonstrating how planning "looks" different across individuals. In future sections, we will describe features we have extracted from these sequences to help identify students' SRL processes throughout gameplay. Because these processes are largely metacognitive, we cannot know explicitly which SRL processes students employ. However, this work represents a first step at trying to analyze and support students through these processes using only trace data.

3.3 Plan-Enhanced Game-Based Learning Environment

This section includes a high-level description of a proposed adaptive game-based learning environment that supports students' goal setting and planning throughout science problem-solving.

Two scenarios are presented for how to enhance game-based learning environments with AI-driven adaptive scaffolding: (1) a student plan recognition framework and (2) a student plan generation framework. In this context, we define plan recognition to be the task of predicting a high-level goal and the set of planned actions to achieve that goal given a sequence of gameplay events. Figure 3.1 illustrates how a plan recognition framework can be incorporated into a game-based learning environment to provide adaptive support for students' strategies.

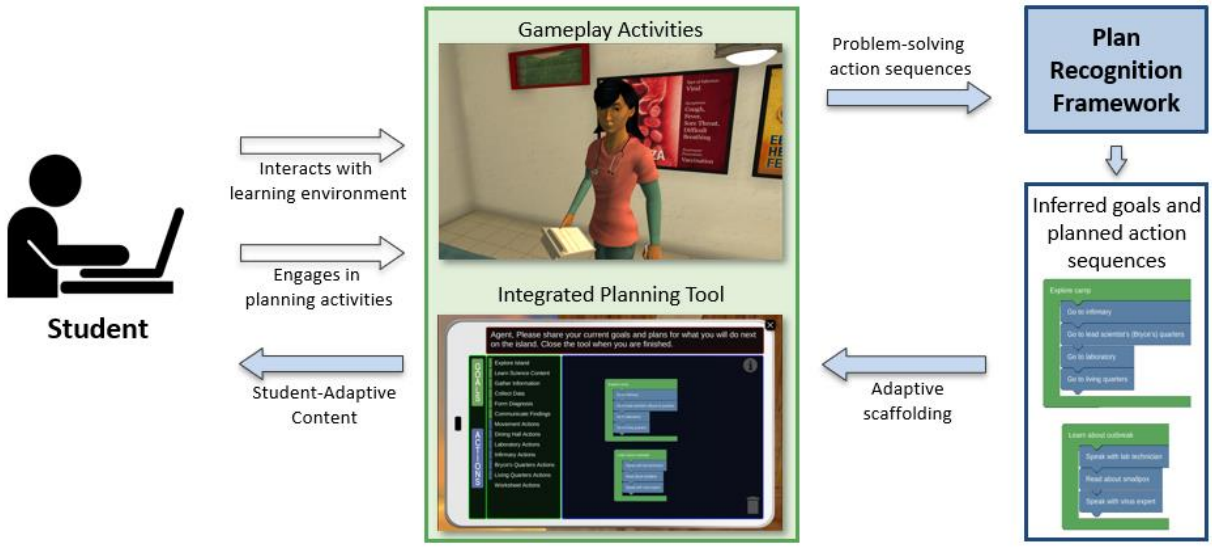

Figure 3.1 Overview of a game-based learning environment that incorporates a plan recognition framework.

As students interact with the game-based learning environment, they engage in SRL processes, and at several points throughout the game, they are prompted to externalize their plans. These gameplay interactions, as well as their planning activities, are automatically logged by the system. All of the data that is logged in the system and passed to the student plan recognition framework, where it is used to infer the current goals and planned action sequences of the student. Based on the outputs of these models, the system can deliver adaptive interventions aimed at improving students' strategies or science problem-solving. For example, if a student has not explicitly planned for a period of time, students could be prompted to verify if the system-recognized plan aligns with their current goal, prompting them to reflect on their current strategy

and adapt their plan. Alternatively, the learning environment could provide hints for how to maneuver the game based on the set of predicted planned actions.

Leveraging the generative features of LLMs creates the potential for the student plan generation framework. 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 prior gameplay events and prior completed goals. Figure 3.2 illustrates how a plan recognition framework can be incorporated into a game-based learning environment to provide adaptive support for students' planning activities.

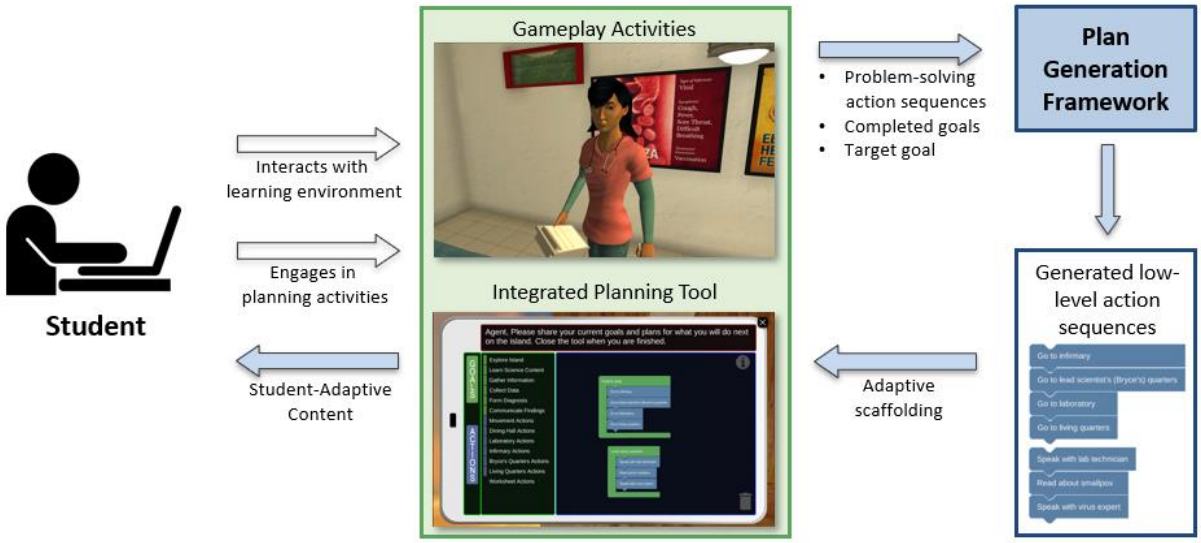

Figure 3.2 Overview of a game-based learning environment that incorporates a plan generation framework.

As students interact with the game-based learning environment, they once again engage in SRL processes, and at several points throughout the game, they are prompted to externalize their plans. 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 series of low-level actions that can be enacted in gameplay. Based on the outputs of these models, the system can deliver adaptive interventions aimed at improving students' planning activities. For example, students can be presented with more efficient plans and given suggestions on actions to include in

their plan through scaffolding. Alternatively, the learning environment could adapt game difficulty based on generated plans for a given goal.

Both frameworks proposed in this work share several fundamental components. A core component of both frameworks is leveraging students' planning activities to evaluate the predictions. A novel aspect of this work is the incorporation of a planning support tool in two science learning environments, CRYSTAL ISLAND and FUTURE WORLDS, which allows students to create their own plans. These explicit plans allow for frameworks that mimic real-time systems by segmenting gameplay through planning tool interactions and can be utilized to evaluate the models without the need for expert annotation. In addition to evaluating the models with student created plans, we have developed plan quality metrics designed to assess student planning during scientific inquiry. Further details on these evaluation metrics can be found in Section 8.1. These metrics (outlined in Chapter 8) can be used to evaluate both students' plans and model-generated plans.

Finally, because the planning support tool was implemented in two science game-based learning environments, we examine the performance of both frameworks within both learning environments. These experiments will aim to demonstrate the generalizability of the frameworks across science learning environments. Details on the implementation of the frameworks will be discussed in Chapters 6 and 7. Through plan recognition and plan generation, we can support both *what* students are planning and *how* they plan to achieve their goals. Both frameworks proposed in this work have the potential to enable adaptive scaffolding in game-based learning environments, by improving student SRL processes and learning.

3.4 Student Planning Scenarios with an Adaptive Learning Environment

As shown in the student planning scenarios, each student accessed the planning tool five times, which could be considered similar planning behavior. However, each student had very distinct behaviors apart from how often they opened the planning tool, illustrating variance in the type of support needed for each student to enhance their learning experience. Below are a few examples of how a learning environment could adapt to such behaviors using the plan recognition and generation frameworks.

Adam had the same goal throughout gameplay but completed all planned actions. This type of behavior is indicative of a student not externalizing their plans fully in the planning support tool. A student plan recognition framework in this context could automatically identify other goals Adam is trying to achieve using their gameplay events and provide additional prompts to externalize those goals. The system could then provide additional scaffolding for formulating the plans through the plan generation framework, prompting Adam to achieve multiple goals.

Barbara had the same plan persist throughout gameplay. In an adaptive learning environment, the student plan recognition framework could automatically identify if this plan was completed during gameplay. If it was already completed, the environment could prompt Barbara to plan by asking them to verify if the recognized goal or plan is accurate. If the plan was not already completed, additional hints or alternative plans could be provided to Barbara through the student plan generation framework, potentially avoiding game abandonment.

Carl voluntarily accessed the planning tool to view and update plans, potentially demonstrating a high SRL skill-level. Plan quality metrics incorporated into the plan generation framework could automatically assess Carl's plans across several science inquiry metrics and provide more detailed hints of alternative actions to achieve their goals. Alternatively, an adaptive learning environment could adjust game difficulty by incorporating new educational concepts based on how quickly Carl achieves their goals or navigates the game's milestones.

CHAPTER 4

CRYSTAL ISLAND DATA CORPORA

This chapter presents the game-based learning environment, the planning support tool, and the study that was conducted using the environment. We analyze trace logs capturing students' gameplay behaviors as well as their interactions with the planning support tool to gain insights into students' planning behaviors.

4.1 Planning Support Tool in CRYSTAL ISLAND

The setting for this work is CRYSTAL ISLAND, a game-based learning environment that teaches microbiology concepts to middle school students (Rowe et al., 2011). CRYSTAL ISLAND is a first-person game where students take on the role of an infectious disease investigator who must solve the mystery of a disease outbreak on a remote island research station (Figure 4.1). Students can walk between different locations, speak to several non-playable characters (NPCs), read books and view posters containing information about microbiology, and conduct tests on potentially contaminated food items using a virtual scanner. To complete the learning scenario, students must correctly identify the disease that is spreading, the object through which the disease is spreading (one of the food items) and prescribe a treatment or prevention plan to stop the spread of the disease.

Figure 4.1 (Left) CRYSTAL ISLAND game-based learning environment, (right) Conversation with an NPC in CRYSTAL ISLAND.

To succeed in CRYSTAL ISLAND, students must engage in effective self-regulation. Students typically have two or three class periods (around 100 minutes of gameplay) to complete the learning scenario, and the size of the problem space makes it so that trial and error approaches will often not be sufficient for solving the mystery. Rather, students must set relevant goals and make deliberate plans in order to succeed in this environment. In this work, we present a novel planning support tool that leverages design concepts from visual programming languages (Weintrop & Wilensky, 2019) and AI planning (Ghallab et al., 2016) to support students' goal setting and planning in CRYSTAL ISLAND.

The planning support tool functions much like a block-based programming environment, where students have access to palettes of predefined blocks that they can snap together using a drag-and-drop interface to create hierarchical (i.e., two-layer) plan structures (Figure 4.2). This interface allows students to configure their own plans from a set of high-level goals and low-level actions rather than having to use natural language to generate goals and plans entirely from scratch, which would be significantly more challenging. Goal blocks represent subgoals that students think will help them achieve the overarching goal of solving the mystery, such as "*learn about outbreak*", "*identify transmission source*", or "*report evidence-based diagnosis*." To achieve those goals, students can construct sequences of action blocks, each representing concrete steps that can be enacted in the CRYSTAL ISLAND game environment. For example, students may plan to "*read about how diseases spread*", "*speak with patients*", or "*use scanner to test objects*." An example plan structure is shown in Figure 4.2, where a student has set the goal of "explore island" and has indicated that the action "go to infirmary" is a step that they intend to take to complete that goal.

We note that, while all goal and action blocks that are provided in the planning support tool are related to the learning scenario presented in CRYSTAL ISLAND, deliberate thought is still required to configure meaningful plans. That is, some combinations of goals and actions may not make much sense, and not all goals and actions are particularly relevant for solving the overarching goal of the learning scenario. Research demonstrates that students struggle to engage in effective self-regulatory goal-setting strategies during learning; however, when provided support from pedagogical agents, they exhibit greater learning outcomes than students without the support (Azevedo et al., 2019; Harley et al., 2018). Therefore, the development of the planning tool provides students with opportunities to self-regulate their learning by allowing them to select

which of the specific goal blocks they want to set as their subgoals for completing the game while also providing support for planning through the overarching goals included in the planning tool.

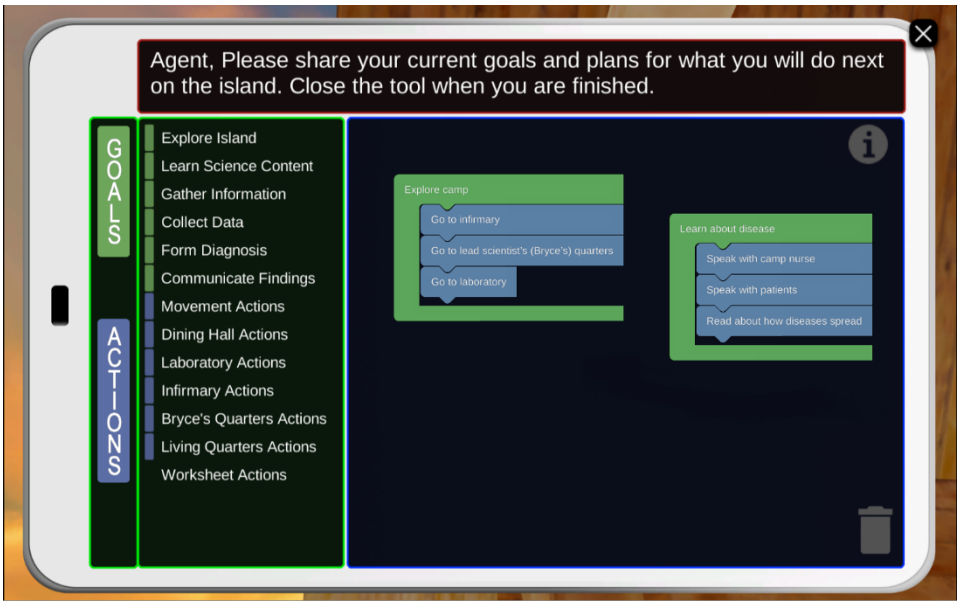

Figure 4.2. Example planning support tool interaction in CRYSTAL ISLAND.

Before students begin playing CRYSTAL ISLAND, they watch a short introduction video explaining the planning support tool and demonstrating how to construct plans using the block-based interface. During gameplay, students are presented with prompts to access the planning support tool whenever they complete milestones in the learning scenario, such as speaking with half of the characters or getting a positive test result using the virtual scanner. To address instances where students are not completing many milestones, prompts are also sent every thirty minutes. However, the game does not force students to stop what they are doing and open the planning support tool, so students may choose to ignore the prompt for planning and thus not generate plans. Students are also able to access the planning support tool voluntarily whenever they like. When a student's use of the planning support tool is in response to a prompt from the system, they are required to modify their plans in some way, such as by adding new goal or action blocks or deleting blocks from their previous plans. When a goal block is deleted, a prompt is triggered that asks the student to indicate whether they have completed that goal. Voluntary use of the planning support tool does not require students to modify their plans, but plans must always be left in a valid configuration - there must be at least one goal block and each goal block must have at least one

action block inside of it. Upon closing the planning support tool, students resume solving the mystery and may choose to enact the plans they set or do something entirely different.

4.2 CRYSTAL ISLAND Study Data

This work uses data collected from 144 eighth grade students (mean age = 13.2; 60% female, 40% male) during asynchronous remote learning in the 2020 school year. Due to the asynchronous nature of the study, students did not have a time limit for completing the game and gameplay time across all students averaged 94.7 minutes (SD = 47.7).

The trace data consists of fine-grained records of students' interactions with every major feature in the learning environment. For example, the details of every conversation with a virtual character, view of a virtual book or poster, interaction with the laboratory testing equipment, interaction with a reading concept matrix, interaction with the diagnosis worksheet, and interaction with the backpack are recorded by the system. For each of these features, the system records the timestamps of when the interactions occurred, when they concluded, and what menu options were selected. Students' problem-solving progress is also recorded by logging the narrative plot points that they complete, their progress in the tutorial, and their performance in the challenge modes.

The benefit of this logging is that it can be analyzed sequentially and in aggregate, both of which are included in this analysis. Some example features generated from the trace data are how many books students read in total, which books they read, how much time they spent viewing each book (on a per book basis), how often they moved back-and-forth between the book and concept matrix, how often they opened and closed the planning tool, and how many plans they created.

CHAPTER 5

FUTURE WORLDS DATA CORPORA

This chapter presents the game-based learning environment, the planning support tool, and the study that will be utilized in proposed work. We will analyze trace logs capturing students' gameplay behaviors as well as their interactions with the planning support tool to compare students' planning behaviors across science game-based learning environments.

5.1 Planning Support Tool in FUTURE WORLDS

FUTURE WORLDS is a game-based learning environment designed to teach sixth graders environmental sustainability concepts (Figure 5.1). Each level of the game presents a problem scenario that identifies an environmental issue and a grid of tiles that represent the current environment for students to explore. Students are tasked with solving the issue and improving the environment by switching existing tiles for more environmentally friendly options. Educational materials like diagrams and reading are provided when students view different tiles. Eco-meters in the game are used to demonstrate the environmental impact of a given tile switch. For example, if a student sees an existing tile that has a poorly managed pig farm and switches it with a pig farm with riparian buffers, the water quality eco-meter will show an improvement. To beat a level, students must hit the goal values on the eco-meters and not exceed the step limit allowed for each level. If students take too many steps, they are prompted to restart the level.

The game, initially developed for a museum, was reworked for classrooms, adding a planning phase before each level. Students view a scenario, then use a block-based tool (Figure 5.2) to outline their strategies by arranging predefined goal (green) and action (blue) blocks into sequences, representing subgoals and steps like identifying issues and adjusting water quality. Actions are categorized into learning, viewing, or changing the environment, with specific actions linked to game elements. Students must formulate at least one complete plan to proceed to gameplay.

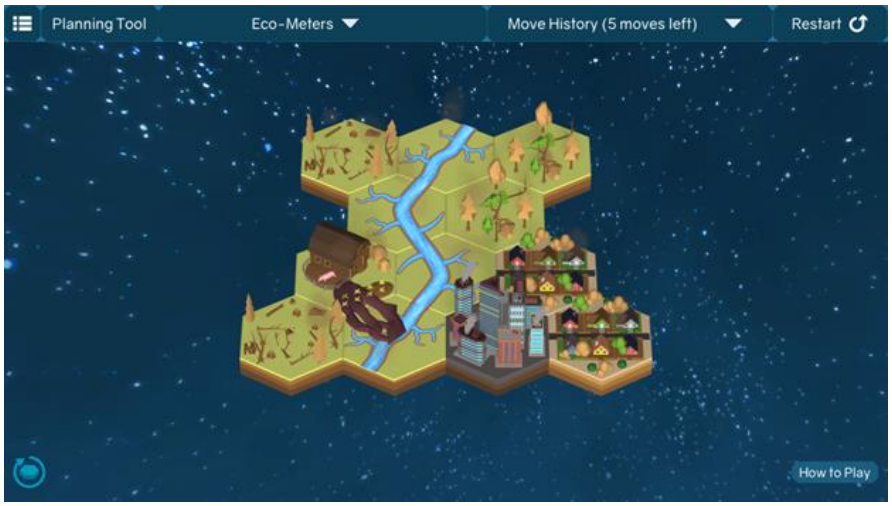

Figure 5.1. An example level of the FUTURE WORLDS learning environment.

5.2 FUTURE WORLDS Study Procedure

A remote pilot study was conducted with 32 students in Texas, USA. Students ranged from 6th to 8th grade, with 43% female and the average age being 12.4 years. Students played FUTURE WORLDS remotely during three 40-minute class periods. Prior to starting the game, students were asked to complete a 10-question pre-test assessing their environmental sustainability knowledge and the achievement goal questionnaire (Elliot et al., 2012). Students were given about 80 minutes to play the game and were offered additional materials if they completed the game early. After completing all levels, students completed the same 10 question knowledge assessment, to analyze learning gain from the game. Students also completed the situational interest scale (Knogler et al., 2015) and the technology acceptance model (Venkatesh & Davis, 2000), to gauge how the game impacted students' interest in science and how they felt about the planning support tool, respectively. Game trace logs of students' actions in each level, as well as their planning tool use, were logged automatically and used for analysis in this work. At the conclusion of the study, we ran focus groups with a subset of the students. The questions were focused on students' understanding of planning and goal setting in general, their thoughts about the game, and their feedback on the planning tool itself.

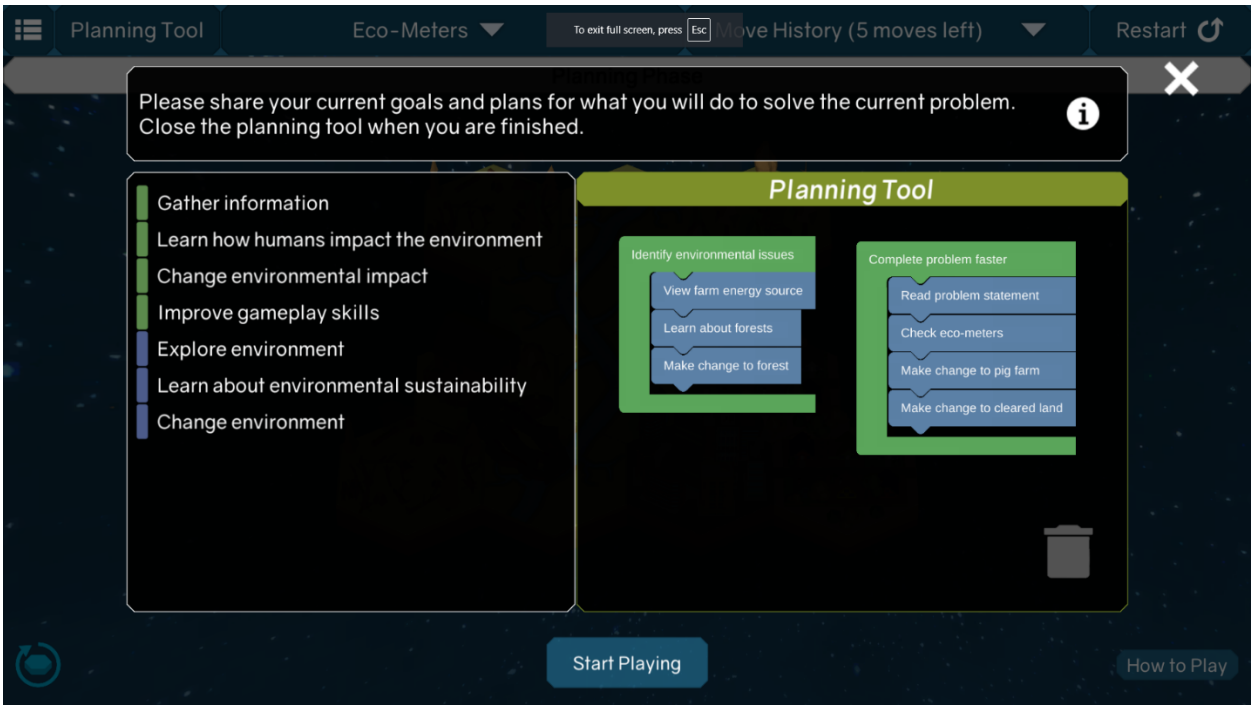

Figure 5.2. Example plans in the planning support tool.

CHAPTER 6

STUDENT PLAN RECOGNITION

This chapter describes the work completed to enable adaptive game-based learning environments through plan recognition in CRYSTAL ISLAND and FUTURE WORLDS. First, we discuss the development of a student plan recognition framework (Goslen et al., 2022a). We report initial performance of the framework in CRYSTAL ISLAND, evaluated on six supervised machine learning models for the task of goal and action sequence recognition. Next, we present an improvement on this framework through the inclusion of automatic plan completion detection and a multi-task architecture (Goslen et al., 2022b). Findings demonstrate both methods improve framework performance for both goal and action sequence recognition. We then present the application of these methods for the task of plan recognition in FUTURE WORLDS. Results demonstrate a similar performance to the CRYSTAL ISLAND baseline student plan recognition framework and demonstrate empirical evidence of the generalizability of these machine learning techniques for the task of student plan recognition in science game-based learning environments.

6.1. Developing a Baseline Student Plan Recognition Framework

Our student plan recognition framework uses sequences of students' gameplay interactions as input to predict: (1) the student's goal and (2) their plan for completing that goal. We evaluate our framework's predictions against plans students created while using the planning support tool.

6.1.1 Event Subsequence Partition Method

CRYSTAL ISLAND's data logging system automatically generates event sequences, or the sequence of in-game events representing the actions each student took while playing the game. For our framework, we extracted three components from each in-game event: event type, event argument, and location. There were 9 event types, including reading a book or article, speaking with a NPC, or scanning an item for a virus or bacteria. The event argument component provides additional detail about the event type, like the title of the book or article, the name of the character the player spoke to, or the reasoning selected for the scanned item. There were 108 unique event arguments.

Lastly, the location component indicates where on the island the event took place. There were 24 fine-grained locations in this version of CRYSTAL ISLAND, such as the infirmary and the cafeteria.

Once the three components from each in-game event were extracted, we constructed the sequence of events for each student. Each event sequence captures key actions the student performed in the game. To allow for recognition of students' plans at multiple points during gameplay, we created several subsequences of students' in-game events by splitting the complete event sequence for each student at every interaction they had with the planning support tool. Based on our theory-based operationalization of planning in CRYSTAL ISLAND (see Section 3), our framework assumes that all in-game events after a planning tool interaction until the next planning tool interaction are aimed at enacting the externalized goals and plans from the planning tool interaction. Therefore, for our analysis, an event sequence starts with the in-game event occurring immediately after a student's planning tool interaction and ends with the in-game event directly before the next planning tool interaction, providing 385 event sequences across students.

Through this partitioning process, we identified that there was a variance in lengths of event sequences (min=1 event, max=454 events, med=30 events). This indicates that there was a wide variety in how often students used the planning support tool. Since the machine learning models that we used to perform goal and plan recognition require fixed-length sequences as input, we preprocessed these event sequences to make them all the same length. Prior work from a similar domain identified highest LSTM performance with a maximum event sequence length of 50 when compared to longer event sequences (Min et al., 2017). Thus, to reduce incorporating game events unrelated to plan enactment in the input data, we used the median length of 30 as the number of events included in each sequence. Any sequences with less than 30 events had blank event entries added to the end to pad them to the correct length, and sequences with more than 30 events were cut short at 30. Next, we constructed a cumulative representation of the event sequences to allow for event-level prediction, providing 11,550 total sequences used as input to our framework. Lastly, as done in prior work, we converted the event sequences to a sparse 2d (30 x 141) binary matrix using one-hot encoding vectors, where each row represents an event in the event sequence and each column represents a potential feature (Min et al., 2017).

6.1.2 Prediction Tasks

To recognize students' plans from their gameplay, we constructed two separate prediction tasks: (1) the selected goal(s) from the planning support tool and (2) the sequence of planned actions students indicated would help them achieve those goals (Figure 6.1). This formalization allows for predicting multiple goals and planned action sequences at once, making each prediction task a multi-label classification task.

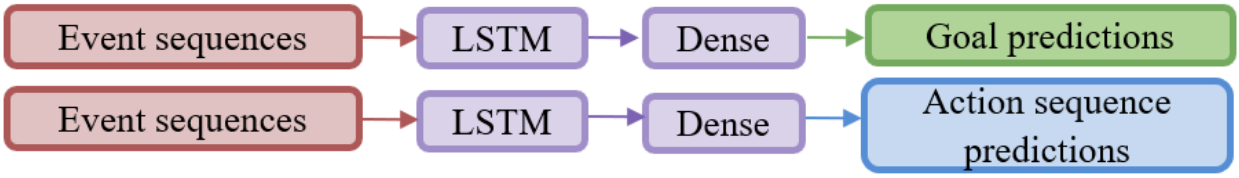

Figure 6.1 Example single-task model architecture.

Goal recognition. Students were able to select from 20 possible goals in the planning support tool. The design of the planning support tool incorporated a mapping of each of these goals to five high-level goal categories: gather information, learn science content, collect data, form diagnosis, and communicate findings. For the goal recognition task, we used these categories as labels, mapping each goal in the plan to the category. To allow for the set of students who selected more than one goal at a time, we constructed a binary label vector of length five, where each column corresponds to one of the five goal categories. For example, in Figure 4.2, there are two goals: "explore camp" and "learn about disease", which map to goal categories "gather information" and "learn science content." In our representation, this would translate to [0 1 0 0 1], where the first 1 entry indicates goal category "gather information" and the second represents goal category "learn science content."

Planned Action Sequence Recognition. To create plans for their goals, students could select from 55 possible actions. These actions were mapped to six planned action categories based on the design of the planning support tool: "explore", "speak with characters", "read science content", "examine poster", "gather and scan items", and "evaluate hypothesis." We wanted to construct a similar label representation as done in the goal recognition task; however, most plans contained more than one planned action and most planning tool interactions contained more than one plan. To create a succinct planned action sequence representation, we utilized NLP techniques. First, we

mapped all planned actions to their corresponding action category. Then, for each plan, we concatenated all action categories together. Next, for each set of action categories, we applied SpaCy word embeddings and averaged the embeddings across all words (Levy & Goldberg, 2014; Srinivasa-Desikan, 2018). Lastly, we applied k-means clustering, with the Elbow method indicating the appropriate number of clusters to be 4. The four resulting clusters aligned with the predominant action category in each plan. The first cluster primarily contained plans with "read science content" actions, the second cluster contained mostly plans with "explore" actions, "gather and scan items" actions were found in the third cluster, and the fourth cluster contained predominantly "speak with characters" actions. In future sections, we utilize these action categories as our naming convention for each cluster. Once these clusters were determined, we constructed a binary vector of length four in a similar manner to the goal recognition task.

Model Selection and Evaluation. To assess the ability to recognize students' goals and planned action sequences, we constructed six supervised machine learning models: support vector machines (SVM), random forests (RF), naive Bayes (NB), logistic regression (LR), multi-layer perceptron (MLP), and long short-term memory (LSTM) networks. We tuned hyperparameters for each of the models through a nested 5-fold cross validation. For each fold, we used a stratified student-level split to ensure similar distributions of classes between train and test splits. The LSTM allowed for a sequential input, meaning we were able to use the 2d one-hot encoding vector as input. SVM, RF, NB, LR and MLP required a non-sequential representation of the data, so we summed the one-hot encoding vectors across in-game events to create a single vector that can be interpreted as the number of times each event type, argument, and location occurs in a sequence.

To evaluate these models, we used macro-average F-measure, which calculates the F-measure for each class, sums them, and then divides by the number of classes. F-measure bases its calculations on false positives and false negatives, which are typically predictions that can cause student frustration in adaptive learning environments. Therefore, it is able to highlight incorrectly classified labels and is an indicator of optimal performance for multi-label classification tasks (Liu & Chen, 2015; Madjarov et al., 2012). Additionally, macro-average is adept at handling label imbalances, which are present in our dataset (Pereira et al., 2018). For these reasons, we used this metric to evaluate all our models.

6.1.3 Results

In this section, we present the results of the student plan recognition framework evaluated with six supervised machine learning models. Results from the six models were compared against a naive majority baseline model that only predicted the majority class.

Single-task Goal Recognition Results. Table 6.1 shows the goal recognition results for all six machine learning models, along with majority baseline results for comparison. Overall, five out of the six models improved on the macro-averaged majority baseline. Additionally, the same models improved on the majority baseline for four goal categories. Interestingly, RF seemed to overfit to the majority goal category, "*gather information*." Some models tend to ignore less-represented classes due to data limitations, causing a model with an imbalanced dataset to overfit to more-represented classes. This may be the case for RF since its decisions are based on information gain. The LSTM performed best for the two least represented goal categories (i.e., "*form diagnosis*" and "*communicate findings*") and the second highest-performing model for two other goal categories. Additionally, considering macro-average, the LSTM was the most improved model on the majority baseline.

Table 6.1 F-measure results for each predictive model and goal category in the goal recognition task. Results are averaged across the nested 5-fold cross validation, with distribution representing the test set. Bolded values indicate highest performance.

	Collect data	Comm. findings	Form diagnosis	Learn science content	Gather info.	Overall
	21%	3%	3%	24%	49%	Macro-avg
Maj.	0.00	0.00	0.00	0.00	0.74	0.15
SVM	0.20	0.07	0.22	0.20	0.71	0.28
RF	0.00	0.00	0.00	0.00	**0.74**	0.15
NB	**0.42**	0.12	0.23	**0.43**	0.58	0.35
LR	0.24	0.16	0.40	0.27	0.67	0.35
MLP	0.29	0.19	0.31	0.16	0.64	0.32
LSTM	0.32	**0.35**	**0.47**	0.35	0.62	**0.42**

Table 6.2 F-measure results for each predictive model and planned action sequence cluster in the planned action sequence recognition task. Results are averaged across the nested 5-fold cross validation, with distribution representing the test set. Bolded values indicate highest performance.

	Read science content	Explore	Gather and scan items	Speak with characters	Overall
	8%	27%	28%	36%	Macro-avg
Maj.	0.00	0.00	0.00	0.55	0.14
SVM	0.36	0.35	0.20	0.29	0.30
RF	0.31	0.41	0.00	0.18	0.22
NB	**0.53**	**0.54**	0.17	0.48	**0.43**
LR	0.46	0.50	0.21	0.43	0.40
MLP	0.29	0.19	**0.31**	**0.64**	0.32
LSTM	0.48	0.47	**0.31**	0.38	0.40

Single-task Action Sequence Recognition Results. Planned action sequence results are shown in Table 6.2. For this task, all models improved on the macro-averaged majority baseline, with NB performing best. Additionally, all models performed better than the majority baseline for three of the planned action sequence clusters. NB outperformed the other models for two of the planned action sequence classes. We noticed that this model often predicted all four classes at a time, which might have caused the results to be inflated. MLP improved on the majority baseline for the "*speak with characters*" cluster.

6.1.4 Discussion and Limitations

Overall, the machine learning-based models show clear improvement with respect to macro-averaged F-measure over a naive baseline on the student goal and plan recognition tasks. Prior work on student goal recognition found LSTMs to be the best performing model on a multiclass goal recognition task (Min et al., 2017). Our work extends these findings by showing that LSTMs also perform effectively for goal recognition in a multi-label context, demonstrating potential for concurrent goal recognition. Allowing for prediction of more than one goal at a time mimics

students' planning tool interactions and makes the student plan recognition framework more robust. Student plan recognition proved to be a more difficult task than student goal recognition. Unlike goal recognition, there was not a single model that performed best across all plan classes. For example, naive Bayes showed the highest macro-average F-measure, but its predictions were consistently every plan class for a given set of input actions. This type of prediction is not ideal to inform run-time scaffolding because it does not provide a precise indication of what students are planning.

The imbalanced labels in the dataset presented challenges in training and evaluating the models for student goal recognition and plan recognition. However, it is representative of the types of plans generated by students through their use of the planning support tool in CRYSTAL ISLAND. Notably, we saw planning support tool usage decrease over time, with students trending toward using the tool frequently in the first half of the game, but less so as time went on. There were also different levels of granularity associated with the different goal categories and plan classes. For example, goals related to gathering information typically occurred early in the game, and they encompassed a relatively broad set of possible actions. In comparison, goals in the "*communicate findings*" category ideally occurred after a student formed a hypothesized diagnosis, which typically occurs later in the game. The steps involved to communicate findings are directly outlined in the game, and as a result, one would expect plans related to this goal to occur less frequently. Encouragingly, the results show the promise of using machine learning-based multi-label classification techniques for student goal and plan recognition despite the inherent challenges of imbalanced data.

The wide variety of student plans also presented distinctive challenges for the student plan recognition framework. Some students frequently used the planning support tool and updated plans without being prompted, while other students opened and closed the planning support tool only when required. This limits our framework because if students do not update their plans, our framework interprets all input actions as being towards the same goal and plan. Similarly, if students use the planning support tool sparingly, then the goal and action labels might not be fully representative of the event sequences enacted in between planning support tool uses. Further enhancements to the framework could be added by identifying when a plan has been completed through gameplay or a goal, so it is not singularly relying on students to update their goals and

plans. Such improvements could alter the distribution in goal and plan labels and potentially help with recognition performance.

6.2 Improving the Student Plan Recognition Framework

After analyzing the results from six machine learning models, we identified two potential improvements to the data-driven student plan recognition framework: (1) the automatic detection of plan completion and (2) a multi-task architecture for the prediction tasks.

6.2.1 Methods

Automatic Plan Completion Detection. A strength of the student plan recognition framework is that it allows for predicting concurrent plans. However, this leads to a key assumption that students mark their plans as being completed or remove them from the planning support tool when their goals change. Our framework uses students' constructed plans as ground truth, and any plan left in the planning support tool is used as a label. This creates issues when training models, as students' event sequences might not align with the stale goals and planned actions. Thus, we incorporated a preprocessing step to remove a goal and planned action sequence from the label set if it has already been completed. We identified when students completed plans by checking if they had enacted all in-game events included in their plan for a given goal. Our aim was to reduce noise in the input for our plan recognition models.

Multi-task Architecture. In our original construction of the student plan recognition framework, each prediction task was trained separately (Figure 6.1). Multi-task model architecture allows each task in the model to be informed by the other. In a multi-task architecture, the model is trained on a shared representation of both tasks, increasing the potential for generalization across tasks (Ruder, 2017). This could potentially benefit model performance for both prediction tasks in the SRL context, as goals and plans are inherently related to each. To potentially improve the framework, we constructed a multi-task LSTM (Figure 6.2) and compared results to the original LSTM results.

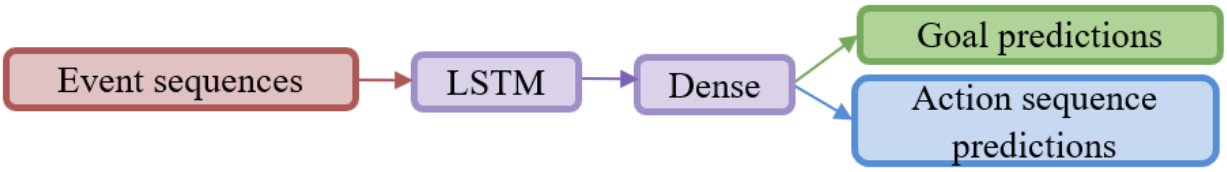

Figure 6.2 Multi-task model architecture.

6.2.2 Results

Multi-task Goal Recognition Results. Results shown in Table 6.3 demonstrate that overall macro-average F-measures were the same between single-task and multi-task goal recognition. Additionally, the performance across individual goal categories remained very similar as well. When incorporating automatic plan completion detection into the multi-task model, we saw a 3% increase in performance, indicating that this preprocessing step helped reduce noise in the dataset. We also saw a performance increase for "collect data" and "communicate findings", further demonstrating that plan completion logic helped boost performance. A total of 77 goals were removed during pre-processing using the plan completion logic.

Table 6.3 Goal recognition F-measure results comparing the single-task LSTM with the multi-task LSTM and the enhanced framework incorporating automatic plan completion detection. Results are averaged across the nested 5-fold cross validation, with distribution representing the test set.

	Collect data	Comm. findings	Form diagnosis	Learn science content	Gather information	Overall
N dist.	21%	3%	3%	24%	49%	
Single-task	0.32	0.35	0.47	0.35	**0.62**	0.42
Multi-task	0.31	0.37	0.47	**0.36**	0.61	0.42
N dist.	22%	5%	6%	22%	46%	
Enhanced Multi-task	**0.36**	**0.48**	0.47	0.34	0.60	**0.45**

Multi-task Action Sequence Recognition Results. Unlike the goal recognition tasks, we saw an increase in overall performance for the multi-task planned action sequence model, implying that a students' selected goal aids in recognizing students planned actions (Table 6.4). Additionally, we

saw an increase in macro-average F-measure when incorporating automatic plan completion detection. A total of 58 sets of planned action sequences were removed from the label set using the plan completion logic. The automatic plan completion multi-task model performed best for "gather and scan items" and "speak with characters" clusters as well. Interestingly, we saw a performance decrease for the "read science content" cluster. This performance drop could be due to the associated goal categories in these plans based on the multi-task architecture. In-game actions related to "read science content" can be used in a variety of ways across gameplay, with reading materials being available in every location in the game. More analysis is needed to understand how and when students utilized these types of planned actions to understand the resulting performance drop.

Table 6.4 Planned action sequence recognition F-measure results comparing the single-task LSTM with the multi-task LSTM and the enhanced framework incorporating automatic plan completion detection. Results are averaged across the nested 5-fold cross validation, with distribution representing the test set.

	Read science content	Explore	Gather and scan items	Speak with characters	Overall
N dist.	8%	27%	28%	36%	
Single-task	**0.48**	**0.47**	0.31	0.38	0.40
Multi-task	0.34	0.44	0.49	0.39	0.42
N dist.	10%	28%	34%	28%	
Enhanced Multi-task	0.31	0.46	**0.53**	**0.40**	**0.43**

6.2.3 Discussion and Limitations

In this work, we found that multi-task models of goal and action sequence prediction boosted overall F-measures relative to single-task models. Furthermore, we found that including a pre-processing step of removing completed goals and action sequences from the label set improves model performance in both tasks. These findings show promise for multi-label multi-task student plan recognition models in game-based learning environments, as does accounting for student goal and action completion in student plan recognition models. While these results indicate improvement in student plan recognition models, there were some limitations with the framework.

Using students' goal setting and planning processes provides a new way to construct plan recognition models, but it also relies heavily on how students utilize the embedded planning support tool in CRYSTAL ISLAND. Event sequences are segmented when students open and close the tool and there was considerable variance in the number of times students opened the planning tool. If a student does not open the planning support tool until the end of the game, the presented framework could train student plan recognition models based upon a set of goals and action sequences spanning an entire gameplay session. More analysis needs to be done exploring the relationships between game play activity and planning activity to better understand when and why students choose to interact with the planning support tool.

Additionally, our framework partially assumes students will update their plans once they have completed a goal or changed their strategy. Anecdotally, we have observed that this is not always the case. In some cases, students may leave their plans in the planning support tool, and their interactions with the planning support tool may decrease over the course of gameplay. Our action deletion process attempts to alleviate part of this problem. Encouragingly, it shows promise in helping to reduce noise in the dataset. Additional exploration into how students altered their plans throughout gameplay is needed to better understand how to address issues of non-updated plans and planning support tool interactions declining over time.

Lastly, there was an imbalance in the dataset's label distribution, which might have affected overall performance of the LSTMs, especially for the goal recognition task. The imbalanced selection of goal categories could point to a greater pattern in student strategies to solve the mystery of the game. Because of the size of the dataset, performance for the least represented classes still had a weak performance, despite the methodological improvements. It would be important to explore the use of language models that can be pre-trained on much larger datasets.

6.3. Applying Student Plan Recognition to FUTURE WORLDS

To examine the generalizability of the student plan recognition framework across science game-based learning environments, we kept the methods as close to the CRYSTAL ISLAND framework as possible. Using students' FUTURE WORLDS gameplay and planning data, we constructed two separate multi-label prediction tasks: (1) the selected goal(s) from the planning support tool and (2) the sequence of planned actions students indicated would help them achieve those goals.

6.3.1 Methods

Event Sequence Partition Method. Like CRYSTAL ISLAND, FUTURE WORLDS' data logging system automatically generates event sequences, or the sequence of in-game events representing the actions each student took while playing the game. To represent each in-game event that a student performs, we created a vector to represent the action target, the tile on which the action is taken, and the action argument. Action target represents one of the nine distinct actions that a student can take in the game, such as "Opened Tile Content" and "Submitted Tile Change". The tile argument represents the tile on which the action was taken and can represent one of 21 possible tiles. If the action is not related to a tile, this feature is empty. The action argument provided additional details about the argument and can have 18 unique values. We extracted sequences of these features by considering action sequences between planning instances.

Once the three components from each in-game event were extracted, we constructed the sequence of events for each student. Based on our theory-based operationalization of planning in CRYSTAL ISLAND (see Section 3), we split event sequences by student planning tool usage. In FUTURE WORLDS, students are required to plan before each level and have the option to plan during the level as well. We maintained the event sequence partition method from CRYSTAL ISLAND, with one additional condition of ending an event sequence when a level ends. Therefore, for our analysis, an event sequence starts with the in-game event occurring immediately after a student's planning tool interaction and ends with the in-game event directly before the next planning tool interaction, providing 200 event sequences across students.

To be consistent with the CRYSTAL ISLAND plan recognition methods, we set the maximum sequence length to 20, as that was the median number of events between planning tool uses in FUTURE WORLDS. Any sequences with less than 20 events had blank event entries added to the end to pad them to the correct length, and sequences with more than 20 events were truncated at 20. We also constructed a cumulative representation of the event sequences to allow for event-level prediction, providing 4,000 total sequences used as input to our framework. Lastly, as done in prior work, we converted the event sequences to a sparse 2d (20 x 49) binary matrix using one-hot encoding vectors, where each row represents an event in the event sequence and each column represents a potential feature (Min et al., 2017).

Goal recognition. Students were able to select from 15 possible goals in the planning support tool. The design of the planning support tool incorporated a mapping of each of these goals to four high-level goal categories: gather information, change environmental impact, learn how humans impact the environment, and improve gameplay skills. For the goal recognition task, we used these categories as labels, mapping each goal in the plan to the category. Similarly to the CRYSTAL ISLAND representation, we constructed a binary label vector of length four, where each column corresponds to one of the four goal categories.

Planned Action Sequence Recognition. In FUTURE WORLDS, students could select from 23 possible actions. These actions were mapped to three planned action categories based on the design of the planning support tool: "*explore environment*", "*change environment*", and "*learn about environmental sustainability*." To simplify the label representation process for planned action sequence recognition, we opted not to utilize word embeddings and clustering. Instead, we found the most used action category in a given plan as the label for that plan. We did this to minimize preprocessing time and increase the interpretability of the prediction. Once the majority action category was determined for each plan in a planning tool instance, we constructed a binary vector of length 3 in a similar manner to the goal recognition task to allow for the prediction of multiple plans at a time. For example, consider a plan with three planned actions: "*view farm energy source*," "*view pig farming techniques*," and "*make change to pig farm*." Each planned action would be mapped to an action category: "*explore environment*," "*explore environment*," and "*make change to environment*," respectively. Then, we would assign the label "*explore environment*" to this plan since it is the majority action category. The final step in the label representation process would be to repeat this for all other plans in the planning tool instance and convert the assignment to a binary vector of length three.

Improvements to the Framework. We also examined the effects of automatic plan completion detection and a multi-task architecture on student plan recognition performance in FUTURE WORLDS. We applied the automatic plan completion detection as a preprocessing step, removing goals and planned action sequences from the label set if they were completed in the gameplay events. We also applied the multi-task architecture to the LSTM to examine its impact on predictive performance compared to the FUTURE WORLDS baseline and CRYSTAL ISLAND results.

6.3.2 Results

Single-task Goal Recognition Results. Table 6.5 shows the FUTURE WORLDS goal recognition results for all six machine learning models, along with the majority baseline results for comparison. Overall, four out of the six models improved on the macro-averaged majority baseline. Like the CRYSTAL ISLAND results, RF seemed to overfit to the majority goal category, "*gather information*." Because this data was also imbalanced, this confirms our justification for RF ignoring less-represented classes due to data limitations. The LSTM had the highest overall macro-average F-measure. Naive Bayes performed best for three goal categories (i.e., "*change environmental impact*", "*learn how humans impact the environment*" and "*improve gameplay skills*").

Table 6.5 F-measure results for each predictive model and goal category in the goal recognition task in FUTURE WORLDS. Results are averaged across the nested 3-fold cross validation, with distribution representing the test set. Bolded values indicate highest performance.

	Gather Info.	Change Env. Impact	Learn How Humans Impact the Env.	Improve Gameplay Skills	Overall
	53%	25%	16%	5%	Macro-avg
Maj.	**0.75**	0.00	0.00	0.00	0.19
SVM	0.66	0.04	0.05	0.00	0.19
RF	0.74	0.00	0.00	0.00	0.18
LR	0.68	0.12	0.07	0.01	0.22
NB	0.12	**0.41**	**0.30**	**0.07**	0.23
MLP	0.67	0.08	0.11	0.03	0.22
LSTM	0.71	0.15	0.08	0.04	**0.24**

Single-task Action Sequence Recognition Results. Planned action sequence results for FUTURE WORLDS are shown in Table 6.6. For this task, three of the six models improved on the macro-averaged majority baseline, with NB performing best. All models performed better than the majority baseline for two of the planned action sequence classes. NB outperformed the other models for two of the planned action sequence classes. Notably, LSTM performed the third best out of the six, which differs from the CRYSTAL ISLAND performance. The size of the dataset could be a limiting factor for the models in this case. Additionally, the planned action sequence classes were more balanced than the goal categories, which could indicate why their is overall higher performance than the goal recognition task.

Table 6.6 F-measure results for each predictive model and planned action sequence class in the planned action sequence recognition task for FUTURE WORLDS. Results are averaged across the nested 3-fold cross validation, with distribution representing the test set. Bolded values indicate highest performance.

	Explore Env.	Change Env.	Learn About Env. Sustainability	Overall
	39%	37%	24%	Macro-avg
Maj.	**0.61**	0.00	0.00	0.20
SVM	0.29	0.18	0.13	0.20
RF	0.32	0.15	0.10	0.19
LR	0.39	0.15	0.10	0.19
NB	0.20	**0.56**	**0.40**	**0.39**
MLP	0.36	0.33	0.21	0.30
LSTM	0.33	0.26	0.19	0.26

Multi-task Goal Recognition Results. Results shown in Table 6.7 demonstrate that overall macro-average F-measures dropped between single-task and multi-task goal recognition. The performance across individual goal categories remained very similar, while being a bit lower than the single-task results. When incorporating automatic plan completion detection into the multi-task model, we saw a 1% increase in performance, indicating that this preprocessing step helped reduce noise in the dataset. We also saw a performance increase for "change environmental impact" and "learn how humans impact the environment", further demonstrating that plan completion logic helped boost performance. A total of 36 goals were removed during pre-processing using the plan completion logic.

Table 6.7 Goal recognition F-measure results comparing the single-task LSTM with the multi-task LSTM and the enhanced framework incorporating automatic plan completion detection for FUTURE WORLDS. Results are averaged across the nested 3-fold cross validation, with distribution representing the test set.

	Gather Info.	Change Env. Impact	Learn How Humans Impact the Env.	Improve Gameplay Skills	Overall
N dist.	53%	25%	16%	5%	Macro-avg
Single-task	**0.71**	0.15	0.08	0.04	0.24
Multi-task	0.68	0.11	0.07	0.00	0.22
N dist.	51%	27%	17%	6%	
Enhanced Multi-task	0.67	**0.20**	**0.12**	0.00	**0.25**

Multi-task Action Sequence Recognition Results. Unlike the goal recognition tasks, we saw an increase in overall performance for the multi-task planned action sequence model, implying that a students' selected goal aids in recognizing students planned actions (Table 6.8). There was no increase in macro-average F-measure when incorporating automatic plan completion detection. A total of 42 sets of planned action sequences were removed from the label set using the plan completion logic. The automatic plan completion multi-task model performed best for "change environment" and "learn about environmental sustainability" classes as well. Interestingly, we saw a performance decrease for the "explore environment" class. This performance drop could be due to the associated goal categories in these plans based on the multi-task architecture. In-game actions related to "explore environment" are mainly clicking on tiles, which may happen in conjunction with any of the goals. More analysis is needed to understand how and when students utilized these types of planned actions to understand the resulting performance drop.

Table 6.8 Planned action sequence recognition F-measure results comparing the single-task LSTM with the multi-task LSTM and the enhanced framework incorporating automatic plan completion detection for FUTURE WORLDS. Results are averaged across the nested 3-fold cross validation, with distribution representing the test set.

	Explore Env.	Change Env.	Learn About Env. Sustainability	Overall
N dist.	39%	37%	24%	Macro-avg
Single-task	0.33	0.26	**0.19**	0.26
Multi-task	**0.38**	0.24	0.17	**0.27**
N dist.	38%	37%	24%	
Enhanced Multi-task	0.31	**0.31**	**0.19**	**0.27**

6.3.3 Discussion and Limitations

Overall, the machine learning-based models achieved clear improvements with respect to macro-averaged F-measure over a naive baseline on the student goal recognition tasks. For the planned action sequence recognition, only half of the models improved on the majority baseline, further showing that student plan recognition proved to be a more difficult task than student goal recognition. Much like the CRYSTAL ISLAND framework performance, LSTMs were found to have highest performance for goal recognition and NB had the highest performance for planned action sequence recognition.

Furthermore, there was also a label imbalance in the FUTURE WORLDS data, which presented challenges in training and evaluating the models for student goal recognition and plan recognition. In CRYSTAL ISLAND, we saw planning support tool usage decrease over gameplay. However, in FUTURE WORLDS, we designed the planning tool to be used at fixed points throughout gameplay (i.e. before each level). The mandatory planning before the level could explain why there were more "*explore environment*" goals than any other goal category. However, notably, students did voluntarily access the tool intermittently throughout levels. This similarity in label imbalance is striking because gameplay and science reasoning occur very differently in FUTURE WORLDS. Because FUTURE WORLDS is a puzzle-based game, trial and error is expected, meaning we expect students to be creating and testing hypotheses at a more rapid pace than in CRYSTAL ISLAND. Prior to the data collection, we expected students to adopt strategies across each level to simply submit changes to tiles to examine the effect on the environment rather than gathering information first. This label imbalance may indicate that students rely on planning to help get oriented in the game and explore the environment. The imbalance also further indicates the need for the models to be able to handle such an imbalance, as students may utilize these strategies more in science game-based learning environments.

The size of the data set seemed to be a limiting factor for performance as well. The highest performance we saw in either prediction task for FUTURE WORLDS was 39% macro-average F-measure, which is still quite low when considering the framework for real-time support. While the combination of automatic plan completion detection and multi-task architectures demonstrated the highest performance for both goal and planned action sequence recognition using LSTMs, the performance was even lower than 39%. More data or utilization of pre-trained models is needed to improve the predictive performance. Additionally, transfer learning between science learning environments could be explored to examine the impact on performance. Because of the similar trends in performance, these findings show promise for the generalizability of multi-label multi-task student plan recognition models in science game-based learning environments, as does accounting for student goal and action completion in student plan recognition models.

CHAPTER 7

STUDENT PLAN GENERATION

This chapter introduces 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. The framework is initially constructed and evaluated on CRYSTAL ISLAND planning data. We then investigate the extension of the framework to the FUTURE WORLDS learning environment. Lastly, we evaluate this framework on students' planning data from both CRYSTAL ISLAND and FUTURE WORLDS. We present an initial mixed-methods analysis of the generated plans in this chapter. In the next chapter, we utilize plan evaluation metrics for a deeper analysis of the LLM-generated and student-generated plans.

7.1. Plan Generation Framework Construction

In this work, we harness the capabilities of pre-trained LLMs to analyze textual representation of 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 encode the input because the event sequences consist of information contextually specific to CRYSTAL ISLAND (i.e., lists of game actions and locations). The second phase involves decoding the output of the framework, the generated plans. We use a sequence-to-sequence (seq2seq) 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 framework, 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, 2020) to examine the impact of few-shot learning and prompt engineering with a large language model for plan generation.

7.1.1 Methods

Fully-supervised (T5) Approach. 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., 2016).

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 (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 (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* : G_c,*Goals* : G_t] where E is the series of game events prior to planning, G_c represents a set of goals that a student has completed, and G_t denotes the set of target goals the student constructed during planning. Note, that 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,G_c = {},G_t = {*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, G_c = {*Explore island*}, G_t = {*Find sick individuals*}]. For the T5 model, we set a maximum event sequence length of 30 (Goslen et al., 2022a). In cases where the event sequences were longer than 30 events, we used the 30 game events closest to the plan generation prediction point. For example if there were 50 events in an event sequence, we would use events 20 through 50 as input, rather than events 1 through 30.

For our fully-supervised approach, we opted for a 5-fold student-level cross validation. 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.

Few-Shot Learning (GPT-3.5) Approach. 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. The end of the prompt consisted of the input event sequence (E), the set of completed goals (G_c) and the target goal (G_t). Because the GPT-3.5 API has a token limit per call-response, we limited the number of examples used for a few-shot learning approach to two plans in the prompt.

Table 7.1 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.

Evaluation. To evaluate both language models' ability to generate plans, we developed evaluation methods to qualitatively and quantitatively assess the generated plans. The evaluation methods in this section are meant to be exploratory in nature and provide a range of techniques for plan evaluation. Our quantitative metrics 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). 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, 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

Table 7.1 An example GPT-3.5 prompt used for the CRYSTAL ISLAND learning environment implementation. Note: event sequences and example plans have been reduced for space. Actual prompts included event sequences of length 30 and two example plans.

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:

each of these groups to their 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.

7.1.2 Results

Quantitative Analysis. 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 7.1 illustrates the distribution of 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).

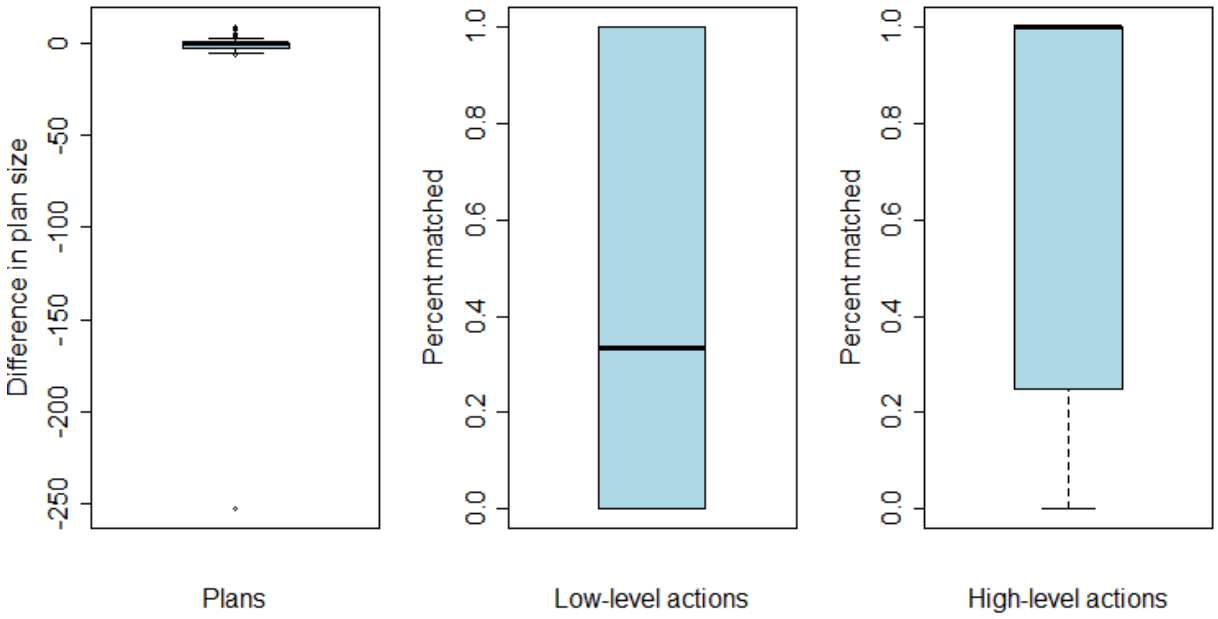

Figure 7.1. Distribution metrics for T5 generated plans for CRYSTAL ISLAND.

The plan size difference had a median of 0.00 (mean = -0.89, SD = 10.81). 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 253, signifying that the T5 model was larger than the corresponding student plan by 253 low-level actions. Upon further investigation, this is due to T5 repeating actions several times. The next chapter will present findings with duplicates removed.

The match percentage for low-level actions (p_L) had a median of 0.33 (mean = 0.39, SD = 0.41), as depicted in the center boxplot. Among the 565 generated plan instances, 25.48% generated all low-level actions present in the corresponding students' plans. We also observed that 41.06% of generated plans had no low-level actions matching those of the student plan.

The match percentage for high-level actions (p_H) had a median of 1.00 (mean = 0.71, SD = 0.42). Notably, there was an increase in exact matches for high-level actions to 63.89% and a decrease in the amount of plans with no high-level action match to 21.59%. Although only a quarter 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.

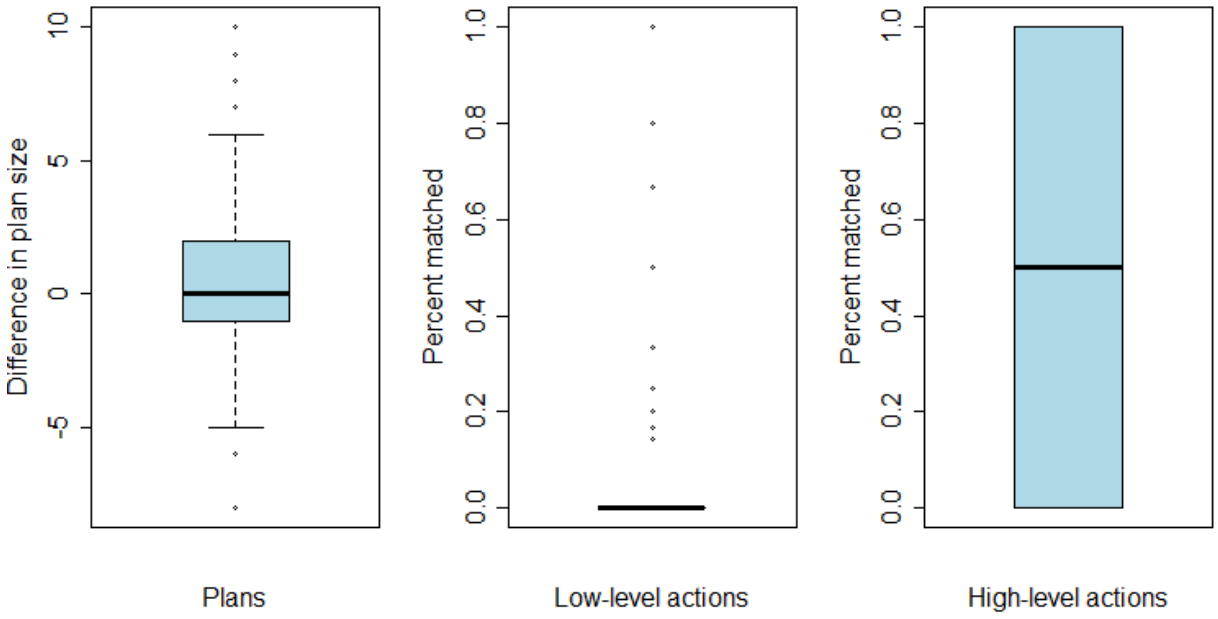

Figure 7.2. Distribution metrics for GPT-3.5 generated plans for CRYSTAL ISLAND.

Figure 7.2 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 0.00 (mean=0.47, SD=2.53). The maximum and minimum plan size differences were 10 and -8, respectively. This difference is much 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 0.00 (mean=0.12, SD=0.28). Out of 565 GPT generated plans, 7.26% had plans where all low-level actions matched that of the students' plans and 78.05% contained no low-level action matches to the corresponding student plan. For GPT-3.5, the median p_H was 0.50 (mean=0.50, SD=0.43). Similarly to T5, we saw an increase in generated plans where all high-level actions were present in the students' plan to 38.05% and a decrease in no matches to 34.51%. These results were lower than the T5 match rates at the high-level, indicating T5 more closely aligned with students' plans.

Plan Composition Analysis. Table 7.2 shows the high-level action distribution for students, T5 and GPT-3.5. Both the T5 and GPT-3.5 "gather information" high-level goals 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 neither LLM is ideal in this case. For goal categories "form diagnosis" and "learn science content", T5 had very similar distributions to students' plans. For the other goal categories, the T5 model seemed to overfit to the most student-used action categories. Interestingly, students used "gather and scan items" only 50% of the time for "collect data" goals, but T5 generated plans with 71% of actions being "gather and scan items". This demonstrates a case where the generated plan might benefit students, as gathering and scanning items is crucial for collecting the data necessary to solve the mystery. 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 "form diagnosis" category has a more even distribution of action categories than students. The students primarily used "read science content" goals in this category, which is one strategy of many that could be used to form a diagnosis. The diversity of action categories used in the GPT-3.5 plans highlights a potential to show students other strategies they might not be considering in the game.

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 LLM-

Table 7.2. High-level action category distribution for each high-level goal in the CRYSTAL ISLAND planning tool across student, T5, and GPT-3.5 plans.

		Collect data	Comm. Findings	Form diagnosis	Learn science content	Gather info.
T5	Explore	3.65%	0.00%	0.00%	16.57%	97.03%
	Speak with characters	11.73%	85.71%	1.87%	56.40%	1.70%
	Read science content	13.43%	11.43%	70.09%	24.13%	0.99%
	Gather and scan items	70.80%	0.00%	3.74%	1.16%	0.00%
	Examine poster	0.00%	2.86%	24.30%	1.74%	0.28%
	Evaluate hypothesis	0.39%	0.00%	0.00%	0.00%	0.00%
GPT-3.5	Explore	2.82%	0.00%	0.40%	0.69%	96.30%
	Speak with characters	2.82%	39.22%	36.65%	36.77%	2.65%
	Read science content	0.00%	11.76%	29.88%	41.24%	0.00%
	Gather and scan items	88.73%	0.00%	7.57%	2.41%	1.06%
	Examine poster	5.63%	41.18%	23.51%	18.90%	0.00%
	Evaluate hypothesis	0.00%	7.84%	1.99%	0.00%	0.00%
Student	Explore	8.21%	9.68%	0.00%	18.67%	62.76%
	Speak with characters	21.64%	45.16%	2.56%	35.13%	17.52%
	Read science content	13.43%	6.45%	73.50%	20.25%	5.44%
	Gather and scan items	49.75%	3.23%	4.27%	9.49%	8.67%
	Examine poster	2.49%	22.58%	17.95%	14.24%	4.08%
	Evaluate hypothesis	4.48%	12.90%	1.71%	2.22%	1.53%

generated 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 7.2. 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.

7.2. Applying Student Plan Generation to FUTURE WORLDS

To examine the generalizability of the plan generation framework, we applied the same CRYSTAL ISLAND methods to FUTURE WORLDS. We align the fully-supervised and few-shot learning techniques as closely as possible to the CRYSTAL ISLAND methods and examine the trends in performance between learning environments.

7.2.1 Methods

T5. We once again pre-trained the T5 model using a pre-processed English language corpus of approximately 700GB in size called the Colossal Clean Crawled Corpus (C4). We selected T5-small, for fine-tuning our FUTURE WORLDS plan generation model.

As input to our model, we created tuples of student in-game events containing three attributes: the event target (9 distinct actions that a student can take in the game), the tile on which the event is taken (21 possible tiles), and the event argument (17 unique values). For example, one such event would be denoted as e = (*Submitted Tile Change, Farm using Nonrenewable Energy, Farm using Solar Energy*). Note this feature extraction aligns with prior work done in CRYSTAL ISLAND and FUTURE WORLDS (Chapter 6).

Once we constructed a 3-tuple for each event, we partitioned each student's event sequences based on the student's utilization of the planning support tool, similar to the CRYSTAL ISLAND implementation. As additional input features, we included completed goals and the target goal of the planning tool instance. If there was more than one goal in a planning tool instance, we treated each plan as a separate row of input. Formally, we denote the input as I : [E = (e_i,e_{i+1},...,e_{i+k}), *Completed* : G_c,*Goals* : G_t] where E is the series of game events prior to planning, G_c represents a set of goals that a student has completed, and G_t denotes the set of target goals the student constructed during planning. Note, that G_c could be empty if a student had not completed any of their goals in the planning tool. To represent the sequence of actions taken by the student, we consider a maximum of 20 actions taken by the student since their last instance of planning. The maximum sequence length was determined by considering the median number of actions between consecutive planning instances across students in our dataset.

For our fully-supervised approach, we opted for a 3-fold student-level cross validation, since the dataset was smaller than the CRYSTAL ISLAND data. We fixed the hyperparameters to

Table 7.3 An example GPT-3.5 prompt used for the FUTURE WORLDS learning environment implementation. Note: event sequences and example plans have been reduced for space. Actual prompts included event sequences of length 20 and two example plans.

For each given action 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 tile-based world where there is an imbalance of environmental factors like food availability, water quality and air quality. The player must properly balance these factors by changing tiles to be more sustainable. Eco-meters are used to demonstrate the effect of changes made by the player and demonstrate the optimal values needed for resources to win the game. Players can take the following actions to solve the mystery:
1) learning about environmental sustainability through viewing diagrams or reading more information
2) exploring the environment by viewing a tile
3) making a change to a tile
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:
OpenedMoreInfo Organic Pig Farm
OpenedTileContent Farm using Nonrenewable Energy
SelectedTileOption Farm using Solar Energy
SubmittedTileChange Farm using Nonrenewable Energy Farm using Solar Energy
OpenedEcometerDropdown
Completed Goals:
Target Goal: identify environmental issues
Actions: make change to river, view city
Generate a set of actions for the following:
Event sequence:
OpenedDiagram Biofuel
SelectedTileOption Farm using Solar Energy
SubmittedTileChange Farm using Nonrenewable Energy Farm using Solar Energy
ConceptMatrices How do diseases spread? Infirmary
OpenedEcometerDropdown
Completed Goals: identify environmental issues
Target Goal: identify environmental issues
Actions:

have 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. These parameters were kept consistent with the CRYSTAL ISLAND implementation.

GPT-3.5. For our FUTURE WORLDS few-shot learning plan generation implementation, we also utilized the gpt-3.5-turbo model and generated plans using the chat completions API. Like the CRYSTAL ISLAND implementation, we included a description of FUTURE WORLDS, 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 in the prompt. We once again selected examples randomly from the set of goals within a goal category, since not all goals had the same usage amount. There were four goal categories in FUTURE WORLDS: gather information, change environmental impact, learn how humans impact the environment and improve gameplay skills. The end of the prompt consisted of the input event sequence (E), the set of completed goals (G_c) and the target goal (G_t). See Table 7.3 for an example prompt.

Evaluation. We applied the same evaluation techniques to our FUTURE WORLDS generated plans: a quantitative analysis of the plan size and match percentages, and a qualitative analysis of the plan composition for each goal category.

7.2.2 Results

Quantitative Analysis. To compare the fully-supervised and few-shot learning approaches in FUTURE WORLDS, we applied the match percentage analysis to the T5 and GPT-3.5 generated plans. Figure 7.3 illustrates the distribution of 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 0.00 (mean = -14.53, SD = 54.50). 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 256, signifying that the T5 model was larger than the corresponding student plan by 256 low-level actions. This aligns with the performance of the T5 model in CRYSTAL ISLAND, where actions were repeated several times. The next chapter will present findings with duplicates removed.

The match percentage for low-level actions (p_L) had a median of 0.00 (mean = 0.31, SD = 0.42), as depicted in the center boxplot. Among the 260 generated plan instances, 23.08% generated all low-level actions present in the corresponding students' plans. We also observed that 60.77% of generated plans had no low-level actions matching those of the student plan.

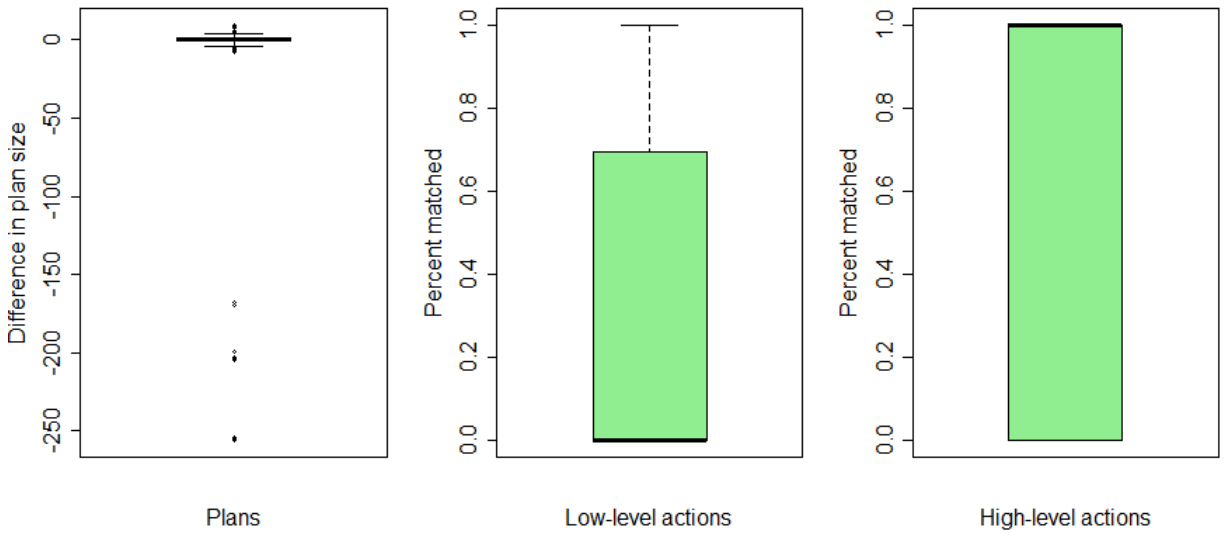

Figure 7.3. Distribution metrics for T5 generated plans for FUTURE WORLDS.

The match percentage for high-level actions (p_H) had a median of 1.00 (mean = 0.68, SD = 0.47). Notably, there was an increase in exact matches for high-level actions to 65.00% and a decrease in the amount of plans with no high-level action match to 31.15%. Although only a quarter 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 they could show students a wide variety of plans that would be similar to their own plan, allowing them to reflect on their plan.

Figure 7.4 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 -1.00 (mean=-0.86, SD=2.31). The maximum and minimum plan size differences were 7 and -11, respectively. Once again, this difference is much 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 0.00 (mean=0.17, SD=0.31). Out of 260 GPT-generated plans, 9.18% had plans where all low-level actions matched that of the students' plans and 68.36% contained no low-level action matches to the corresponding student plan. For GPT-3.5, the median p_H was 1.00 (mean=0.68, SD=0.47). Similarly to T5, we saw an increase in generated plans where all high-level actions were present in the students' plan to 22.45% and a decrease in no matches to 30.10%. These results were lower than the T5 match

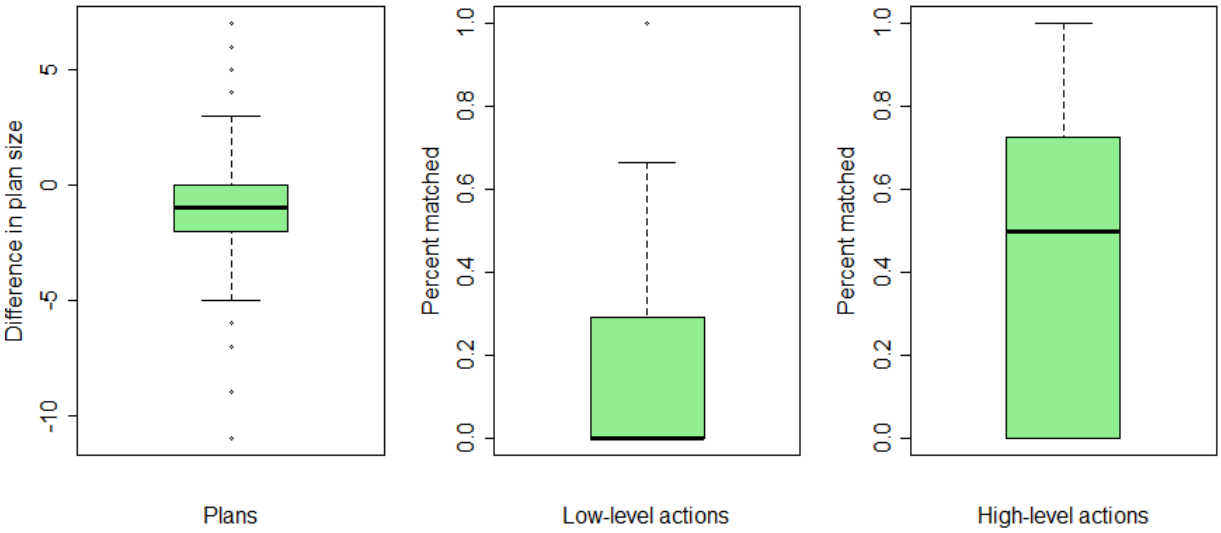

Figure 7.4. Distribution metrics for GPT-3.5 generated plans for FUTURE WORLDS.

rates at the high level, indicating T5 is more closely aligned with students' plans. These findings are consistent with the CRYSTAL ISLAND results, demonstrating that the methods generalize across science game-based learning environments despite differences in game mechanics and dataset sizes.

Plan Composition Analysis. Table 7.4 shows the high-level action distribution for students, T5 and GPT-3.5. Compared to the CRYSTAL ISLAND performance, T5 generated plans seem to differ more from student plans, likely due to the smaller size of the FUTURE WORLDS dataset. Interestingly, T5 only predicted "change environment" actions in two goal categories: "change environmental impact" and "improve gameplay skills." For the "change environmental impact" category, it could be argued that only "change environment" actions could achieve these goals. However, in ideal gameplay, we would want to ensure students understand the effect of the changes to tiles that they make. In an adaptive setting, we may opt to suggest actions in other action categories to ensure students have the underlying knowledge needed to understand the environmental impact. Thus, plans that students created and GPT-3.5 plans may be more beneficial in this case.

"*Improve gameplay skills*" was the least used goal category, with students using only "*learn about environmental sustainability*" actions with these goals. Both T5 and GPT-3.5 generated plans with actions primarily from the "change environment" category for this category.

Table 7.4. High-level action category distribution for each high-level goal in the FUTURE WORLDS planning tool across student, T5, and GPT-3.5 plans.

		Change env. Impact	Learn how humans impact the env.	Improve gameplay skills	Gather info.
T5	Learn about env. sust.	0.00%	1.04%	0.00%	0.26%
	Change env.	100.00%	95.23%	100.00%	52.32%
	Explore env.	0.00%	3.73%	0.00%	47.43%
GPT-3.5	Learn about env. sust.	12.50%	75.86%	7.69%	10.23%
	Change env.	84.38%	20.69%	92.31%	36.96%
	Explore env.	3.13%	3.45%	0.00%	52.81%
Student	Learn about env. sust.	4.40%	22.39%	100.00%	17.36%
	Change env.	84.62%	40.30%	0.00%	21.88%
	Explore env.	10.99%	37.31%	0.00%	60.76%

This demonstrates a case where an adaptive system could assist students through a different strategy than they have previously considered.

Both "*learn how humans impact the environment*" and "*gather information*" categories demonstrated different distributions across students, T5 and GPT-3.5 plans. Because FUTURE WORLDS is a cause-and-effect game, gathering information can be completed through any action in the game. For example, changing a tile can demonstrate the environmental impact of the tile being changed, viewing diagrams can help students understand the curricular content better, and viewing tiles can help orient students in the game. T5 was the only set of plans that did not have "*explore environment*" as the majority action category. In fact, T5 seemed to generate plans primarily containing "*change environment*" actions across all goal categories. It is possible this is due to bias in the model and it has overfit to these actions. This behavior may not be ideal for an adaptive environment, as students have a limited amount of tile changes they can make before they lose a level.

7.3. Transfer Learning for Plan Generation in FUTURE WORLDS

In addition to comparing fully-supervised and few-shot learning using LLMs for plan generation in FUTURE WORLDS, we also investigated a transfer-learning approach to generating plans in FUTURE WORLDS. We explored this methodology for FUTURE WORLDS due to the smaller amount of training data from the dataset (32 students compared to 144 students in CRYSTAL ISLAND) and to test the limits of LLMs for plan generation.

7.3.1 Methods

GPT-3.5 Prompt Engineering. For our FUTURE WORLDS transfer learning plan generation implementation, we again utilized the gpt-3.5-turbo model and generated plans using the chat completions API, to be consistent with the other methodologies. We kept the prompt mostly the same as the non-transfer learning approach, with a description of FUTURE WORLDS, the overarching goal of the game, a list of all goals included in the planning tool, and a list of all actions included in the planning tool. Instead of providing two example inputs from FUTURE WORLDS, we mapped each FUTURE WORLDS goal category to corresponding goal categories from CRYSTAL ISLAND and provided two examples of CRYSTAL ISLAND input and plans. These examples were selected randomly from the set of goals within a goal category. Like the other implementations, the end of the prompt consisted of the input event sequence (E), the set of completed goals (G_c) and the target goal (G_t).

Mapping Categories. Because CRYSTAL ISLAND and FUTURE WORLDS are both based in science reasoning, we were able to align goal categories based on science reasoning steps. Both CRYSTAL ISLAND and FUTURE WORLDS had a "*gather information*" category, so we were able to map these categories directly. The "*learn how humans impact the environment*" goal category of FUTURE WORLDS mapped to "*learn science content*" in CRYSTAL ISLAND, because each category represents learning the educational science content in the game and thus should have similar planned action associations. Finally, we mapped the "*change environmental impact*" goal category in FUTURE WORLDS to the "*collect data*" in CRYSTAL ISLAND. For each of these categories, it is expected that students would utilize these goals for testing hypotheses, and the goals included in each of these categories align with that understanding. Note, due to the limited use of the "*improve gameplay skills*" category and having no corresponding category in CRYSTAL ISLAND, we excluded these plans from the transfer learning experiment. This mapping logic was used to select examples used in the prompt provided to the GPT-3.5 model. For each target goal in an input, we found the corresponding goal category in FUTURE WORLDS, mapped it to the CRYSTAL ISLAND category, and then randomly selected two example CRYSTAL ISLAND inputs from that category for that prompt.

Evaluation. We applied the same evaluation techniques as our previous experiments for our transfer-learned FUTURE WORLDS generated plans: a quantitative analysis of the plan size and

match percentages, and a qualitative analysis of the plan composition for each goal category. In the results section, we refer to plans generated from the transfer learning approach as GPT-3.5-T.

7.3.2 Results

Quantitative Analysis. To analyze the plans generated with transfer learning approaches in FUTURE WORLDS, we applied the match percentage analysis to the GPT-3.5 generated plans. Figure 7.5 illustrates the distribution of three variables considered in this evaluation: (1) the differences in plan size between student-created plans and those generated by the transfer-learned GPT-3.5 model (left), (2) low-level match percentages (center), and (3) high-level match percentages (right).

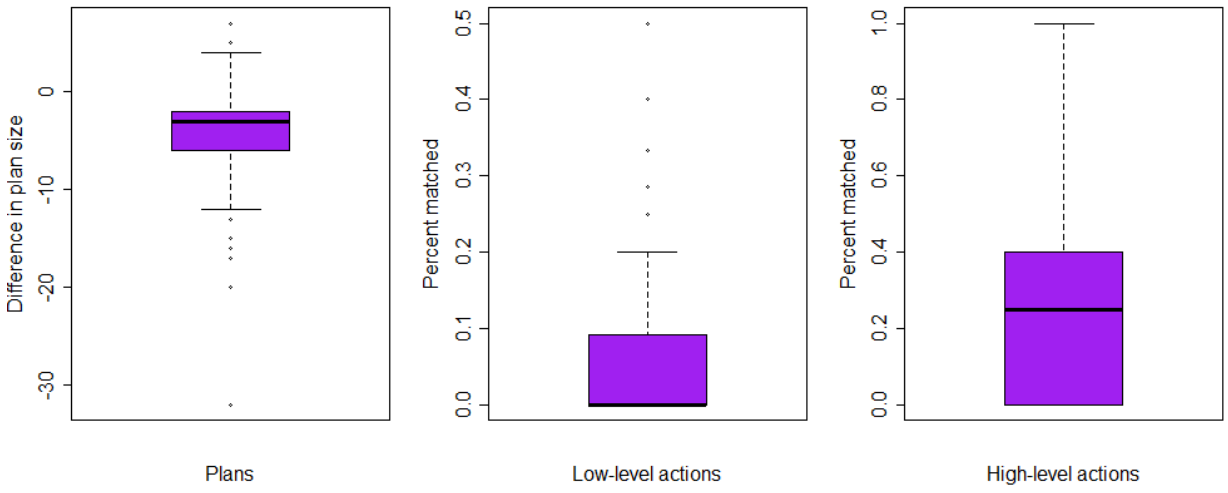

Figure 7.5. Distribution metrics for transfer-learned GPT-3.5 generated plans for FUTURE WORLDS.

The boxplot on the left shows the median plan size difference was -3.00 (mean=-4.17, SD=4.34). The maximum and minimum plan size differences were 7 and -32, respectively, demonstrating longer plans being generated than those trained on FUTURE WORLDS examples. The low-level action match percentage (p_L) had a median of 0.00 (mean=0.05, SD=0.10). Out of 246 GPT generated plans, 0.00% had plans where all low-level actions matched that of the students' plans and 71.66% contained no low-level action matches to the corresponding student plan. The median p_H was 0.25 (mean=0.25, SD=0.23). Similarly to the other models, we saw an increase in generated plans where all high-level actions were present in the students' plan to 1.07% and a

decrease in no matches to 36.36%. These results were lower than both the T5 and GPT-3.5 match rates, indicating the transfer learning approach generated the most distinct plans from student plans.

Plan Composition Analysis. Table 7.5 shows the action category distribution for the GPT-3.5 plans generated with transfer learning (GPT-3.5-T) compared to student plans and non-transfer GPT-3.5 plans. For both "*learn how humans impact the environment*" and "*gather information*" goal categories, GPT-3.5-T had a similar distribution to student plans. This combined with the low match rate indicates the potential of GPT-3.5-T plans being able to suggest similar actions during planning while invoking a similar strategy to students. More analysis is needed to understand which types of plans are optimal.

Table 7.5. High-level action category distribution for each high-level goal in the FUTURE WORLDS planning tool of transfer-learned GPT-3.5 generated plans (GPT-3.5-T) compared to student, T5, and GPT-3.5 plans.

		Change env. impact	Learn how humans impact the env.	Gather info.
GPT-3.5-T	Learn about env. sust.	4.52%	20.79%	14.36%
	Change env.	36.77%	31.68%	32.64%
	Explore env.	58.71%	47.52%	53.00%
GPT-3.5	Learn about env. sust.	12.50%	75.86%	10.23%
	Change env.	84.38%	20.69%	36.96%
	Explore env.	3.13%	3.45%	52.81%
T5	Learn about env. sust.	0.00%	1.04%	0.26%
	Change env.	100.00%	95.23%	52.32%
	Explore env.	0.00%	3.73%	47.43%
Student	Learn about env. sust.	4.40%	22.39%	17.36%
	Change env.	84.62%	40.30%	21.88%
	Explore env.	10.99%	37.31%	60.76%

For the "*change environmental impact*" category, we observed that plans were generated with action from the "*explore environment*" action category. This differs from the other generated plans, as well as student plans. To change the environment in FUTURE WORLDS, a player must open a tile, which would fall under an "explore environment" action, so it could be beneficial to include this type of action in a "change environmental impact" goal. However, a more cohesive plan must

include a “change environment” action to be considered appropriate for this type of goal. The plan quality metrics presented in Chapter 8 may provide more insight into the quality of GPT-3.5-T plans for this category because they take these nuances into account.

7.4. Examples of Generated Plans

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.

7.4.1 CRYSTAL ISLAND Generated Plans

We extracted example plans that showcase the nuances of plan validity analysis. Table 7.6 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

Table 7.6. Example student-created plans and LLM-generated plans for a target goal for each high-level goal in the CRYSTAL ISLAND planning tool.

Target goal	(a) Test objects patients have touched
Student	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	pick up raw chicken, pick up cheese, pick up egg, pick up water, pick up lettuce, pick up yogurt
GPT-3.5	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	fill in symptoms
T5	read about storing food
GPT-3.5	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	read about viruses, read about carcinogens, read about mutagens, read about microbes
T5	fill in symptoms, fill in prediction about anthrax, fill in prediction about smallpox
GPT-3.5	speak with lead scientist, read about smallpox, speak with lab technician, examine poster about smallpox
Target goal	(d) Learn about disease
Student	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	go to laboratory, go to lead scientist's quarters, speak with lab technician, read about how diseases spread
GPT-3.5	go to dining hall, speak with camp cook, go to laboratory, speak with lab technician
Target goal	(e) Explore camp
Student	go to infirmary, go to lead scientist s (bryce s) quarters, go to laboratory, go to living quarters
T5	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	go to beach, go to dining hall, go to forest

noticed T5 did this several times, which could be beneficial in real-time settings. This type of cross-student hint 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 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, thereby increasing performance and engagement in the game.

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 is 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 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. Interestingly, some GPT-3.5 generated

plans included "collect data" goals in the plans, which may show potential for GPT-3.5 providing higher-level suggestions. We will discuss further implications of these findings in Section 7.3.

7.4.2 FUTURE WORLDS Generated Plans

We extracted T5, GPT-3.5, and GPT-3.5-T generated plans from each goal category to compare to student plans (Table 7.7). These examples are meant to provide insights into the performance of the LLMs for plan generation and how they might inform adaptive support for students. Similar to the CRYSTAL ISLAND examples, each distinct target goal shown in the table was selected from a different goal category to provide more comprehensive examples.

The generated plans for the "*identify environmental issues*" goal (a) demonstrated a case where the LLMs generated longer plans than the student created. This goal is part of the "gather information" category and is meant to be exploratory in nature. Thus, having more planned actions is ideal in this case. Additionally, making changes to tiles will help to determine the environmental effect of the tile, but each level limits how many changes you can make, leading to potentially losing the level. The GPT-3.5-T plan may be the best strategy in this case, as it suggests simply viewing tiles and educational content without making tile changes.

To achieve "*adjust energy sufficiency*" (b), students must make several tile changes. None of the plans included the speak action, but they did include several "fill in" actions. Each of the plans, except GPT-3.5, incorporated planned actions with tile changes. The T5 plan seems to be the most efficient plan, as following those direct actions in the game would likely allow the student to win the level. Interestingly, the GPT-3.5-T plan seems to follow a higher-level strategy of learning the educational content before making a tile change. It is important to note that "*read about energy efficiency*" and "*make change to energy source*" are not actions that can be selected in the planning tool and require a bit of inference to be enacted, but they can still be mapped to corresponding gameplay actions. Similarly, the GPT-3.5 plan could be used as more of a hint in a real-time setting, as leading the student to learn about housing may help them correctly identify how to change the housing tile to adjust the energy sufficiency. In all of these cases, the LLM-generated plans have the potential to support a student.

For the target goal "*learn about energy efficiency*" (c), all of the models included "*learn*" actions, which are necessary to accomplish this goal. Additionally, all of the "learn" actions included had content that may impact energy in the level. As we have mentioned previously,

Table 7.7. Example student-created plans and LLM-generated plans for a target goal for each high-level goal in the FUTURE WORLDS planning tool.

Target goal	(a) Identify environmental issues
Student	make change to city, view farm energy source
T5	make change to city, view farm energy source, view housing, view housing, view housing, make change to river, make change to city
GPT-3.5	make change to river, view city, make change to farm using wind energy, make change to forest, make change to suburban housing, check eco-meters, read problem statement
GPT-3.5-T	view power plant, view farm energy source, check eco-meters, view housing, learn about power plants, learn about farm energy sources
Target goal	(b) Adjust energy sufficiency
Student	make change to city
T5	make change to city, make change to forest, make change to pig farm
GPT-3.5	learn about housing
GPT-3.5-T	view city, read about energy efficiency, make change to energy source
Target goal	(c) Learn about energy efficiency
Student	learn about housing, make change to pig farm
T5	learn about housing
GPT-3.5	learn about power plants, make change to power plant, view power plant, make change to power plant, make change to farm energy source
GPT-3.5-T	view energy efficiency diagrams, read about solar energy, learn about biofuels, make change to farm using solar energy, read about riparian buffers, make change to farm using riparian buffers, read about manure spreading, make change to farm with adjusted manure spreading
Target goal	(d) Complete problem faster
Student	make change to city
T5	make change to city
GPT-3.5	make change to city

another strategy for learning in FUTURE WORLDS could be making tile changes, as the student included in their plan. The GPT-3.5 plan in this case included both making tile changes and learning in their plan, with the tile change actions being better suited for the goal. Interestingly, "make change to pig farm" is required to beat the first level, so a student including this in a later plan can indicate them invoking knowledge they gained from the first level.

The "*complete problem faster*" goal was part of the least used goal category in FUTURE WORLDS. We saw similar performance from both LLMs in this case, likely due to a lack of variety

in training data. Upon further investigation, most of the plans generated for this target goal were the same as in the example.

Overall, these patterns demonstrate the generalizability of utilizing LLMs to inform real-time systems with plan generation. Like in CRYSTAL ISLAND, these models generated coherent plans, with generated actions aligning with students' actions. We also identified instances of hallucinations in both T5 and GPT-3.5 plans, but once again they mostly aligned with actions that could be completed in the game. The GPT-3.5-T plans in particular generated planned actions that seemed to be a combination of CRYSTAL ISLAND and FUTURE WORLDS but were not completely off topic. For example, several planned actions contained "*read about*", which is language primarily used in CRYSTAL ISLAND for their educational materials. In FUTURE WORLDS, there is still informational text to read, but there are also educational diagrams to view. We also noticed some GPT-3.5 and GPT-3.5-T plans contained goals, which follows similar trends to the CRYSTAL ISLAND findings. While this does not align with the planning tool, it could give more general hints to students to attempt different strategies. We will discuss further implications of these findings in Section 7.3.

7.5. Discussion

Each LLM approach presented in this chapter generated plans that were generally viable. In this section, we will discuss the pros and cons of each approach, as well as how these models may be integrated into real-time game-based learning environments.

7.5.1 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.

In a practical implementation of this plan generation framework, it might prove more beneficial to use a language model like T5 that can be fine-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. Despite these drawbacks, GPT-3.5 still generated plans that can be enacted in gameplay and could be a good alternative option in cases where there is less data available.

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 enactable 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. 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.

7.5.2 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 for CRYSTAL ISLAND and over one quarter of plans generated for FUTURE WORLDS. 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 in CRYSTAL ISLAND, 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, even though the "*speak*" action differs. Generating different planned actions from students is desirable because LLMs may provide more meaningful advice to students only if there exist some differences between LLM generated plans and students' plans. 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 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 science game-based learning environments like CRYSTAL ISLAND and FUTURE WORLDS. However, by utilizing student plans, we can better understand and support realistic student behavior. 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 them for various diseases in CRYSTAL ISLAND. 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 number 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. This pattern also occurred in FUTURE WORLDS, particularly for the plans generated from the transfer learning approach. 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. This is evident from our findings when applying these methods of FUTURE WORLDS. We observed very similar trends in performance and limitations. Thus, our framework has generality beyond CRYSTAL ISLAND to other science 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 science game-based learning environments, 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 making progress in the game scenario. The existing version of CRYSTAL ISLAND or FUTURE WORLDS 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 is not enough data to train on all 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.

CHAPTER 8

PLAN EVALUATION IN SCIENCE GAME-BASED LEARNING

This chapter introduces three novel plan quality metrics that are rooted in science reasoning theory. We first apply these metrics to the student plans in CRYSTAL ISLAND, as well as to the LLM-generated plans. We present an empirical analysis of the insights derived from these metrics in the context of CRYSTAL ISLAND. We then apply these metrics to student plans from the FUTURE WORLDS environment, as well as LLM-generated plans. Lastly, based on empirical findings from the application of these metrics, we discuss improvements that can be made to the metrics and planning tool, as well as game design implications.

8.1. Plan Evaluation Metric Development

Because the student plan recognition and generation framework are evaluated with students' plans, we identified potential metrics to evaluate both student and model created plans. The three metrics presented below have the potential to improve adaptive scaffolding, as they will help to determine the quality of generated plans through temporal cohesion, scientific reasoning alignment and the size of plans.

8.1.1 Measuring Goal Setting and Planning Quality

McCardle and colleagues (2017) identified that effective goals for regulation should have specific time, content, actions, and standards. They highlight that the more specific each of these features are within a goal, the more effective it is in guiding a learner's self-regulation. This concept has not, however, been expanded to the plans that a learner might make to accomplish those goals or been connected to a specific context such as scientific reasoning. For this thesis, we have adopted a multidimensional operationalization of plan quality that we will be able to quantify using our analytical mapping technique in future research. In this conceptualization, we consider a "good" plan to be one that, if enacted, will result in completed goals set by either the planner or an imposed external standard. In this way, we must consider both the goals and the planned actions in

conjunction with one another when evaluating a plan. We further contextualized plan quality across three dimensions. These dimensions are:

(1) Plan Complexity: the degree of intricacy of one's plans categorized by the number of (sub)actions and (sub)goals as well as the degree of nested-ness.

(2) Action-Goal Cohesion: the degree to which action and goals may refer to a scientific reasoning epistemic activity.

(3) Temporal Coherence: the degree to which the temporal order of a plan follows the guidance of scientific reasoning epistemic activities.

8.1.2 Plan Complexity

We have defined the plan quality dimension of Plan Complexity as the degree of intricacy of one's plans categorized by the number of actions and goals as well as the degree of nested-ness. We highlight that there is no universal standard size of a plan, but rather this is both domain and task specific that must be determined prior to plan evaluation. This may be done with a subject matter expert providing their own plan to the given task to be the "gold standard" by which learners' plans are compared against. Alternatively, this may also be done by defining an acceptable range given the ultimate learning goal for the planner. For example, the minimum of a plan could be one action-goal pair for each of the components of one's chosen scientific reasoning framework (e.g., 8 for epistemic activities or 5 for the OPIRR framework). The upper limit could then be defined using the expert model approach, thereby creating a spectrum of plan complexity quality. In addition, an individual's level of prior knowledge or expertise could be used to determine how nested (e.g., number of planned actions within a goal) the plan should be in which it would be expected the more of an expert a learner is, the less explicitly detailed plan is needed as the learner chunks together steps as expertise research has shown is common (Wirth et al., 2020). This dimension may also, however, be constrained by environmental factors that may limit the intricacy of the structure of a reported plan, as is the case in our study.

8.1.3 Action-Goal Cohesion

We have defined the plan quality dimension of Action-Goal Cohesion as the degree to which action and goals may refer to a scientific reasoning epistemic activity. In other words, this refers to how well the (sub)action a learner chooses is associated or will help accomplish the paired (sub)goal. For example, if a learner's goal is to determine the correct diagnosis for a patient's health problem in a case-based simulation, a "good" plan would include actions related to evidence evaluation such as determining the relevancy of patient history and lab tests to the reported health issue. A "poor" plan would be one in which a learner pairs unrelated communicating finding actions to this goal. While communicating the correct diagnosis and recommended treatment plan is important in medical interactions, this action would be better paired with the (sub)goal of treating the patient or improving patient outcomes, not on finding the diagnosis. Within our study's context, our analytical mapping technique identifies the cohesion of all (sub)action-(sub)goal pairings allocated to a learner within the learning environment and planning tool.

8.1.4 Temporal Coherence

We have defined the plan quality dimension of Temporal Coherence as the degree to which the temporal order of a plan follows the guidance of scientific reasoning epistemic activities. More specifically, temporal coherence argues that a "good" plan is one that follows general temporal guidance of existing theoretical models of scientific reasoning and scientific inquiry skills. For example, in the OPIRR model, there is a clear cyclical nature of the steps one moves through from orientation to reflection. A plan, therefore, using this framework as grounding would argue that a "poor" plan is one that has a learner moving from non-sequential steps while a "good" plan is one in which they follow goal-action pairings that move around the iterative OPIRR cycle. For example, a learner who would first plan to set up a research design (preparation) and then orient themselves to the topic of interest (orientation) is a worse plan than a learner who first plans to understand the current state of the field of their chosen topic before preparing for a research design. However, this highlights the limitations of this particular model of scientific reasoning. Specifically, it assumes that a learner may not backstep within their scientific reasoning process until they have completed their cycle instead of regulating their strategies and learning throughout. As such, while this model is valuable and highlights the cyclical nature of scientific reasoning, it may not be the strongest model when quantifying the quality of plans as it is the ideal cycle of

scientific reasoning. In our study, we have chosen to ground our work in a less temporally defined framework of scientific reasoning by amending Fischer and colleagues' (2014) framework of scientific reasoning epistemic activities. We propose a state model (Figure 8.1) of these activities with transitions informed by the definitions from the original framework and the cyclic temporal relationships identified in other scientific reasoning models (i.e., SDSS and OPIRR). This model shows only one iteration of one scientific reasoning task. We can imagine that science is often iterative and builds upon itself, after communicating and scrutinising one task, this may then inform the problem identification of another task, creating a cyclical loop of the state model similar to the OPIRR model. This has been omitted given the context of more constrained problem spaces typical in GBLEs for primary and secondary science education. In future research, we will use this state model to quantify the temporal coherence of learners' plans.

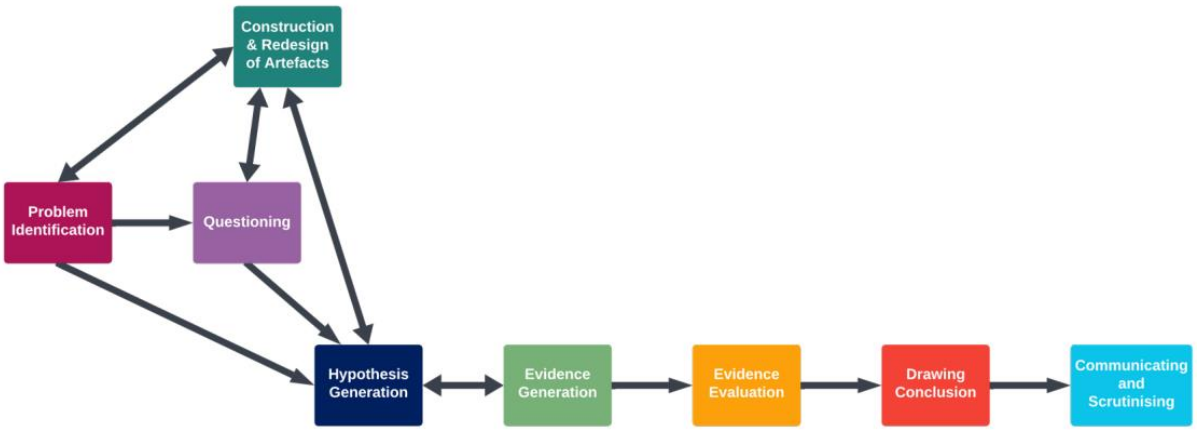

Figure 8.1. State model of Fischer et al. (2014)'s epistemic activities used in scientific reasoning and argumentation.

8.2. Applying Plan Evaluation Metrics to CRYSTAL ISLAND

To better understand how students planned in CRYSTAL ISLAND and to provide a potential approach for assessing LLM-generated plans, we applied these metrics to the CRYSTAL ISLAND learning environment. We developed methods to apply temporal coherence, action-goal cohesion, and plan complexity metrics to the student planning data, as well as LLM-generated plans, to glean insights into how well students and LLMs planned in the context of CRYSTAL ISLAND. We present the

insights found from these metrics and discuss the implications of these findings and potential improvements for adaptive support in real-time.

8.2.1 Methods

To analyze how students set plans consisting of actions and goals using the CRYSTAL ISLAND planning scaffold, we devised an analytical mapping technique that we then used to quantify plan quality across our defined plan quality dimensions. This mapping technique categories a goal-subgoal (rows) and action-subaction (columns) pairing as one of the eight scientific reasoning epistemic activities (cell color) defined by Fischer and colleagues (2014) of various cohesion levels (cell opacity; see Figure 8.2).

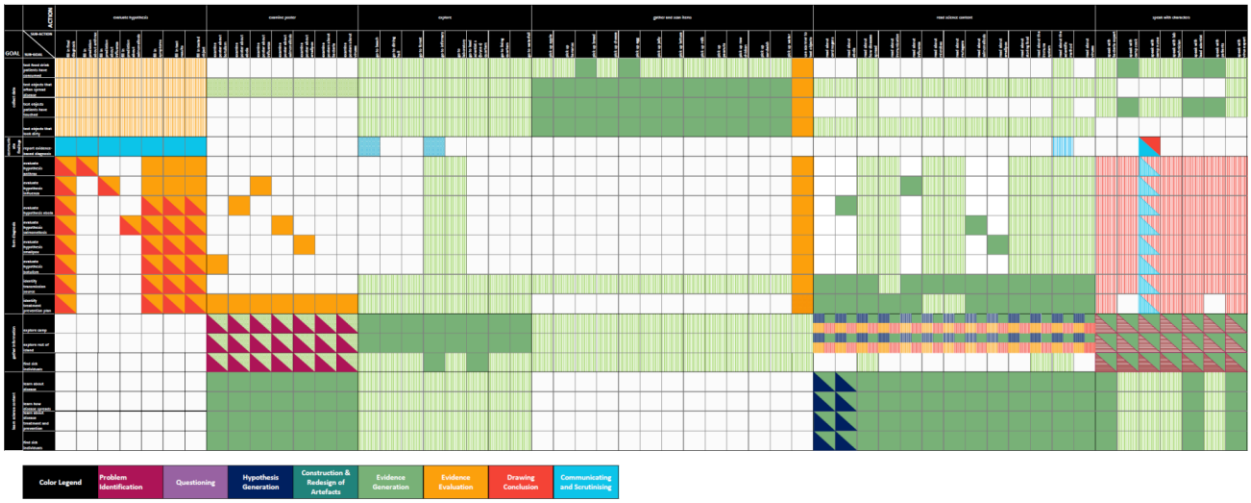

Figure 8.2. Analytical mapping matrix of actions (columns) and goals (rows).

That is, striped cells are actions that could be considered partially cohesive with a goal but are not necessarily required to be explicitly named while solid-colored cells are cells actions that are fully cohesive with its paired goal. For example, if a student has the goal to "*fill in the final diagnosis*" on the diagnosis worksheet to share with Kim, the island's nurse, the action "t*est food and drink patients have consumed*" may be one of the actions the student must take to make that final diagnosis but is more cohesive with the goal of "*identify transmission source*." Namely, while testing the food and drink may lead me to my diagnosis, it also requires the student to then go through evidence evaluation activities (i.e., determining if that evidence is relevant to the final diagnosis) first. However, it is not entirely unrelated as more expert learners may choose to chunk together evidence collection and evaluation activities into a single explicitly shared plan. As such,

in our matrix (see Figure 8.3) the cell at the intersection of the sub-action "*fill in final diagnosis*" and "*test food and drink patients have consumed*" is striped (partially cohesive). Alternatively, the cell at the intersection of the sub-action "*fill in final diagnosis*" and "*identify transmission source*" has a solid fill as it is considered fully cohesive "*drawing conclusion*" epistemic activity. Three experts of CRYSTAL ISLAND each filled in their own matrix and then discussed discrepancies. The agreed upon mapping is shown in Figure 8.2. This mapping is utilized in calculating the metrics for both temporal coherence and action-goal cohesion.

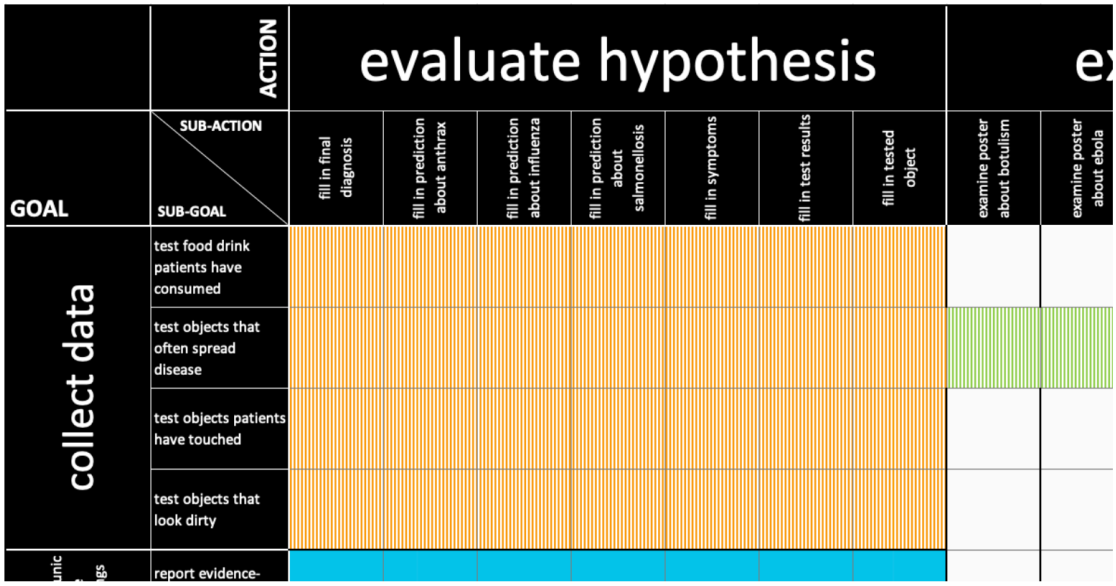

Figure 8.3. Close up sub-section of the mapping matrix for examples of Goal category-goals (rows) and Action category-actions (columns) to identify scientific reasoning epistemic activity (cell color) and plan relevancy (cell opacity).

We employ these three scoring metrics to the student, fully supervised (T5) and few-shot learning (GPT-3.5) generated plans to further examine planning behaviors. For this analysis, we removed any duplicate actions from the T5 and GPT-3.5 plans. Note, all three of these metrics are meant to assess planning behaviors at the plan level. We will discuss limitations of these assumptions, as well as future enhancements to these scoring methods in future sections.

Plan Complexity: the degree of intricacy of one's plans categorized by the number of actions and goals as well as the degree of nested-ness. The planning tool design in CRYSTAL ISLAND does not permit goals to be nested within each other, so we cannot account for the degree of nested-

ness. In this work, we considered the degree of intricacy of one's plans to be the number of planned actions in a plan. For the generated plans, we used a comma as a delimiter to count the number of actions. We recognize this is not a perfect measurement, because some generated plans did not follow the planning tool actions exactly and had planned actions containing commas. However, this measurement of plan size represents how we would preprocess the data in a real-time system without human intervention.

Action-Goal Cohesion: the degree to which action and goals may refer to a scientific reasoning epistemic activity. Scoring for the action-goal cohesion metric was based on the level of cohesion from the epistemic activity mapping. For each planned action in a plan, we assign one point if there is a fully cohesive relationship to an epistemic activity (solid colored cell) and half a point if there is a partially cohesive relationship to an epistemic activity (striped colored cell). No points are added or subtracted if there is no relationship. If there is more than one mapping to epistemic activity (i.e. "*explore camp*" and "*examine poster*" cell), then we will score with the highest coherence score (so in this example, we would score one point instead). We then normalize this metric by the number of planned actions in this plan to get values ranging from 0 to 1. In contrast to the temporal coherence score, we do not consider any order of the actions, simply how many epistemic activity mappings exist.

Temporal Coherence: the degree to which the temporal order of a plan follows the guidance of scientific reasoning epistemic activities. To quantify temporal coherence for each plan, we first mapped each planned action to its corresponding epistemic activity associated with the target goal. We then examined the transitions between epistemic activities, assuming the order of actions in the plan were sequential. Because not all epistemic activities were present in the planning matrix for CRYSTAL ISLAND, we defined the ideal sequence of epistemic activities to be Figure 8.4. Our aim with this metric is to reward planning behaviors that follow the epistemic activity trajectory, moving left to right. Thus, our metric scoring worked as follows: we allotted points for moving to the right, we gave one point for moving exactly one step to the right, 0.8 points for two steps, 0.6 points for three steps and so on. This is meant to emphasize that the ideal next planned actions should map to the next step in the science reasoning process. Backward movements (i.e., evidence evaluation to problem identification) result in negative one points. We also rewarded one point for self-transitions because the environment is not strictly defined enough to know how long a student

should spend in one phase of science reasoning steps. In cases where there are multiple epistemic activity mappings, we were as lenient as possible and chose the epistemic activity that would yield the most points as a next step. If there was no epistemic activity mapping, no points were taken away or rewarded and the last epistemic activity was used in consideration for the next transition. We did not include striped versus solid-colored cells in the logic for this metric. The final step in our metric is to normalize by the number of transitions, as this represents the maximum possible temporal coherence score a plan could have. Thus, our temporal coherence score will range from -1 to 1, with a score of 1 indicating each epistemic activity transition in the plan followed an ideal progression and a score of -1 indicating each epistemic activity transition moving backwards.

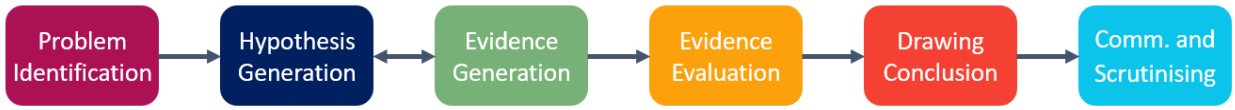

Figure 8.4. State model of expected epistemic activities progression in CRYSTAL ISLAND.

As an example, consider a plan for the goal "*evaluate hypothesis salmonellosis*," with the following planned actions: "*go to laboratory*," "*examine poster about anthrax*," "*examine poster about salmonellosis*," "*speak with camp nurse*." This maps to the corresponding epistemic activities: "*evidence generation*," "*no mapping*," "*evidence evaluation*," "*drawing conclusion*"/ "*communicating and scrutinizing*." In this case, we ignore the action that does not map to an epistemic activity and treat the first transition as "evidence generation" to "evidence evaluation," which is one step to the right. Thus, the plan receives one point for this transition. The next transition "*evidence evaluation*" to "*drawing conclusion*"/"*communicating and scrutinizing*" represents a case where there are two possible epistemic activity mappings. If the transition was from "*evidence evaluation*" to "*drawing conclusion*," the reward would be one point since they are one step away from each other. If the transition was from "*evidence evaluation*" to "*communicating and scrutinizing*," the reward would be 0.8 points because "*communicating and scrutinizing*" is two steps away. With our logic, we would select "*drawing conclusion*" as the next epistemic activity and reward one point, because it has the higher reward. In total all transitions would yield a score of 2, which we would normalize by dividing by two transitions. Thus, our temporal coherence score for this plan would be 1.

8.2.2 Results

In this section, we present descriptive statistics of plan complexity, action-goal cohesion, and temporal cohesion for student plans, as well as fully-supervised (T5) and few-shot learning (GPT-3.5) generated plans. We compare the LLM-generated plan performance to the students and discuss implications.

Plan Complexity. Figure 8.5 illustrates the distribution of plan size across students (left), T5 (center), and GPT-3.5 (right). Student plan size had a median of 2.00 (mean = 2.55, SD = 1.89), with the number of actions included in a plan ranging from 1 to 13. With duplicates removed, T5 plans had a median size of 3.00 (mean = 2.95, SD = 1.38) and ranged from 1 to 7 in length. The median plan size of GPT-3.5 plans was 1.00 (mean = 2.10, SD = 1.41) and ranged from 1 to 8 in length. Results from a Wilcoxon rank sum test show a significant difference in the average plan sizes between student plans and fully-supervised generated plans ($p < 0.001$). This means that on average, T5-generated plans were more detailed than students. GPT-3.5 plans did not have a significant difference in plan size from students, but on average were shorter than student plans. In a narrative-based environment like CRYSTAL ISLAND, "good" planning behavior would require more detailed plans, indicating the capability of T5 for real-time support.

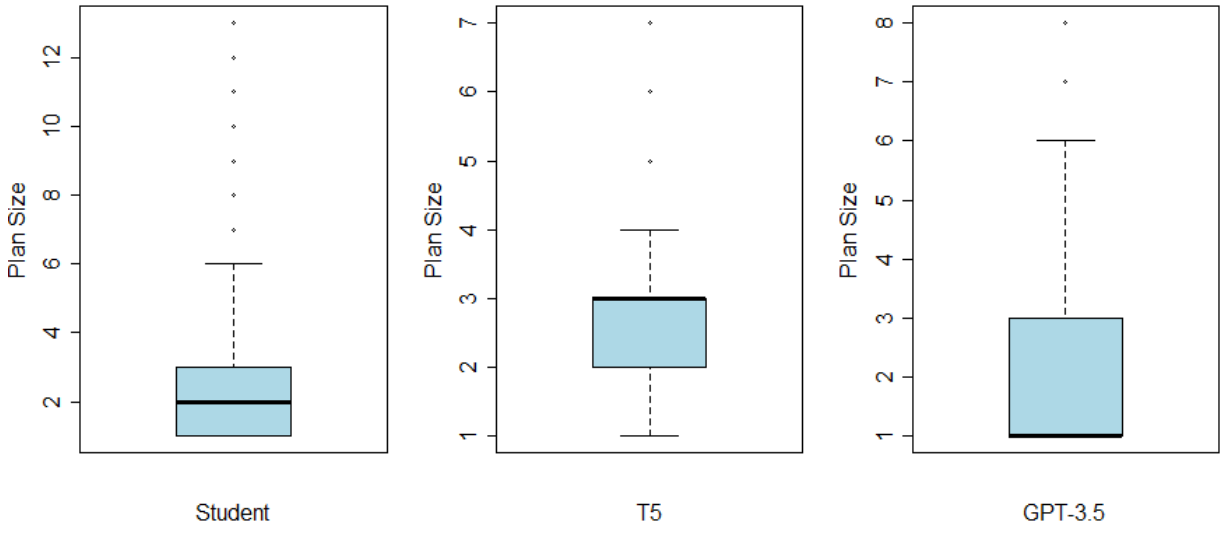

Figure 8.5. Distribution of plan size for student, T5, and GPT-3.5 generated plans for CRYSTAL ISLAND.

Action-Goal Cohesion. Figure 8.6 shows the distribution of action-goal coherence score across student (left), T5 (center), and GPT-3.5 (right) plans. All three sets of plans had a median of 1, meaning that each set of plans has half of their plans score the maximum possible action-goal cohesion score. Student plans had an average action-goal cohesion score of 0.77 (SD = 0.28). T5 and GPT-3.5 plans had an average action-goal cohesion score of 0.79 (SD = 0.24) and 0.69 (SD = 0.37), respectively. There was no significant difference in the average scores. GPT-3.5 plans had the most plans with a score of zero, indicating a plan with no epistemic activity alignment. T5 plans had the most plans where all actions in the plan aligned fully with epistemic activities, outperforming student plans.

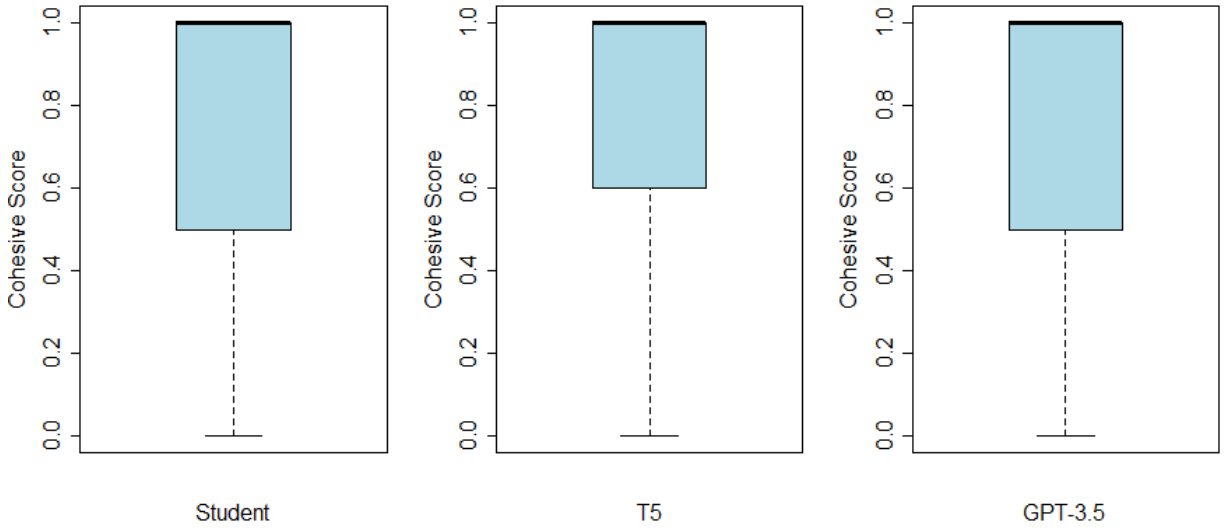

Figure 8.6. Distribution of action-goal cohesion score for student, T5, and GPT-3.5 generated plans for CRYSTAL ISLAND.

Temporal Coherence. Figure 8.7 shows the distribution of temporal coherence score across student (left), T5 (center), and GPT-3.5 (right) plans. Student temporal coherence scores had a median of 1.00 (mean = 0.54, SD = 0.53). T5 plans had a median temporal coherence score of 1.00 (mean = 0.78, SD = 0.48). The median temporal coherence score of GPT-3.5 plans was 0.00 (mean = 0.27, SD = 0.47). Results from Wilcoxon rank sum tests demonstrate a significant difference in the average temporal coherence scores between student plans and both T5 and GPT-3.5 generated

plans ($p < 0.001$, $p < 0.001$, respectively), with T5 scores being significantly higher and GPT-3.5 scores being significantly lower.

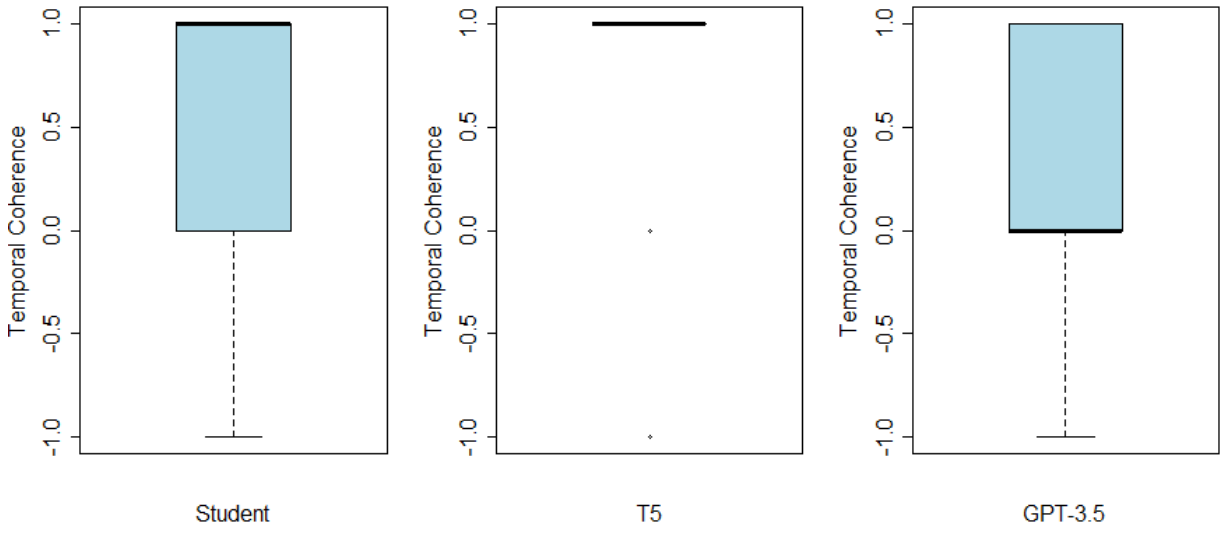

Figure 8.7. Distribution of temporal coherence score for student, T5, and GPT-3.5 generated plans for CRYSTAL ISLAND.

Overall, T5 plans seemed to demonstrate the best planning behaviors, with having the most detailed plans and the highest scores for both action-goal cohesion and temporal coherence. GPT-3.5 plans performed the worst with the lowest scores from all three metrics. These metrics indicate that T5 may be best suited for adaptive support in real-time for CRYSTAL ISLAND.

8.3. Applying Plan Evaluation Metrics to FUTURE WORLDS

To further examine student planning activities in FUTURE WORLDS, we applied the analytical mapping technique utilized in CRYSTAL ISLAND to FUTURE WORLDS. Our aim for this application is to examine the generalizability of this technique across learning environments as they are both rooted in science reasoning. We then use this mapping to quantify three plan quality metrics and use these metrics to evaluate student planning, as well as plans generated from LLMs.

8.3.1 Methods

Like in CRYSTAL ISLAND, this epistemic activity mapping technique categories a goal-subgoal (rows) and action-subaction (columns) pairing as one of the 8 scientific reasoning epistemic activities (cell color) defined by Fischer and colleagues (2014) of various cohesion levels (cell opacity; see Figure 8.8).

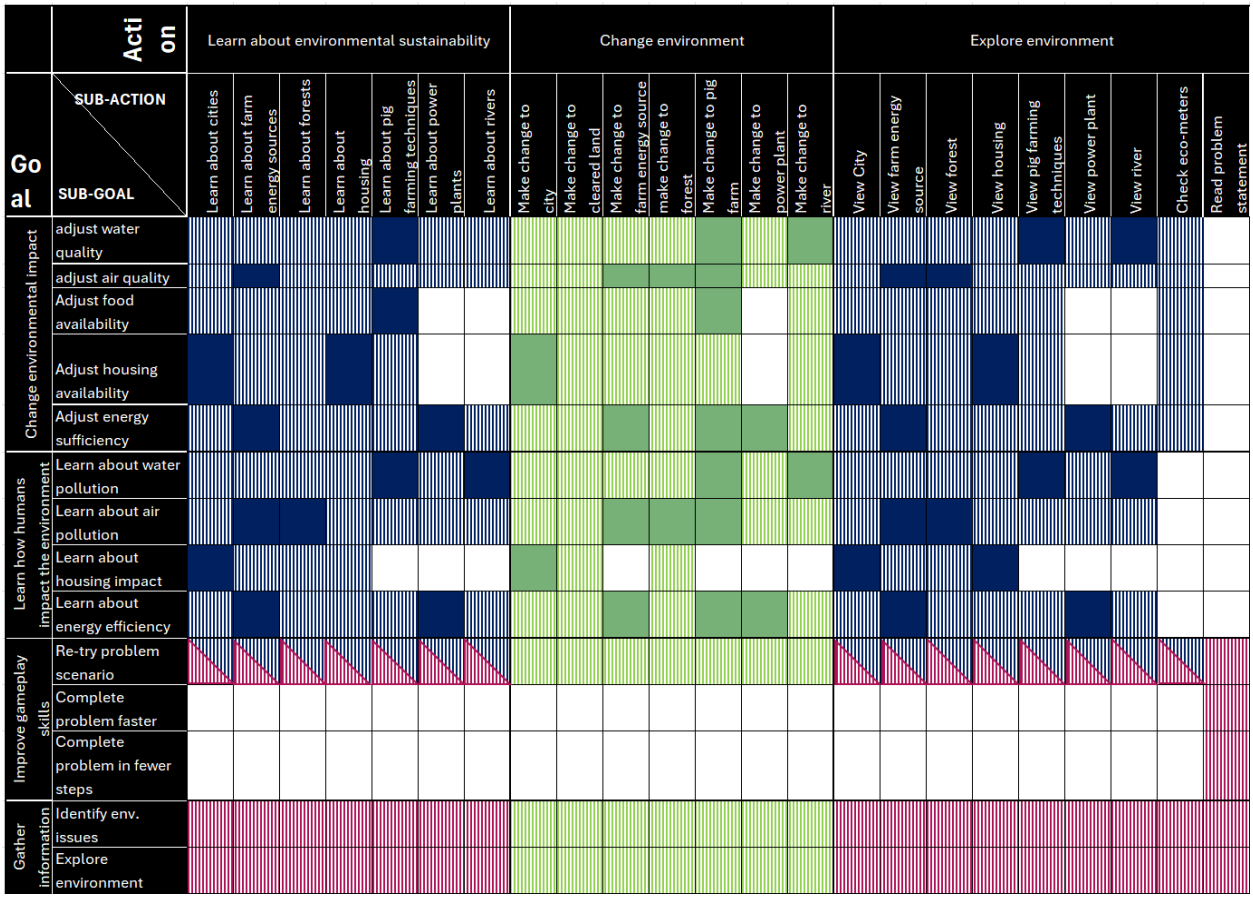

Figure 8.8. Analytical mapping matrix of actions (columns) and goals (rows).

For the FUTURE WORLDS implementation of the plan quality metrics, we applied the same coding scheme as we did in CRYSTAL ISLAND. That is, striped cells are actions that could be considered partially cohesive with a goal but are not necessarily required to be explicitly named while solid-colored cells are cells actions that are fully cohesive with its paired goal. For example, if a student has the goal to "adjust water quality", the action "learn about rivers" may be one of the actions the student must take to generate a hypothesis about how to adjust the water quality but is more cohesive with the goal of "learn about water pollution". Specifically, the river tile has

educational information about water pollution but does not provide the information necessary to make the correct tile change to adjust water quality. However, it is not entirely unrelated as more expert learners may choose to chunk together this information with other actions that could lead to the correct hypothesis. As such, in our matrix (see Figure 8.9) the cell at the intersection of the sub-action "*adjust water quality*" and "*learn about rivers*" is striped (partially cohesive). Alternatively, the cell at the intersection of the sub-action "*adjust water quality*" and "*learn about farming techniques*" has a solid fill as it is considered fully cohesive "*hypothesis generation*" epistemic activity. Two experts of FUTURE WORLDS each filled in their own matrix and then discussed discrepancies to generate the final matrix shown in Figure 8.8. This mapping is once again utilized in calculating the metrics for both temporal coherence and action-goal cohesion.

We employ these three scoring metrics to the student, fully supervised (T5) and few-shot learning (GPT-3.5) generated plans in the same way we did for the CRYSTAL ISLAND planning dataset. The scoring logic for each metric is presented below. We did not include the transfer learning (GPT-3.5-T) generated plans in this analysis due to a large presence of generated planned actions not present in the planning tool.

Plan Complexity: the degree of intricacy of one's plans categorized by the number of actions and goals as well as the degree of nested-ness. Like the CRYSTAL ISLAND implementation, we considered the degree of intricacy of one's plans to be the number of planned actions in a plan because the planning tool in FUTURE WORLDS did not allow for nested plans. Similarly, for the generated plans, we used a comma as a delimiter to count the number of actions.

Action-Goal Cohesion: the degree to which action and goals may refer to a scientific reasoning epistemic activity. We did not make any changes to the scoring for the action-goal cohesion metric in FUTURE WORLDS. The only change between environments was the planning matrix itself. Thus, the score for each plan was assigned as: one point if there is a fully cohesive relationship to an epistemic activity (solid colored cell), half a point if there is a partially cohesive relationship to an epistemic activity (striped colored cell), and no points if there is no relationship. In cases of more than one epistemic activity mapping, we select the mapping with the highest score. The score is then normalized by the number of actions in the plan.

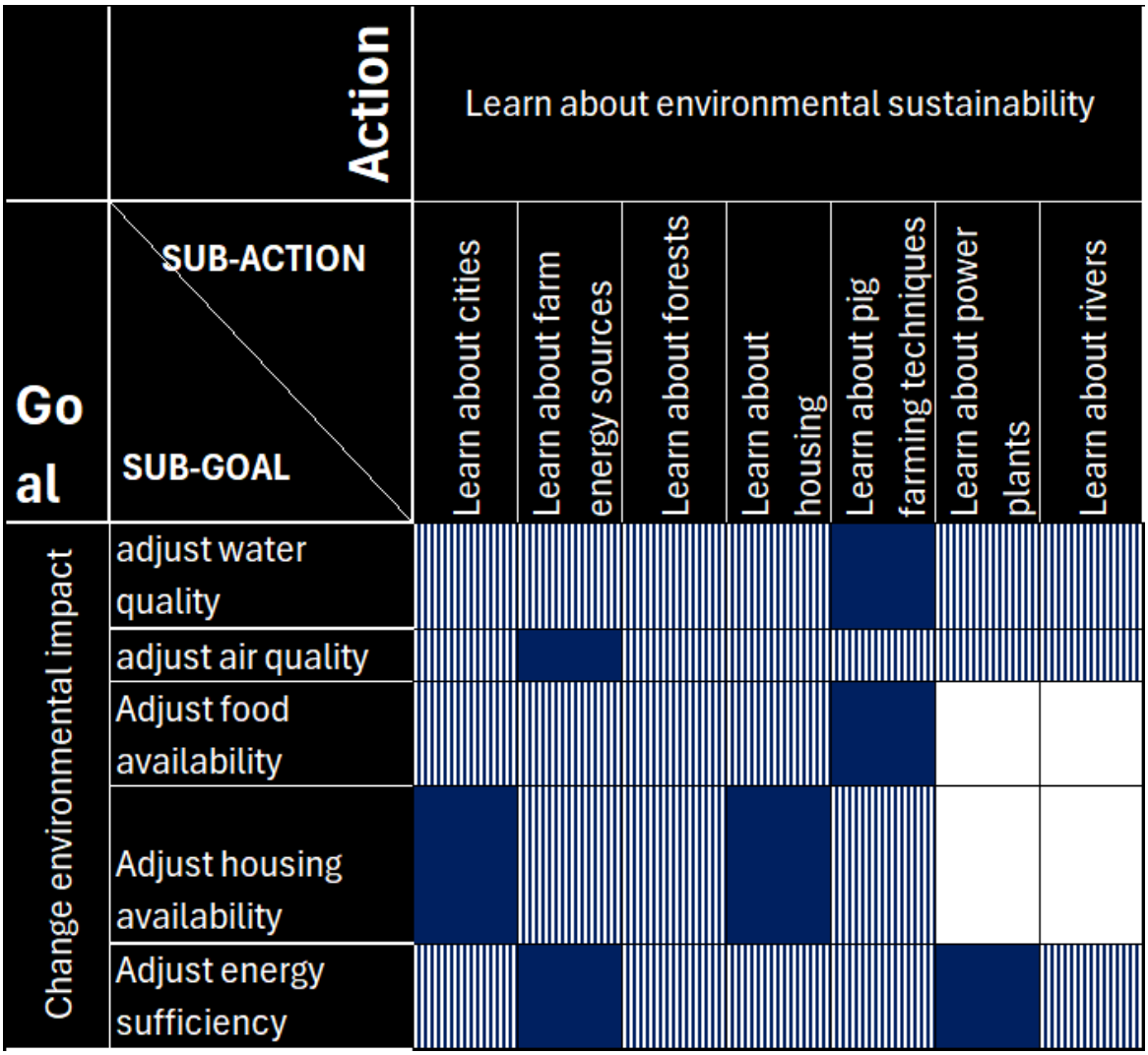

Figure 8.9. Close up sub-section of the mapping matrix for examples of Goal category-goals (rows) and Action category-actions (Columns) to identify scientific reasoning epistemic activity (cell color) and plan relevancy (cell opacity).

Temporal Coherence: the degree to which the temporal order of a plan follows the guidance of scientific reasoning epistemic activities. We applied the temporal coherence metric in the same way we applied it to the CRYSTAL ISLAND data. Because the FUTURE WORLDS planning matrix only utilized three epistemic activities, we expected plans to follow the progression shown in Figure 8.10. Our scoring method was as follows: (1) map each planned action to an epistemic activity, if possible, (2) examine the transitions between epistemic activities in the plan, (3) if a transition was a self-transition or exactly one step to the right, we allotted one point (4) if a transition was more than one step to the right, 0.67 points for two steps, 0.33 points for three steps,

(5) if a transition is backwards (to the left), we subtract one point from the score, (6) after all transitions have been scored, we normalize by the number of transitions made in the plan. Like the CRYSTAL ISLAND logic, if there was no epistemic activity mapping, no points were taken away or rewarded and the last epistemic activity was used in consideration for the next transition and our temporal coherence score will range from -1 to 1.

Figure 8.10. State model of expected epistemic activities progression in FUTURE WORLDS.

8.3.2 Results

Plan Complexity. Figure 8.11 illustrates the distribution of plan size across students (left), T5 (center), and GPT-3.5 (right). Student plan size had a median of 1.00 (mean = 2.03, SD = 1.52), with the number of actions included in a plan ranging from 1 to 8. With duplicates removed, T5 plans had a median size of 1.00 (mean = 1.76, SD = 1.04) and ranged from 1 to 7. The median plan size of GPT-3.5 plans was 2.00 (mean = 2.64, SD = 1.49) and ranged from 1 to 9 in length. Results from a Wilcoxon rank sum test show a significant difference in the average plan sizes between student plans and GPT-3.5 generated plans ($p < 0.001$). This means that on average, GPT-3.5-generated plans were more detailed than students. T5 plans did not have a significant difference in plan size from students, but on average were shorter than student plans. In a puzzle-based environment like FUTURE WORLDS, "good" planning behavior might mean shorter plans. However, we would still prefer plans with more than one action in them, indicating the capability of GPT-3.5 for real-time support in this case.

Action-Goal Cohesion. Figure 8.12 shows the distribution of action-goal cohesion score across student (left), T5 (center), and GPT-3.5 (right) plans in FUTURE WORLDS. All three sets of plans had a median of 0.50. Student plans had an average action-goal cohesion score of 0.53 (SD = 0.22). T5 and GPT-3.5 plans had an average action-goal cohesion score of 0.54 (SD = 0.19) and 0.46 (SD = 0.27), respectively. Results from a Wilcoxon rank sum test show a significant difference in the average cohesive score between student plans and GPT-3.5 generated plans ($p < 0.001$). GPT-

3.5 plans had the most plans with a score of zero, indicating a plan with no epistemic activity alignment. T5 plans had the most plans where all actions in the plan aligned fully with epistemic activities, even more than students did.

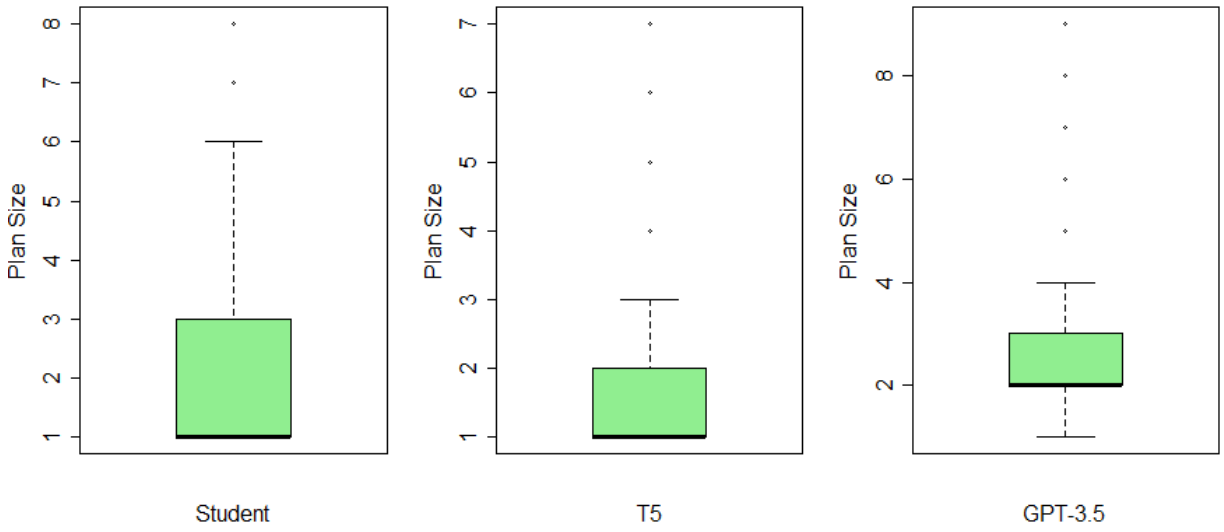

Figure 8.11. Distribution of plan size for student, T5, and GPT-3.5 generated plans for FUTURE WORLDS.

Temporal Coherence. Figure 8.13 shows the distribution of temporal coherence score across student (left), T5 (center), and GPT-3.5 (right) plans. Student temporal coherence scores had a median of 0.00 (mean = 0.36, SD = 0.52). T5 plans had a median temporal coherence score of 0.00 (mean = 0.42, SD = 0.49). The median temporal coherence score of GPT-3.5 plans was 0.00 (mean = 0.15, SD = 0.62). Results from Wilcoxon rank sum tests demonstrate a significant difference in the average temporal coherence scores between student plans and GPT-3.5 generated plans ($p = 0.002$), with GPT-3.5 scores being significantly lower.

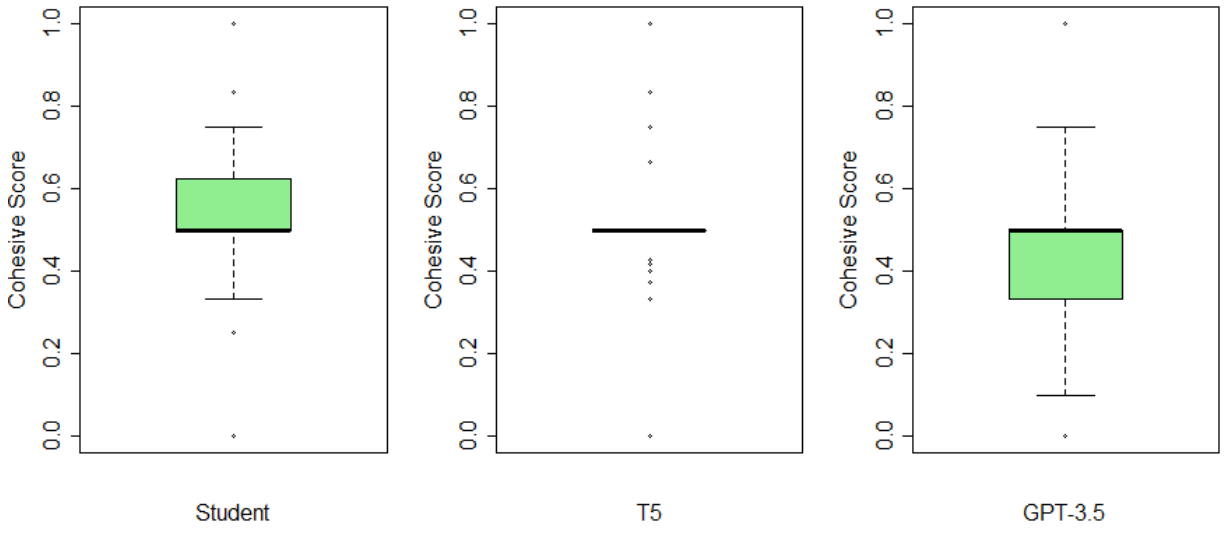

Figure 8.12. Distribution of action-goal cohesion score for student, T5, and GPT-3.5 generated plans for CRYSTAL ISLAND.

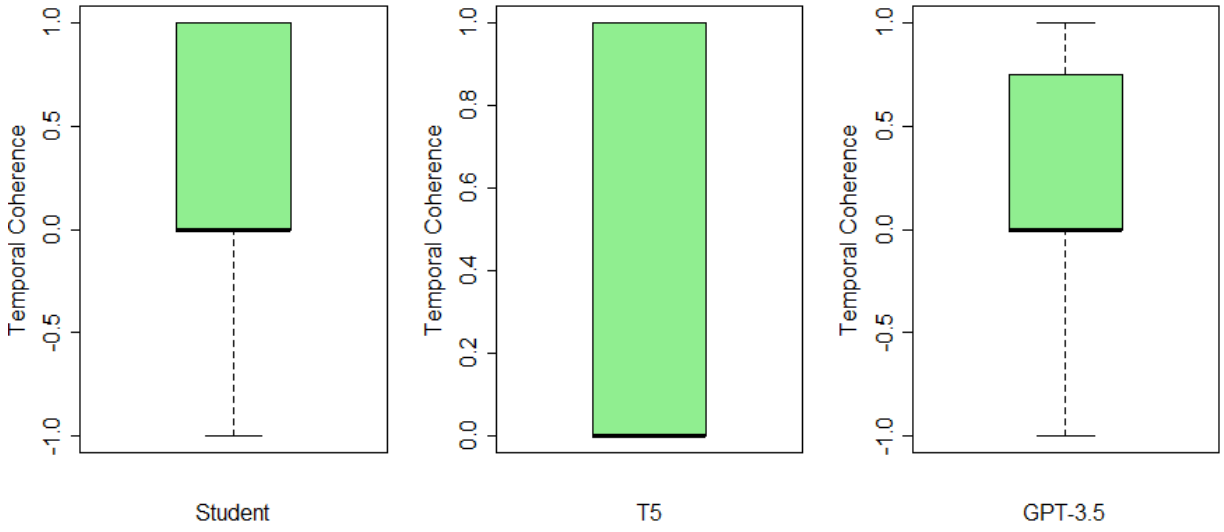

Figure 8.13. Distribution of temporal coherence score for student, T5, and GPT-3.5 generated plans for FUTURE WORLDS.

8.4. Discussion

All three plan quality metrics developed in this work provided nuanced insights into planning behaviors of students compared to the LLM-generated plans. Using these metrics for analysis, several points of improvement were found for the metrics, as well as the design of the planning tool.

The development of the epistemic activity planning matrix mapping is a novel contribution of this work. We successfully applied this matrix development process to two science learning environments, which highlighted benefits of this tool for design implications. One benefit of the planning matrix is that it allows for a visualization of the design of the tool, in terms of the balance of epistemic activities present in the planning tool. For example, in FUTURE WORLDS there are only three epistemic activities present out of seven possible. Additionally, only one goal has all three of these activities present in the mapping. This visualization indicates that we may need to consider higher level goals in the tool to allow for more epistemic activities trajectories to be possible. It could also indicate a need to add more game mechanics to allow for the full scientific reasoning process.

One limitation of the planning tool design highlighted by the plan complexity metric is that our tool does not allow for nested goals. The intention of this design was to provide scaffolds for students to construct plans that were easily enactable in gameplay because planned actions directly translated to game events. A potential design enhancement for the planning tool would be to allow students to visualize the hierarchy of epistemic activities to goal categories to goals to actions, etc., which may allow for nested goals and subgoals. Incorporating high-level goals into the planning tool would provide additional insight into students' overall strategies for the game and may allow for longer plans to be created. A future improvement for the planning tool could be to instruct students to make a nested plan for the entirety of their gameplay and have that plan visible to them throughout the game as an overlay on the screen.

Currently, the temporal coherence metric assesses planning behavior quality at the plan-level, with a high score rewarding a plan that traverses the expected epistemic activity progression. We observed students did not utilize the planning tool in this way, nor did our planning tool support this type of behavior. First, we observed students often created small plans of one or two actions. With the temporal coherence metric calculation, we can only score temporal coherence if there are

at least two actions in a plan, because the score examines the transition between epistemic activities. In other words, a plan with one action would receive a score of zero because there is no transition. To prevent this, we may want to add real-time support to students to create longer plans. Next, for the ideal plan-level temporal coherence behavior, our planning tool would need to have each epistemic activity present for each goal in the matrix. We can see from the rows of the matrix that no goal had every epistemic activity mapped to it, meaning that a student could not construct a plan fully following the expected epistemic activity progression with the planning tool as is. The planning tool in this case may need to be redesigned to allow for students to construct temporally coherent plans. Another solution would be to extend the definition of temporal coherence to assess planning across planning instances rather than at the plan level.

Another consideration of planning behavior that could be captured in the metrics is the change in goals and plans between planning instances. For example, some students had the same goal persist across planning instances and often completed the actions within the plan. In the plan recognition and generation frameworks, we considered a goal as accomplished if all actions were completed in gameplay. In plan evaluation, completing a plan would be considered a good planning behavior. However, then selecting the same goal after completing it, may be an indicator of needing intervention. We could also consider the adaptation of plans, as it is a key step in SRL. For instance, if a goal persisted across planning instances, with the planned actions changing, and was not completed in gameplay, we could consider this to be a student monitoring their goal and adapting their plan. We could then add to the metrics to reward such planning behavior within temporal coherence.

CHAPTER 9

IMPLICATIONS OF ADAPTIVE LEARNING ENVIRONMENTS WITH STUDENT PLAN RECOGNITION AND GENERATION

This chapter provides an overview of how the work undertaken in this dissertation research can be combined to enable adaptivity in learning environments. We first revisit the three student planning scenarios presented in Chapter 3 and compare the student planning with LLM-generated plans at the same point to demonstrate potential use-cases for interventions. Then, we discuss the implications of the dissertation work for the field of learning analytics and for SRL skill intervention.

9.1. Student Planning Scenarios Revisited

In this section, we revisit the student planning scenarios and apply the plan evaluation metrics to the student planning activities. We also inspect the T5 and GPT-3.5 plans that were generated at these specific points of planning, compare the scores of these plans to the scores of the student plans, and discuss the implications of these LLM-generated plans as potential interventions in real-time. These plans are visualized in several figures. Goals for each plan are presented at the top in bold, and actions are the non-bolded elements connected to the goals. Figure 9.1 shows an interpretation of the plan visualization shown in the following sections. The colors presented in these figures represent the epistemic activity the action-goal pair map to, with solid and dashed borders indicating the cohesive value (fully or partially). Actions with a white background represent no epistemic activity mapping.

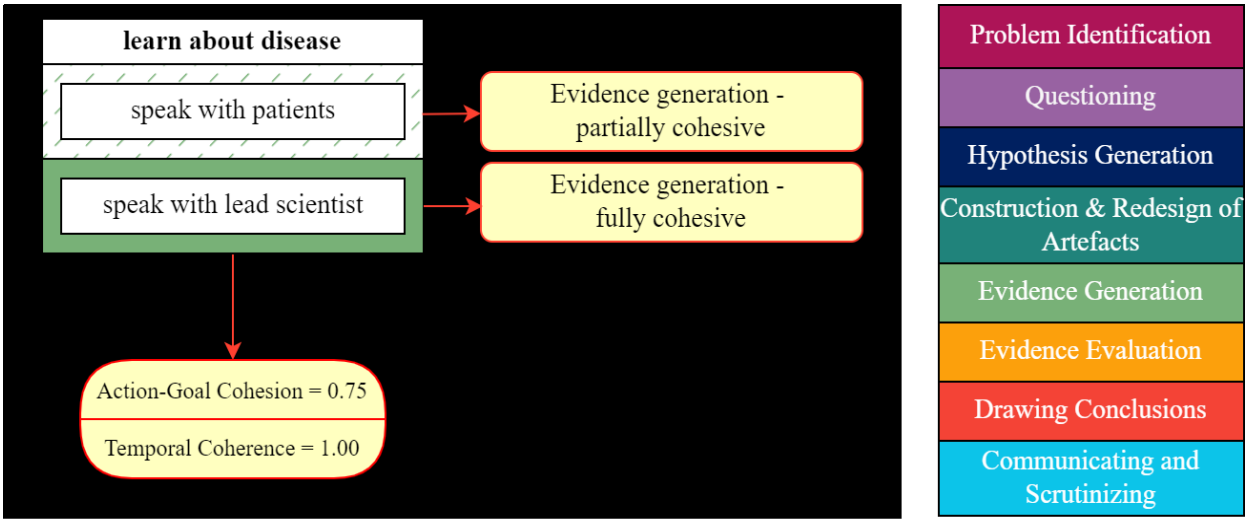

Figure 9.1 An example illustration of a planning instance with epistemic activity mapping and scoring (left), as well as the epistemic activity key (right).

9.1.1 Student A (Adam)

Scenario Description. In this first example scenario, we consider a student named Adam who accessed the planning tool five times throughout gameplay. In each of these planning tool interactions, Adam either created one plan or edited the existing plan. After completing the tutorial, they created a plan with the goal of "*learn about disease*" with two planned actions: "*speak with patients*" and "*speak with lead scientist*." Then, Adam immediately went to the infirmary and spoke with Kim and the patients. After a few minutes, Adam left the infirmary and wandered outside. They were then prompted to plan again because speaking with the camp nurse is a major plot point. During this planning instance Adam removed the action "*speak with patients*" from the plan and closed the tool. Immediately after this, they went to the lead scientist's house, read some informational text, and spoke with the lead scientist. Then, Adam began collecting items to test for viruses or bacteria. Next, Adam went to the laboratory and began testing the items he collected, receiving a positive scan result on an egg for pathogenic bacteria. Because this is another major plot point of the game, Adam was again prompted to plan. In this planning tool instance, they updated their plan to remove "*speak with lead scientist*" and added "*use scanner to test objects,*" while keeping the same goal. After this planning tool interaction, Adam kept scanning other objects for viruses or bacteria. Once leaving the laboratory, Adam was prompted yet again to open

the planning tool because they scanned a contaminated object. This time, Adam included "*read about how diseases spread*" in their plan and removed all other planned actions. Instead of following this plan, Adam then went to Kim and submitted an incorrect hypothesis for the cause of the illness and a treatment prevention plan. Soon after, at a 30-minute milestone of gameplay, Adam was prompted to plan, and they kept the same goal while changing the planned action back to "*use scanner to test objects*." Adam continued to test objects and interact with the game environment for another hour, but did not interact with the planning tool during that time. They successfully solved the mystery after 96 minutes of gameplay. Figure 9.2 shows Adam's full trajectory of planning tool instances.

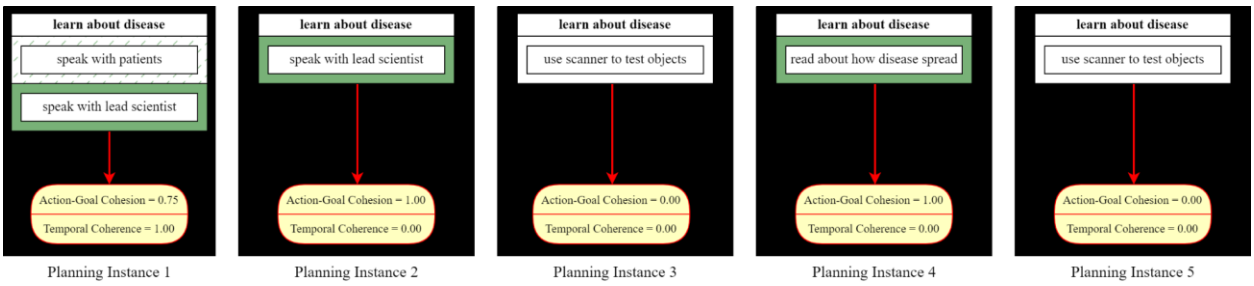

Figure 9.2 Adam's planning progression with corresponding scores.

LLM plan generation comparison. Adam had the same goal persist throughout planning tool instances. This emphasizes a limitation of the LLM-driven plan generation framework. Because the target goal is treated as input to the framework, the framework is only as good as the goal selected by the student. In other words, the plan generation framework is receiving multiple event sequences for the same goal. This is not ideal behavior; we would want the student to select a range of goals throughout gameplay. A potential improvement of this process in real time could be to first utilize the plan recognition framework to detect the goal the students are aiming to achieve, then provide that goal to the plan generation framework as input. Then support could be provided to the student by suggesting a different goal selection and a potential plan for that goal.

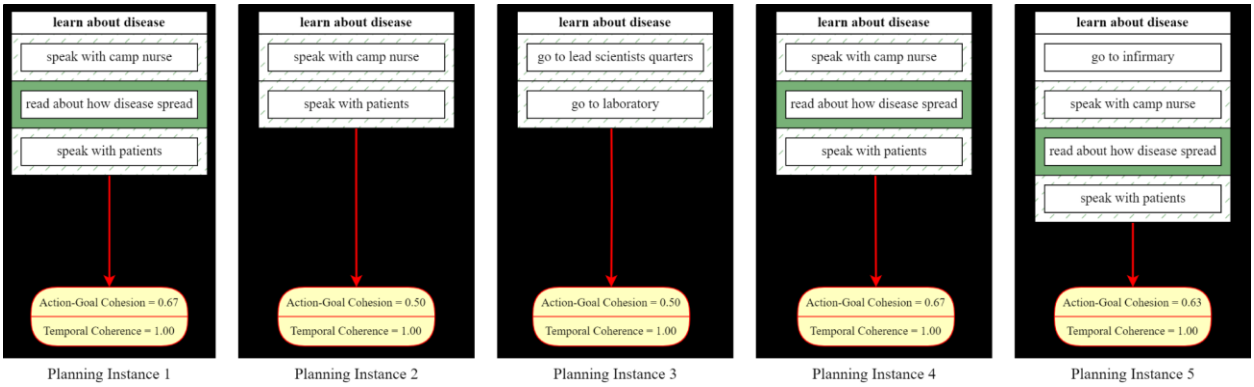

Figure 9.3 The T5-generated plans for Adam's planning progression with corresponding scores.

Despite having the same goal as input for each prediction point, both T5 and GPT-3.5 generated mostly unique plans across planning instances (Figures 9.3 and 9.4, respectively). T5 did have one duplicate plan (see "*learn about disease*", planning instance 1). We observed this plan present in T5-generated plans across students (see Barbara and Carl). This behavior likely represents the most probable plan for the target goal. Depending on the case for the student, it could be beneficial to provide the repeating plan if they are struggling to complete the goal. However, if a student is repeating the same behavior, this could mean that the student is frustrated, or if a support tool is repeatedly suggesting the same game events, it could become frustrating for the student. In this case, an adaptive system should first consider the actions a student has completed before making a suggestion.

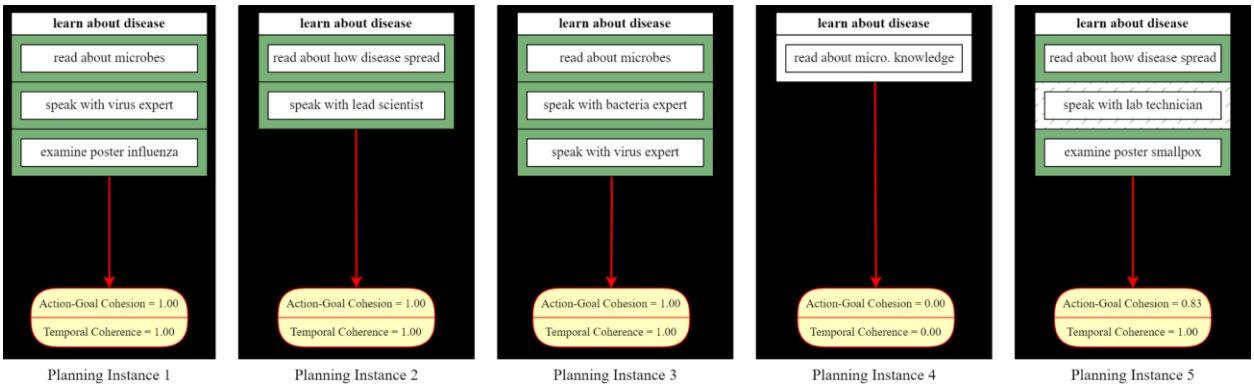

Figure 9.4 The GPT-3.5-generated plans for Adam's planning progression with corresponding scores.

Plans generated by GPT-3.5 scored highest in both action-goal cohesion and temporal coherence for four out of five plans, suggesting that GPT-3.5 may be optimal in this case. Interestingly, both T5 and GPT-3.5 generated plans that resulted in only the "*evidence generation*" epistemic activity. "*Evidence generation*" was the primary epistemic activity present in the CRYSTAL ISLAND planning matrix, so this finding aligns with the distribution of the matrix. These scenarios, however, highlight another gap in the temporal coherence scoring process. In the context of a given plan, rewarding planning behavior that stays within the same epistemic activity mapping makes sense because there is not a specific action limit for each epistemic activity. In other words, we do not have a precise understanding of how long a student should spend within one epistemic activity. However, it is not ideal for a student to stay in one epistemic activity for the entirety of gameplay and such behavior would potentially be an indicator that the student did not fully externalize his or her plans. Thus, another factor to consider in the temporal coherence plan evaluation metric is the progression of epistemic activities across gameplay, not just in the context of a single plan.

9.1.2 Student B (Barbara)

Scenario Description. Barbara also interacted with the planning tool five times throughout gameplay and did not successfully solve the mystery. They first planned after the tutorial, where they created one plan with the goal of "*explore camp*" and three planned actions: "*go to infirmary,*" "*speak with camp nurse,*" and "*speak with patients*." Immediately after creating this plan, Barbara went to the infirmary and completed all actions in the plan. During the next planning tool interaction, Barbara kept this plan in the tool despite having completed it. They also added another plan for the goal "*learn about outbreak*" with nine planned actions attached to this goal. In every subsequent planning tool interaction, these plans persisted. Barbara played the game for 71 minutes and never solved the mystery. Figure 9.5 shows Barbara's planning progression throughout gameplay. The dashed borders in this case indicate a partially cohesive mapping to "problem identification."

LLM plan generation comparison. Planning instances three through five were the same for Barbara. With the developed plan quality metrics, each of these planning instances scored the same because they are plan-level metrics. A metric is needed for planning behaviors across gameplay to better emphasize when a student is planning poorly. An intervention could be created for when a

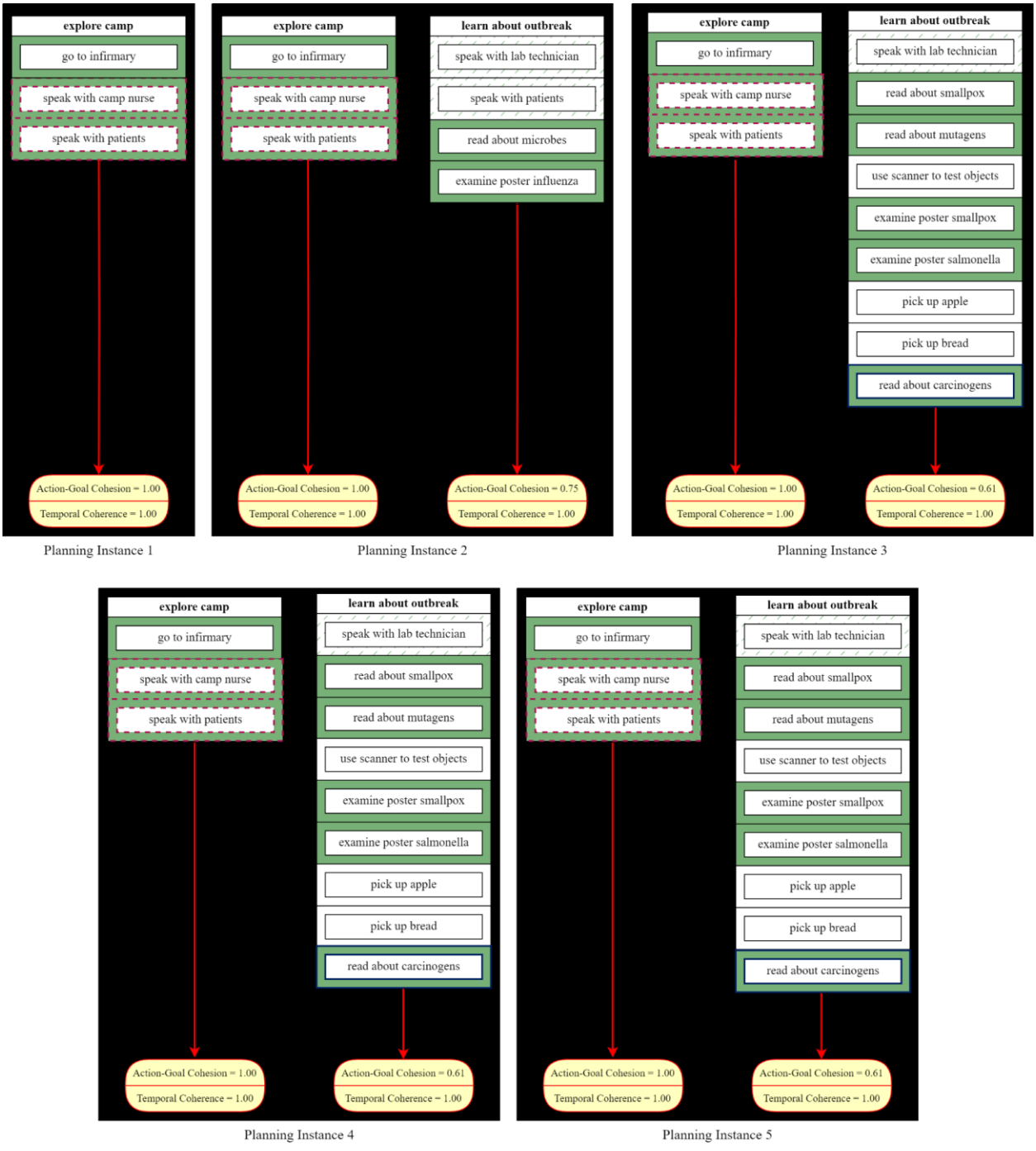

Figure 9.5 Barbara's planning progression with corresponding scores.

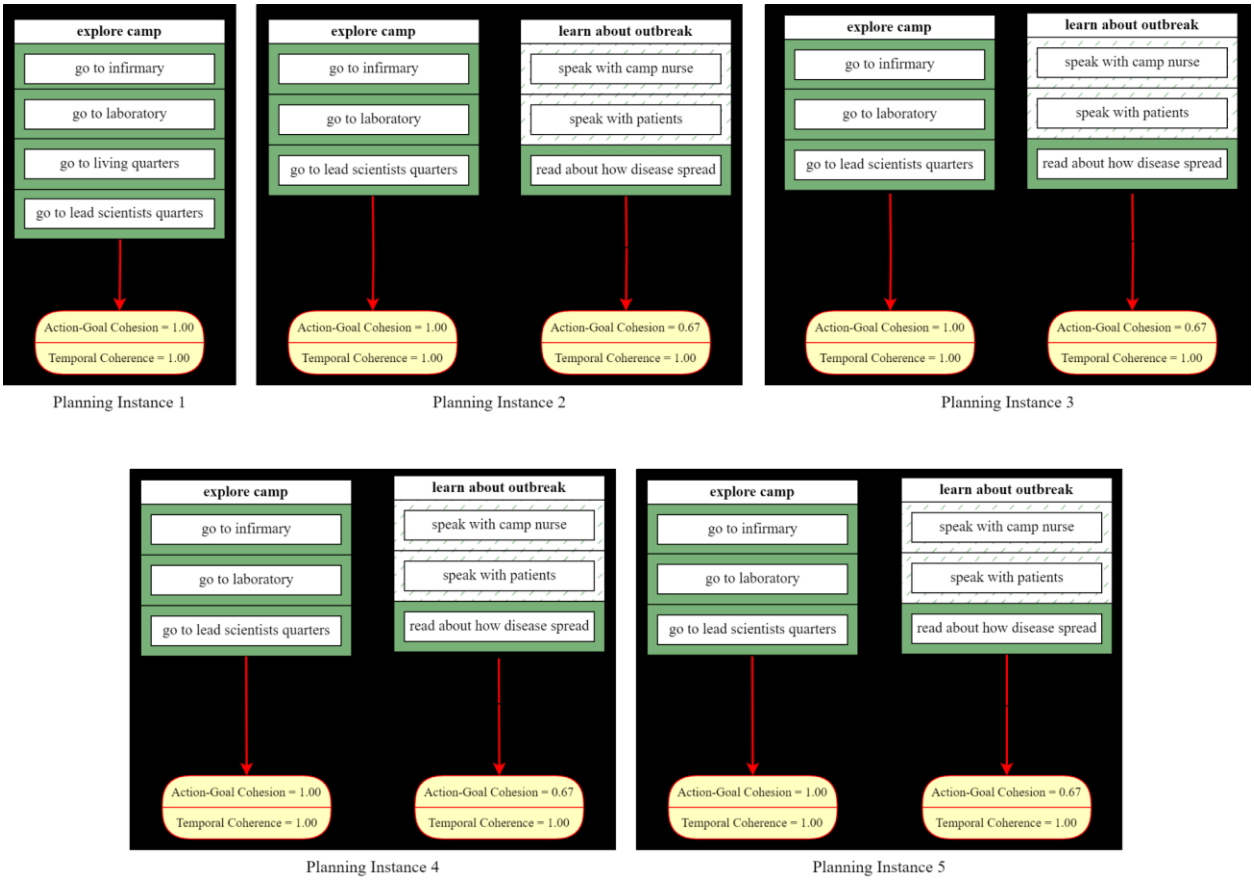

Figure 9.6 The T5-generated plans for Barbara's planning progression with corresponding scores.

repeated planning instance occurs with no changes that asks the student to reflect if they have accomplished any of the goals present in the planning tool instance.

There is also potential to monitor gameplay for when students are in specific epistemic activity phases. For example, "*explore camp*" and "*learn about outbreak*" are goals meant to be more exploratory and we would expect them to happen earlier in the game. If we monitored Barbara's gameplay in real-time to assess if they had started to test items, then we could suggest goals further along in the epistemic activity progression like "*evaluate hypothesis*" goals. The student plan recognition framework could be leveraged for this use case to help identify gaps in goals. This type of assistance has the potential to prevent Barbara from losing the game.

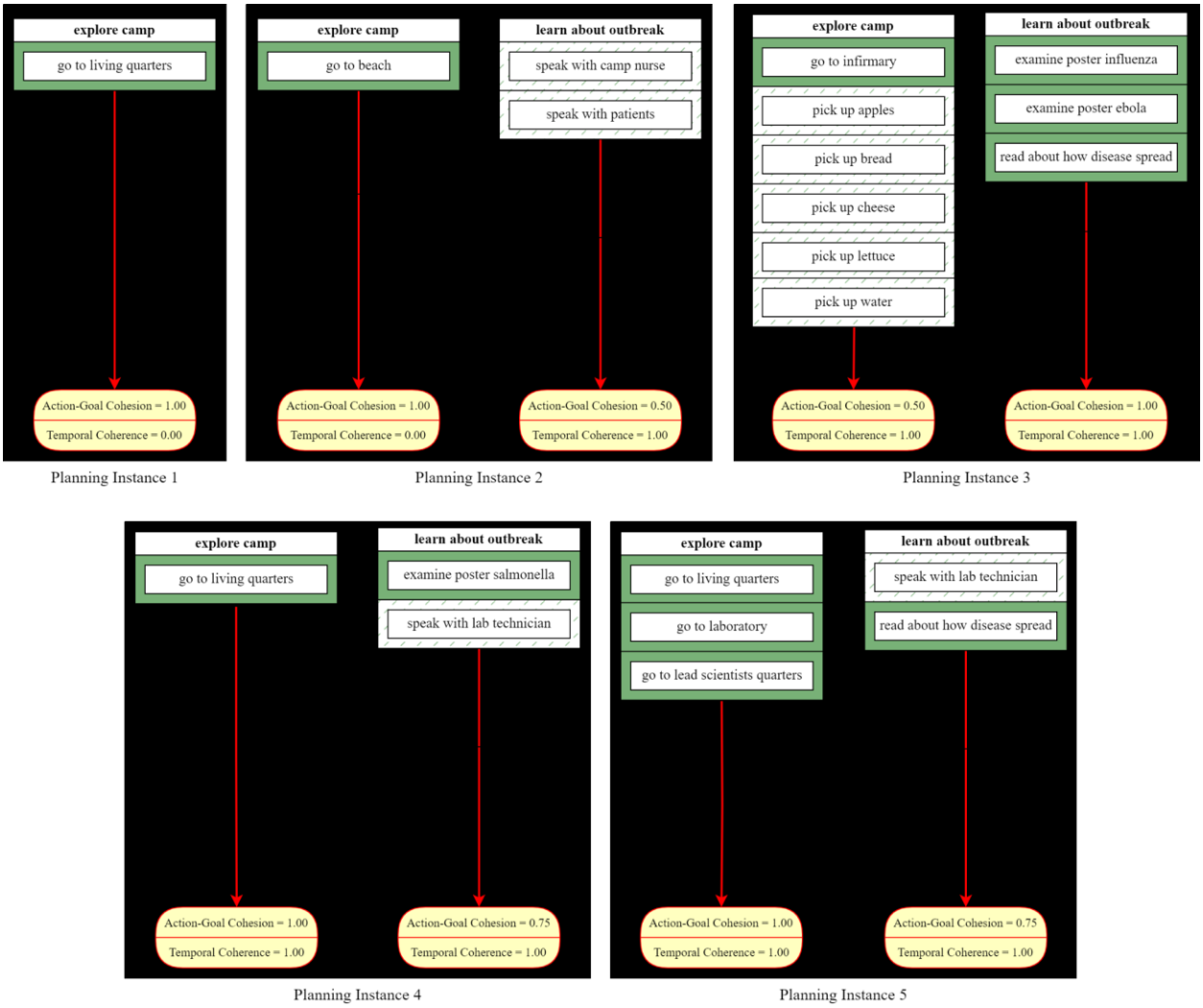

Figure 9.7 The GPT-3.5-generated plans for Barbara's planning progression with corresponding scores.

Both T5-generated plans and plans generated by GPT-3.5 outperformed student-constructed plans for action-goal cohesion, indicating these plans may be good alternatives to suggest in real time. However, T5 had duplicate plans for four out of five planning instances, indicating the model potentially overfit to these plans (Figure 9.6). An ideal intervention would have more variety in the plans being suggested to students. GPT-3.5 generated plans with only one action for four goals (Figure 9.7). The plans in this case would receive a score of zero for temporal coherence because they did not have a transition between epistemic activities. This highlights the need to emphasize longer plans as ideal planning behavior in the prompting.

9.1.3 Student C (Carl)

Scenario Description. For our last scenario, we consider a student named Carl. Like the other students, Carl planned five times during gameplay. Carl's first planning tool instance occurred after completing the tutorial and they created a plan with the goal of "*explore camp*" and five planned "*go to*" actions. After planning, Carl went to the infirmary and then the dormitory, completing two out of the five actions. They were then prompted again, where they kept their "*explore camp*" plan the same and added another plan with the goal "*learn about outbreak*" and two planned actions: "*use scanner to test objects*" and "*speak with lab technician*." Shortly after, Carl opened the planning tool again voluntarily and removed the "*explore camp*" plan.

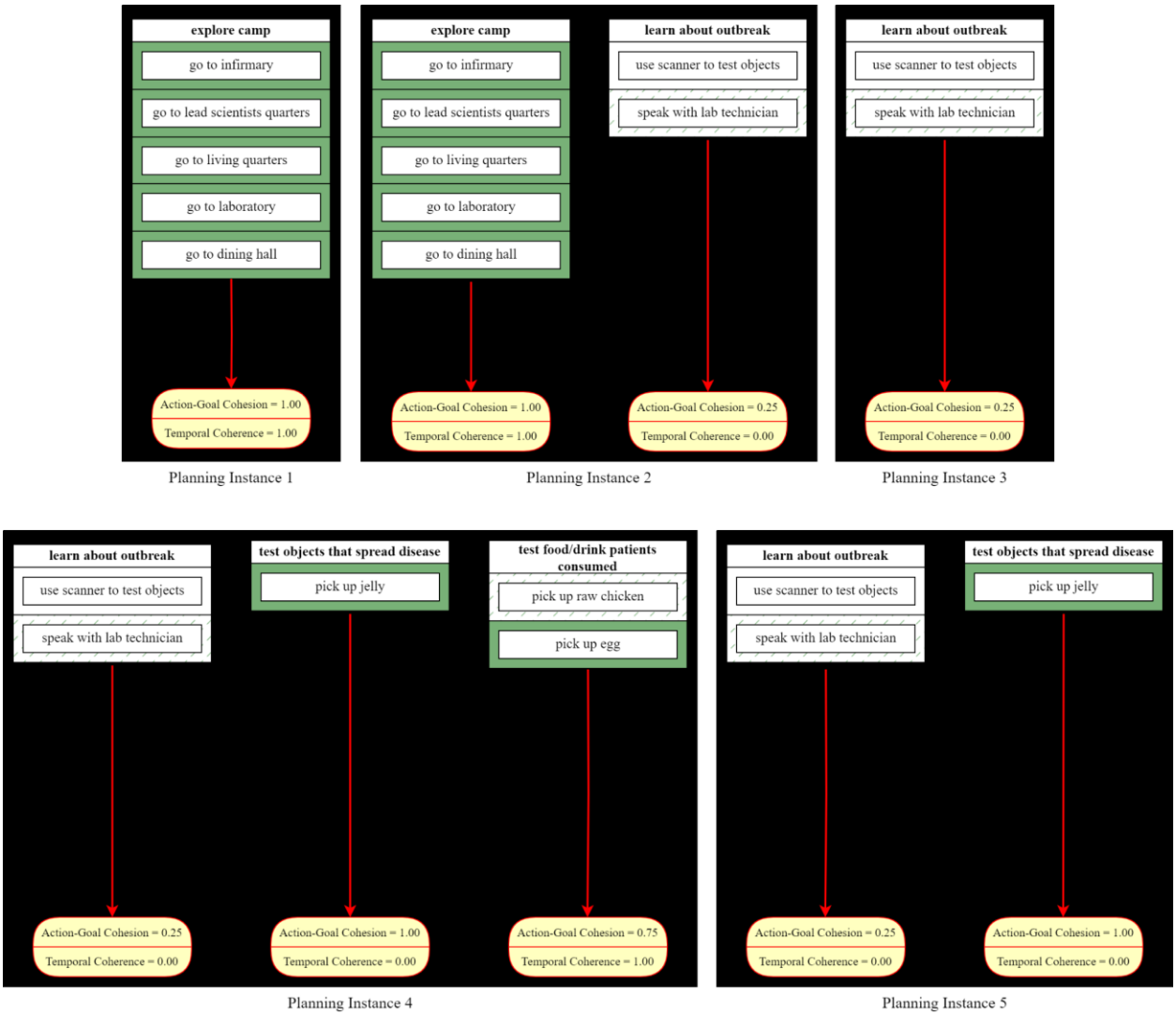

Figure 9.8 Carl's planning progression with corresponding scores.

After 30 minutes of gameplay, Carl was prompted to plan again. In this interaction, they added two plans with the goals "*test food/objects patients have consumed*" and "*test food/objects that can spread disease*." The first plan contained two "*pick up*" actions and the second plan contained one "*pick up*" action. Carl then proceeded to pick up several items from the dorm. Carl then opened the planning tool voluntarily and removed the "*test food/objects patients have consumed*" plan. Carl did not plan for the remainder of gameplay. They successfully solved the mystery and played in total for 73 minutes. Figure 9.8 shows Carl's plan progression.

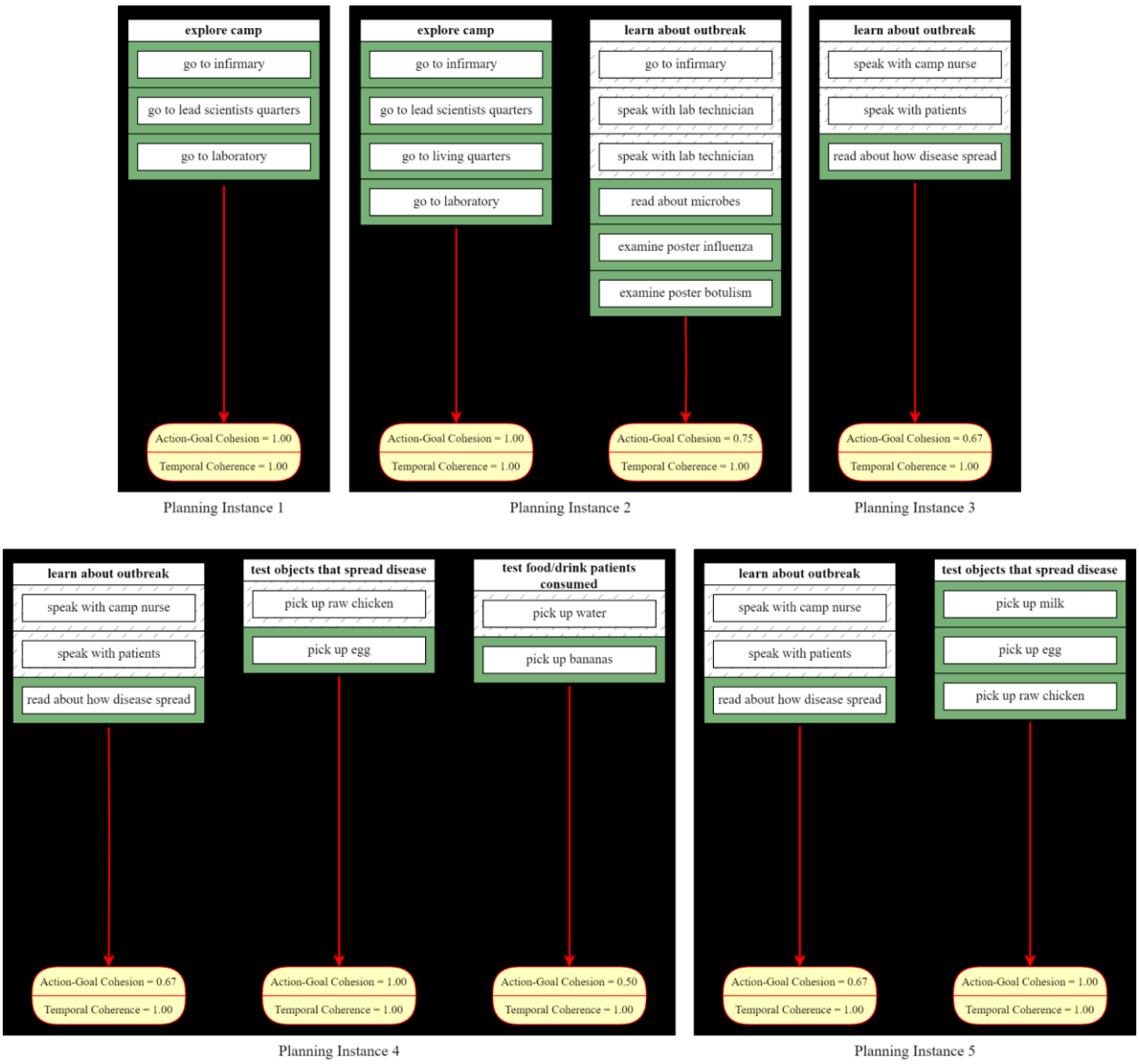

Figure 9.9 The T5-generated plans for Barbara's planning progression with corresponding scores.

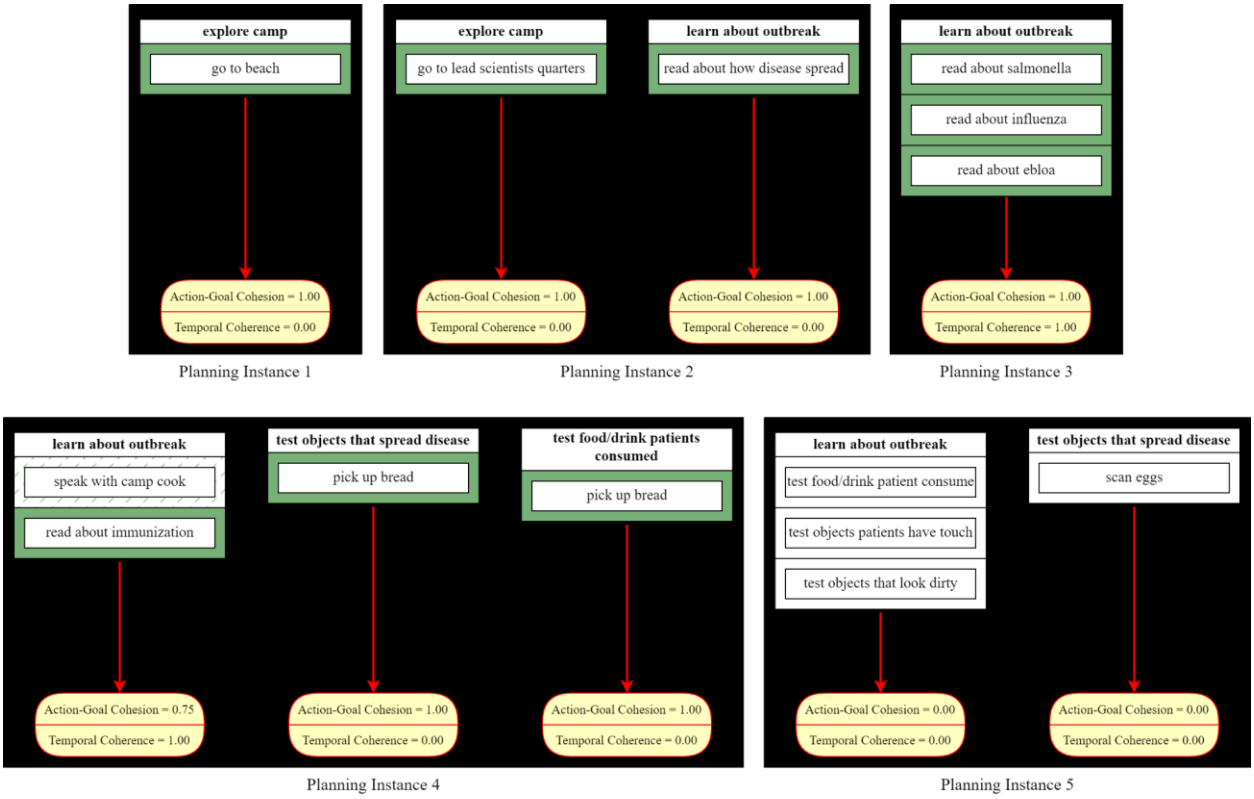

Figure 9.10 The GPT-3.5-generated plans for Carl's planning progression with corresponding scores.

LLM plan generation comparison. Carl demonstrated a change in goal selection throughout gameplay that aligns more closely to the progression experts would expect in CRYSTAL ISLAND. We still observed repeated plans and plans of length one, re-emphasizing limitations stated earlier (see Chapter 8, Section 8.5). T5-generated plans in this case were longer than student plans and had high temporal coherence scores, indicating that they included more detail and remained coherent (Figure 9.9). An adaptive system could utilize these plans to suggest actions to the student to lengthen the student plans and provide more guidance. This would likely improve student planning behavior and their navigation of the game.

In planning instance 5, plans generated by GPT-3.5 contained goals rather than planned actions for the goal "*learn about outbreak*" (Figure 9.10). These goals are still playable within the game but represent the potential for nested plans in CRYSTAL ISLAND, as we could include planned actions within the provided goals. They also demonstrate the potential to provide more high-level support to students in the goal selection process. Additionally, "*scan eggs*" was not an action

present in the planning tool but is a combination of the actions "*pick up eggs*" and "*use scanner to test objects.*" This summarization of the two actions shows a potential improvement for the planning tool to be more concise for students.

9.2. Theoretical Implications

Based on the findings from this dissertation and the key takeaways from the student planning scenarios, both the student plan recognition framework and the student plan generation framework have the potential to impact student learning and planning. These frameworks, as well as the plan evaluation metrics, also have the potential to impact the learning analytics community by improving our understanding, assessment, and assistance of student planning in science game-based learning environment. We will discuss the implications of this work on learning analytics and real-time SRL skill intervention in this section.

9.2.1 Implications for Learning Analytics

In our work, we highlight how using learning analytics (i.e., process-level data; Giannakos et al., 2022) provides much more detailed student learning data beyond examining aggregated, product-level data (e.g., test scores only). For our study, we combined process and product data; we examined student planning during gameplay, overall gameplay time, and whether they solved the mystery. This allowed us to examine student learning processes that occur during a task or series of tasks, which can inform the development of scaffolded support to be provided while students are working in real-time instead of as feedback after the fact. Without the analytics data, we would not be able to examine student learning processes.

Despite the contributions of adaptive scaffolding, we are still faced with a significant challenge related to the granularity between process and product data and the types of possible inferences possible between them (Azevedo, 2020). The data collected in our studies illustrate an imbalance in the amounts of data because process data are more voluminous than product data, and this dimension makes the automatic generation of adaptive scaffolds challenging for game-based learning environments. In addition, the learning analytics community needs to explore how the collection of real-time data regarding planning tools should be used to provide individualized adaptive scaffolds when planning and other self-regulated learning behaviors are so dynamic, unpredictable, and may follow various trajectories (Dever et al., 2023).

Our results revealed that students who did not plan spent significantly less time playing the game and were less likely to solve the mystery. This is contrary to a possible misconception that planning time might take away from gameplay time and thus, solving the mystery correctly. In fact, results revealed interacting with the planning tool is positively predictive of solving the mystery. However, our results also suggest there seems to be a limit to the amount of time it is useful to spend using the planning tool, as results also revealed spending too much time on the planning tool is negatively related to solving the mystery. These findings indicate the quantity vs. quality dichotomy related to planning and SRL, students' perceptions of the value of planning as a learning activity, and how internal (cognitive) representations of planning are represented and mapped onto external representations of planning (tools) such as those illustrated in Section 9.1. More specifically, students need to balance the amount and quality of time planning with the amount and quality of other SRL processes needed to learn complex instructional material. This balance requires a certain level of sophistication that not all young students possess (Hoyle & Dent, 2018).

From an SRL perspective, this suggests that although planning is an important component of self-regulation (Greene et al., 2024; Winne, 2018), planning is likely most effective when engaging in other SRL processes (i.e., metacognitive monitoring, effective strategy use, and self-reflection) as well. For instance, by examining if students used the planning tool, we are also examining if they are monitoring their plans by opening the planning tool during gameplay (i.e., after making their original plan). If they make changes to their plan, this indicates they are reflecting and making adaptations to their original plan. These inferences can be further supported by collecting multimodal data to understand when, what, how, and why students are monitoring and regulating (Azevedo & Gasevic, 2019). This highlights the interdependent nature of the learning phases, as outlined in the COPES model of SRL (Winne & Hadwin, 1998, 2008). In this study, we examined the use of the planning tool, but results suggest students are engaging in all SRL phases, and it would be difficult to single out planning from the other SRL processes.

Thus, although the tool allows students to plan their gameplay behavior, examining how students used the planning tool provides us with a rich source of information about all aspects of SRL, not just planning.

9.2.1 Implications for SRL Skill Intervention

Our results suggest that implementing the planning tool in CRYSTAL ISLAND allows students to generate and externalize their plans, which appears beneficial for their learning and gameplay (Goslen et al., 2024b). However, future studies are needed to understand the mechanisms of how this occurs. Perhaps the use of the planning tool fosters students' metacognitive awareness of their planning, based on discrepancies between prior knowledge, task demands, and learning outcomes. If this were the case, we would assume students have the procedural and conditional knowledge of what, when, why, and how to plan. While theories of SRL focus on the importance of planning (Pintrich, 2000; Winne & Azevedo, 2022; Zimmerman, 2013), there is indeed an implicit assumption that students know how to do so effectively and how to demonstrate this. Our student scenarios suggest that students might be capable of planning, but sometimes struggle externalizing this. In addition, whereas students can externalize their plans using the static planning tool, they may soon become overwhelmed as they monitor what they externalize in a static tool while enacting the plans in real-time during gameplay. This may lead to extraneous cognitive load (Wang & Lajoie, 2023). As SRL theory does not focus on what planning "looks like", we are unable to distinguish between being able to plan versus being able to externalize a plan. We interpret our results using a gameplay outcome (i.e., solving or not solving the mystery) as a proxy for planning. For example, we presume students who spent time using the planning tool were more likely to solve the mystery. However, we cannot confirm if we are differentiating between students who can plan or cannot plan, or if we are distinguishing between students who can or cannot externalize their plans (i.e., know how to plan, but cannot use the planning tool effectively). Future studies should focus on making that distinction because it will advance theoretical assumptions of SRL, and impact how we design adaptive scaffolds to improve students' planning and overall SRL abilities.

Furthermore, SRL theory focuses on the adaptive nature of SRL (Winne, 2018). Yet, it is unclear if students are aware of this. For example, our study shows some students returned to their plan during gameplay, but among those students, not everyone made changes to their plan. As we did not ask students why they returned to the planning tool, we cannot confirm if students who returned to the planning tool without making changes were reflecting on their plan and remained satisfied with it or if they wanted a reminder of their plan that (they erroneously assumed) they could not change after they first made it. Incorporating a prompt asking students why they chose

to reopen the planning tool or implementing a think-aloud protocol during gameplay could address this uncertainty, which is important because designing support would look different for these different actions. In the near future, we envision students explaining why they went back to the planning tool, whether they are considering changes they are considering, what types of changes, when the changes need to be implemented, etc. by engaging with NPCs using generative AI techniques (Johnson, 2023).

When designing adaptive support, it is crucial to make sure students understand the value of planning. If students do not see the value of planning, there is a small likelihood they will use any kind of tool. All theories of SRL stress the impact of motivation on SRL (Bong et al., 2023; Pintrich, 2000; Schunk & Zimmerman, 2012), yet it can be difficult to induce student motivation in the classroom because students have different knowledge, skills, and interests (Azevedo & Taub, 2020). Game-based learning has been shown to increase motivation (Nietfeld et al., 2023; Plass et al., 2015, 2020), however more support is needed to ensure students understand the value of using in-game tools, such as the planning tool, that focus on SRL. In our analysis of CRYSTAL ISLAND planning, we found that about half of the students showed limited use of the planning tool, possibly due to a lack of perceived value of it. Specifically, 63 students used the planning tool two times or less and 37 students did not use the tool at all. Thus, if we include system elements that inform students about how to plan, as well as the value of planning, this might increase the use of the planning tool. For example, we can implement an intelligent agent to formulate the plan with the student based on their prior knowledge (of microbiology and SRL). The system can help the student monitor their performance and foster reflection to reassess the plan and if they might want to make changes to it. This indicates a potential use of intelligent agents in games (Lester et al., 2020) to foster planning, SRL, and motivation for students.

Finally, our results demonstrate the use of the planning tool as a teaching and research tool (Azevedo et al., 2019). As researchers, implementing this tool and examining how students use it provides evidence of how we can identify planning through learning analytics. Although beyond the scope of this study, it is likely students demonstrate different ‘types’ of planning in the data, some of which lead to successful gameplay and some that do not. It will be difficult to identify and visualize these different types of planning because a plan is a multi-dimensional construct (i.e., inclusion of goals or sub-goals, hierarchy of goals, explicitness of goals, hypothetical goals, complexity of goals, sequence of goals, quality of goals, timing of forming a plan and when

students adjust plans could all be dimensions used to categorize plans). Therefore, future research will focus on determining the different types of plans found in the data and what the optimal visualizations of those data could be to induce, foster, and support students' planning. As a teaching tool, the planning tool itself can be used to teach the importance of planning in the classroom across subjects, within and beyond CRYSTAL ISLAND. In addition, teachers can access the data visualizations in a teacher dashboard (Bodily et al., 2018; Wiedbusch et al., 2021) to monitor students' planning and other SRL processes, which can impact their instructional decision making (see Taub & Azevedo, 2023).

Taking these results and interpretations into consideration, we believe including a planning tool within a science game-based learning environment has the potential to enhance students' self-regulatory skills by allowing them to externalize their plans during game-based learning. This also provides advancements for adaptive scaffolding in real-time based on learning analytics data captured by the system and visualized for teachers to implement this into the classroom.

CHAPTER 10

CONCLUSION

Self-regulatory skills are critical for students, particularly during scientific inquiry. Research demonstrates that students struggle to engage in effective self-regulatory goal-setting strategies during learning; however, when provided support from pedagogical agents, they exhibit greater learning outcomes than students without the support (Azevedo et al., 2019; Harley et al., 2018). Therefore, the development of a planning support tool provides students with opportunities to self-regulate their learning by allowing them to select which of the specific goal blocks they want to set as their subgoals for completing the game while also providing support for planning through the overarching strategies included in the planning tool. With recent advances in artificial intelligence, particularly with LLMs, there is significant potential to build systems to automatically predict and generate students' goals and planned actions as they interact with a digital learning environment and deliver real-time adaptive support for self-regulatory skills. This dissertation has reported on the investigation of the construction of a plan recognition framework that can accurately predict students' plans given a sequence of gameplay events. We have also examined various methods for developing a LLM-driven plan generation framework for real-time adaptive support for student learning and planning. Additionally, this work has contributed to the advancement of metrics for evaluating student planning behaviors during science learning. Both frameworks, as well as plan evaluation metrics, have been evaluated on two science game-based learning environments, demonstrating the generalizability of the frameworks across multiple science game-based learning environments. In combination with using plan quality metrics to assess model-generated plans, this work presented two frameworks that have the potential to enable adaptive support for students' SRL processes.

10.1 Hypotheses Revisited

This dissertation investigated the following thesis statement:

> *Plan recognition and plan generation frameworks leveraging classical machine learning techniques and large language models, respectively, can be used to automatically predict student plans and generate alternative plans to provide adaptive support and furthermore*

can be applied to multiple student populations and multiple science game-based learning environments.

The following hypotheses have been tested to evaluate the performance of both a student plan recognition framework and a plan generation framework with respect to the ability to predict and generate valid plans that can be enacted during gameplay. Machine learning models used for the plan recognition framework have been evaluated in terms of macro-average F-measure via 5- or 3- fold student-level cross-validation. LLMs for the student plan generation framework have been evaluated in terms of the percentage they match student plans and the plan quality metrics developed in this dissertation.

- H.1. Applying the plan recognition framework methods to the FUTURE WORLDS dataset will yield similar trends in results compared to evaluations of the framework using CRYSTAL ISLAND data.
    - H.1.1. Because of the smaller size of the FUTURE WORLDS dataset, the traditional machine learning methods will be lower in predictive performance but will still have LSTMs outperform other models.

      Among the two prediction tasks (multi-label goal and planned action sequence recognition), neither outperformed the CRYSTAL ISLAND *predictive performance in terms of macro-average F-measure.*

      For FUTURE WORLDS*, LSTMs had the highest predictive performance for goal recognition and third highest performance for planned action sequence recognition.*

      This hypothesis is partially accepted.

    - H.1.2. Incorporating both multi-task learning and automatic plan completion detection will improve predictive performance.

      Incorporating a multi-task architecture improved prediction performance for planned action sequence recognition, but not goal recognition.

      The combination of multi-task learning and automatic plan completion detection demonstrated performance improvement for both goal and planned action sequence recognition.

      This hypothesis is accepted.

- H.2. Large language models will demonstrate potential for generating student plans given a set of game actions, completed goals and a target goal for both learning environments despite small datasets.
    - H.2.1. Given that T5 uses a fully-supervised methodology, it will successfully generate plans similar to students' plans for both learning environments.

      For CRYSTAL ISLAND*, we observed that T5 had similar plan composition to student plans and identified that a quarter of T5-generated plans matched student plans exactly.*

      For FUTURE WORLDS*, T5-generated plans differed in plan composition compared to student plans, but still had about one quarter of generated plans exactly match student plans.*

      This hypothesis is partially accepted.

    - H.2.2. Using a few-shot learning approach, GPT-3.5 will generate valid plans that are more distinct from student plans for both learning environments.

      For CRYSTAL ISLAND*, we observed that plans generated with few-shot learning had distinct plan composition compared to student plans, with only* 7.26% of *generated plans matching student plans exactly.*

      For FUTURE WORLDS*, few-shot learning plans differed in plan composition compared to student plans, with about* 9.18% *of generated plans exactly matching student plans.*

      This hypothesis is accepted.

- H.3. The novel plan evaluation metrics that have been devised in collaboration with researchers in the learning sciences will provide further insight into the quality of both student and language model generated plans.
    - H.3.1. In terms of temporal coherence, students will achieve higher scores for their plans than the generated plans for both learning environments.

      Plans generated with T5 had the highest temporal coherence scores on average for both learning environments.

      This hypothesis is rejected.

- H.3.2. Action-goal cohesion performance will be similar for student plans and generated plans across learning environments.

  For CRYSTAL ISLAND*, we observed all sets of plans had the same median value of 1.0 and there was no significant difference in the average action-goal cohesion score across the plans.*

  For FUTURE WORLDS*, plans generated by GPT-3.5 had the same median value as student plans but had a significantly lower mean action-goal cohesion score ($p < 0.001$).*

  This hypothesis is partially accepted.

- H.3.3. Generated plans will be much longer than student plans, demonstrating more plan complexity.

  For CRYSTAL ISLAND*, T5-generated plans were significantly longer than student plans ($p < 0.001$), and plans generated by GPT-3.5 had no significant difference in plan complexity. This indicates that T5 plans had the highest plan complexity.*

  For FUTURE WORLDS*, plans generated by GPT-3.5 were longer than student plans ($p < 0.001$), and T5-generated plans had no significant difference in plan complexity. Plans generated by GPT-3.5 had the highest plan complexity based on these results.*

  This hypothesis is partially accepted.

- H.4. Applying the plan generation framework methods to the FUTURE WORLDS dataset will yield similar trends in results compared to the CRYSTAL ISLAND dataset.
  - H.4.1. Replicating plan generation methods in FUTURE WORLDS will produce similar results to CRYSTAL ISLAND in terms of T5 and GPT-3.5 performance.

    We found plans generated by T5 to be closely aligned with student plans and plans generated by GPT-3.5 to be more summative in nature in both learning environments. T5-generated plan performance in FUTURE WORLDS *had lower match percentage than* CRYSTAL ISLAND*, likely due to the small size of the dataset.*

    This hypothesis is partially accepted.

  - H.4.2. LLMs will enable transfer learning approaches for the task of plan generation between science learning environments.

Plans generated with transfer learning largely included planned actions not found in the planning tool, but were still viable because they represented higher-level strategies a student could take in FUTURE WORLDS.

This hypothesis is accepted.

10.2 Summary

This dissertation has described work that focuses on the key component of SRL, planning, which requires students to set goals and enact plans for achieving those goals. We have investigated plans crafted by students during gameplay utilizing an in-game planning support tool in a game-based learning environment called CRYSTAL ISLAND. With the ultimate goal of providing automated support for students' planning activities during game-based learning, we have created and improved a student plan recognition framework that leverages multi-task learning and automatic plan completion to predict students' goals and their subsequent planned action sequence to accomplish that goal. The dissertation built on these results by extending the student plan recognition framework to another science game-based learning environment, FUTURE WORLDS, finding similar trends in overall performance. These findings demonstrate the generalizability of the framework across science game-based learning environments.

Additionally, we have constructed a plan generation framework that leverages text-based representations of students' interactions in science game-based learning environments as input into LLMs to output a series of low-level actions that can be enacted to accomplish a goal in both learning environments. We examined both fully-supervised and few-shot learning techniques for the purpose of generating plans in real-time to support student learning and planning. We applied this methodology to both CRYSTAL ISLAND and FUTURE WORLDS corpora. For both learning environments, the plan generation framework demonstrated robust plans that were enactable in gameplay. Using a mixed-method approach, we found that generated plans generally aligned with student plans in terms of planned action matching and plan composition.

Finally, plan quality metrics have been developed through alignment with scientific reasoning theory. These generated plans were evaluated with the developed plan quality metrics to assess the performance of the LLMs in terms of scientific reasoning steps. Findings from both CRYSTAL ISLAND and FUTURE WORLDS applications demonstrate the viability of the plan generation framework providing planning support to students in real-time, with plans largely

aligning with scientific reasoning steps. The dissertation presents an in-depth discussion of the implications of utilizing these plans and the developed metrics for real-time support, examining three scenarios of student gameplay within CRYSTAL ISLAND. When taken together, these analyses have the potential to enable adaptive scaffolding for *what* students plan to do next and *how* they plan to enact their goals. The dissertation has advanced research on SRL analytics by demonstrating methods for adaptive support and assessing planning behaviors that generalize across science game-based learning environments.

10.3 Future Work

The findings from this dissertation point towards several areas for future research. First, the current plan recognition framework leverages student planning activities as labels to evaluate performance. While this is a novel contribution of the work, it limits the real-time applicability of the framework since the prediction point comes after a plan is constructed by a student. Applying the framework to other online learning environments where student planning is not as explicit would provide more insight into the potential of the framework in real time. In such settings, we could pre-train the framework on the existing CRYSTAL ISLAND and FUTURE WORLDS corpora, provide real-time game events as input to the model, and evaluate the model output by potentially asking students to verify if their current goal or plan is what the model predicted.

Second, we could extend the plan generation framework by exploring the performance of more recent releases of ChatGPT, like GPT-4, GPT-4o, and GPT-o1, or open-source models, like Llama. This investigation could prove useful to understanding the potential benefit of these models in educational settings. Additionally, exploring how natural language representations of the input data impact performance is a promising area for future work. For example, constructing sentences of gameplay that resemble more of a narrative may positively impact the framework performance, as it may provide more context to the language models.

Third, further exploration of transfer learning with LLMs would prove beneficial for understanding the capabilities of LLMs. In educational settings, available data is often limited, which in turn, limits the robustness of machine learning models for predicting learning behaviors. LLMs provide new possibilities for learning environments because they are pre-trained on large datasets. Examining LLMs' ability to generate plans across learning environments using fully-supervised or few-shot learning approaches would be an interesting area of future work.

Fourth, additional work is needed to improve the plan evaluation metrics to assess plans constructed by students as well as model-generated plans. Incorporating both sequentiality of goals and other high-level strategy metrics, like achievement goal orientation, into the models may help improve real-time adaptive support and has the potential to scaffold planning. Exploring how such methods could be applied to generating other types of SRL scaffolding or how such methods could enhance current tools would provide insight into the generalizability of these techniques. For example, reflection is a necessary SRL skill that is closely related to goal setting and planning. A potential planning scaffold could be to generate several plans and ask the student to reflect on which may work best. Then, a student could construct a plan on their own and be prompted to reflect on which plan worked best for them.

Finally, it will be important to implement both frameworks, as well as the plan quality metrics, into learning environments and deploy them in classrooms. Obtaining student feedback on the planning tools and real-time support provided from the frameworks is paramount in understanding the impact such tools can have on student learning. Providing feedback on students' plans in real-time, as well as assisting them in constructing plans, offers the potential to create more effective learning experiences for students.

REFERENCES

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., … Zoph, B. (2024a). GPT-4 Technical Report (arXiv:2303.08774). arXiv. http://arxiv.org/abs/2303.08774

Azevedo, R., & Gašević, D. (2019). Analyzing multimodal multichannel data about self-regulated learning with advanced learning technologies: Issues and challenges. *Computers in Human Behavior*, 96, 207–210. https://doi.org/10.1016/j.chb.2019.03.025

Azevedo, R., Mudrick, N. V., Taub, M., & Bradbury, A. E. (2019). Self-regulation in computer-assisted learning systems. In J. Dunlosky & K. A. Rawson (Eds.), *The Cambridge handbook of cognition and education (pp. 587–618). Cambridge University Press*. https://doi.org/10.1017/9781108235631.024

Azevedo, R., Mudrick, N., Taub, M., & Wortha, F. (2017). Coupling between metacognition and emotions during STEM learning with advanced learning technologies: A critical analysis, implications for future research, and design of learning systems. *Teachers College Record, 119*(13).

Azevedo, R., & Taub, M. (2020). The challenge of measuring processes and outcomes while learning from multiple representations with advanced learning technologies. In P. Van Meter, A. List, D. Lombardi & P. Kendeou (Eds.), *Handbook of learning from multiple representations and perspectives* (pp. 532–553). Routledge. https://doi.org/10.4324/9780429443961

Azevedo, R., & Wiedbusch, M. (2023). Theories of metacognition and pedagogy applied in AIED systems. *Handbook of Artificial Intelligence in Education*, 45-67.

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*(1), 99–109.

Ben-Eliyahu, A., & Bernacki, M. L. (2015). Addressing complexities in self-regulated learning: A focus on contextual factors, contingencies, and dynamic relations. *Metacognition and Learning*, 10, 1-13.

Bernacki, M. L. (2017). Examining the cyclical, loosely sequenced, and contingent features of self-regulated learning: Trace data and their analysis. In *Handbook of self-regulation of learning and performance* (pp. 370-387). Routledge.

Bercher, P., Alford, R., & Höller, D. (2019). A Survey on Hierarchical Planning-One Abstract Idea, Many Concrete Realizations. *IJCAI*, 6267–6275.

Bisson, F., Larochelle, H., & Kabanza, F. (2015, June). Using a recursive neural network to learn an agent's decision model for plan recognition. In *Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, 918–924.

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

Bodily, R., Kay, J., Aleven, V., Jivet, I., Davis, D., Xhakaj, F., & Verbert, K. (2018). Open learner models and learning analytics dashboards: A systematic review. In S. B. Shum, R. Ferguson, A. Merceron, & X. Ochoa (Eds.), *Proceedings of the 8th international conference on learning analytics and knowledge* (pp. 41–50). ACM. https://doi.org/10.1145/3170358.3170409

Boeda, G. (2021). Multi-Agent Cooperation in Games with Goal Oriented Action Planner: Use Case in WONDER Prototype Project. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, *17*(1), 204–207.

Boekaerts, M., & Pekrun, R. (2015). Emotions and emotion regulation in academic settings. *Handbook of educational psychology*, 3, 76-90.

Bong, M., Reeve, J., & Kim, S. I. (Eds.). (2023). *Motivation science: Controversies and insights*. Oxford University Press.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., & others. (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. *International Conference on Artificial Intelligence in Education*, 327–339.

Carpenter, D., Cloude, E., Rowe, J., Azevedo, R., & Lester, J. (2021, April). Investigating student reflection during game-based learning in middle grades science. In *LAK21: 11th International Learning Analytics and Knowledge Conference* (pp. 280–291).

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. D. O., Kaplan, J., ... & Zaremba, W. (2021). Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

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. *International Conference on Artificial Intelligence in Education*, 217–228.

Cogliano, M., Bernacki, M. L., Hilpert, J. C., & Strong, C. L. (2022). A self regulated learning analytics prediction-and-intervention design: Detecting and supporting struggling biology students. *Journal of Educational Psychology*, *114*(8), 1801–1816. https://doi.org/10.1037/edu0000745

Cloude, E., Carpenter, D., Dever, D., Azevedo, R., & Lester, J. (2021). Game-based learning analytics for supporting adolescents' reflection. *Journal of Learning Analytics*, 8(2), 51–72. https://doi.org/10.18608/jla .2021.7371

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 S. Isotani, E. Millán, A. Ogan, P. Hastings, B. McLaren, & R. Luckin (Eds.), Artificial Intelligence in Education (Vol. 11626, pp. 36–40). Springer International Publishing. https://doi.org/10.1007/978-3-030-23207-8_7

Conati, C., Gertner, A., & Vanlehn, K. (2002). Using Bayesian networks to manage uncertainty in student modeling. *User Modeling and User-Adapted Interaction*, *12*(4), 371-417.

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.

Dever, D. A., Sonnenfeld, N. A., Wiedbusch, M. D., Schmorrow, S. G., Amon, M. J., & Azevedo, R. (2023a). A complex systems approach to analyzing pedagogical agents' scaffolding of self-regulated learning within an intelligent tutoring system. *Metacognition and Learning*, 1-33.

Dever, D. A., Wiedbusch, M. D., Romero, S. M., Smith, K., Patel, M., Sonnenfeld, N., ... & Azevedo, R. (2023b). Identifying the effects of scaffolding on learners' temporal deployment of self-regulated learning operations during game-based learning using multimodal data. *Frontiers in Psychology*, *14*.

D'Mello, S. K., & Graesser, A. (2023). Intelligent tutoring systems: How computers achieve learning gains that rival human tutors. In *Handbook of educational psychology* (pp. 603-629). Routledge.

Duarte, F. F., Lau, N., Pereira, A., & Reis, L. P. (2020). A survey of planning and learning in games. *Applied Sciences*, *10*(13), 4529.

Elliot, A. J., & McGregor, H. A. (2001). A 2× 2 achievement goal framework. *Journal of personality and social psychology*, *80*(3), 501.

Fischer, F., Kollar, I., Ufer, S., Sodian, B., Hussmann, H., Pekrun, R., ... & Eberle, J. (2014). Scientific reasoning and argumentation: advancing an interdisciplinary research agenda in education. *Frontline Learning Research*, *2*(3), 28-45.

Geden, M., Emerson, A., Carpenter, D., Rowe, J., Azevedo, R., & Lester, J. (2021). Predictive student modeling in game-based learning environments with word embedding representations of reflection. *International Journal of Artificial Intelligence in Education*, *31*(1), 1–23. https://doi.org/10 .1007/s40593-020-00220-4

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

Ghallab, M., Nau, D., & Traverso, P. (2016). *Automated planning and acting*. Cambridge University Press.

Giannakos, M., Spikol, D., Di Mitri, D., Sharma, K., Ochoa, X., & Hammad, R. (2022). Introduction to multimodal learning analytics. In M. Giannakos, D. Spikol, D. Di Mitri, K. Sharma, X. Ochoa & R. Hammad (Eds.), *The multimodal learning analytics handbook* (pp. 3–28). Springer International Publishing.

Goslen, A., Carpenter, D., Rowe, J., Henderson, N., Azevedo, R., & Lester, J. (2022a, January). Leveraging Student Goal Setting for Real-Time Plan Recognition in Game-Based Learning. In Proceedings of the 23rd International Conference on Artificial Intelligence in Education, (pp. 78-89).

Goslen, A., Carpenter, D., Rowe, J., Azevedo, R., & Lester, J. (2022b, October). Robust Player Plan Recognition in Digital Games with Multi-Task Multi-Label Learning. In Proceedings of the Eighteenth Annual AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, (pp. 105-112).

Goslen, A., Kim, Y. J., Rowe, J., & Lester, J. (2024a, July). LLM-Based Student Plan Generation for Adaptive Scaffolding in Game-Based Learning Environments. *International Journal of Artificial Intelligence in Education*. https://doi.org/10.1007/s40593-024-00421-1

Goslen, A., Taub, M., Carpenter, D., Azevedo, R., Rowe, J., & Lester, J. (2024b, September). Leveraging student planning in game-based learning environments for self-regulated learning analytics. Journal of Educational Psychology. https://psycnet.apa.org/record/2025-19911-001

Graesser, A., & McNamara, D. (2010). Self-regulated learning in learning environments with pedagogical agents that interact in natural language. *Educational Psychologist*, *45*(4), 234-244.

Greene, J. A., Anderson, J. L., O'Malley, C. E., & Lobczowksi, N. G. (2018). Fostering self-regulated science inquiry in physical sciences. *Connecting self-regulated learning and performance with instruction across high school content areas*, 163-183.

Greene, J. A., Bernacki, M. L., & Hadwin, A. F. (2024). Self-regulation. *Handbook of educational psychology*, 314-334.

Gupta, A., Carpenter, D., Min, W., Rowe, J., Azevedo, R., & Lester, J. (2022, October). Enhancing multimodal goal recognition in open-world games with natural language player reflections. In Proceedings of the AAAI conference on artificial intelligence and interactive digital entertainment (Vol. 18, No. 1, pp. 37–44).

Harley, J. M., Taub, M., Azevedo, R., & Bouchet, F. (2017). Let's set up some subgoals: Understanding human-pedagogical agent collaborations and their implications for learning and prompt and feedback compliance. *IEEE Transactions on Learning Technologies*, *11*(1), 54-66.

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

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

Hoffmann, J., & Nebel, B. (2001). The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, *14*, 253-302.

Hooshyar, D., Yousefi, M., & Lim, H. (2019). A systematic review of data-driven approaches in player modeling of educational games. *Artificial Intelligence Review*, *52*(3), 1997–2017. https://doi.org/10.1007/s10462-017-9609-8

HuggingFace. T5-small. https://huggingface.co/docs/transformers/model_doc/t5

Janssen, E., Depaepe, F., Claes, E., & Elen, J. (2019). Fostering students' scientific reasoning skills in secondary education: An intervention study. *International Journal of Science, Mathematics and Technology Learning*, *26*(1), 1-19.

Jiao, Y., Shridhar, K., Cui, P., Zhou, W., & Sachan, M. (2023). Automatic Educational Question Generation with Difficulty Level Controls. *International Conference on Artificial Intelligence in Education*, 476–488.

Johnson, W. L. (2023). How to harness generative AI to accelerate human learning. *International Journal of Artificial Intelligence in Education*. Advance online publication. https://doi.org/10.1007/s40593-023-00367-w

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

Klahr, D., & Dunbar, K. (1988). Dual space search during scientific reasoning. *Cognitive science*, *12*(1), 1-48.

Knogler, M., Harackiewicz, J. M., Gegenfurtner, A., & Lewalter, D. (2015). How situational is situational interest? Investigating the longitudinal structure of situational interest. *Contemporary Educational Psychology, 43,* 39-50.

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*.

Lester, J. C., Ha, E. Y., Lee, S. Y., Mott, B. W., Rowe, J. P., & Sabourin, J. L. (2013). Serious games get smart: Intelligent game-based learning environments. *AI Magazine, 34*(4), 31-45.

Lester, J. C., Spain, R. D., Rowe, J. P., & Mott, B. W. (2020). Instructional support, feedback, and coaching in game-based learning. In J. L. Plass, R. E. Mayer, & B. D. Homer (Eds.), *Handbook of game-based learning* (pp. 209–237). MIT Press

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

Levy, O., & Goldberg, Y. (2014, June). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2, 302-308.

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. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

Liu, S. M., & Chen, J. H. (2015). A multi-label classification based approach for sentiment classification. *Expert Systems with Applications, 42*(3), 1083–1093. https://doi.org/10.1016/j.eswa.2014.08.036

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

Madjarov, G., Kocev, D., Gjorgjevikj, D., & Džeroski, S. (2012). An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition, 45*(9), 3084–3104. https://doi.org/10.1016/j.patcog.2012 .03.004

McCardle, L., Webster, E. A., Haffey, A., & Hadwin, A. F. (2017). Examining students' self-set goals for self-regulated learning: Goal properties and patterns. *Studies in Higher Education, 42*(11), 2153-2169.

McLaren, B. M., & Nguyen, H. A. (2023). 20. Digital learning games in artificial intelligence in education (AIED): A review. In B. du Boulay, A. Mitrovic & K. Yacef (Eds.), Handbook of artificial intelligence in education (pp. 440–484). Edward Elgar Publishing.

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. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2590-2596, New York, New York.

Min, W., Mott, B., Rowe, J., Taylor, R., Wiebe, E., Boyer, K. E., & Lester, J. (2017, September). Multimodal goal recognition in open-world digital games. In *Thirteenth Artificial Intelligence and Interactive Digital Entertainment Conference*, 80–86.

Nietfeld, J. L., Sperling, R. A., & Young, T. M. (2023). More than just fun and games: The role of games in postsecondary education to support self-regulated learning. *New Directions for Teaching and Learning*, 2023(174), 41–47. https://doi.org/10.1002/tl.20547

OECD. (2022). *Trends shaping education 2022.* OECD.

Omarchevska, Y., Lachner, A., Richter, J., & Scheiter, K. (2022). It takes two to tango: How scientific reasoning and self-regulation processes impact argumentation quality. *Journal of the Learning Sciences, 31*(2), 237-277.

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

Pereira, R. B., Plastino, A., Zadrozny, B., & Merschmann, L. H. (2018). Correlation analysis of performance measures for multi-label classification. *Information Processing & Management, 54*(3), 359–369. https:// doi.org/10.1016/j.ipm.2018.01.002

Pintrich, P. R. (2000). The role of goal orientation in self-regulated learning. In M. Boekaerts, P. R. Pintrich, & M. Zeidner (Eds.), *Handbook of self-regulation* (pp. 451–502). Academic Press. https://doi.org/10.1016/B978-012109890-2/50043-3.

Plass, J. L., Homer, B. D., & Kinzer, C. K. (2015). Foundations of game-based learning. Educational Psychologist, 50(4), 258–283. https://doi.org/ 10.1080/00461520.2015.1122533

Plass, J. L., Mayer, R. E., & Homer, B. D. (Eds.). (2020). *Handbook of game-based learning*. MIT Press.

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

Rabkina, I., Kantharaju, P., Wilson, J. R., Roberts, M., & Hiatt, L. M. (2022). Evaluation of Goal Recognition Systems on Unreliable Data and Uninspectable Agents. *Frontiers in Artificial Intelligence*, 4, 734521. https://doi.org/10.3389/frai.2021.734521

Rabkina, I., Kantharaju, P., Wilson, J. R., Roberts, M., & Hiatt, L. M. (2021). Comparing Hierarchical Goal Recognition via HTN, CCG, and Analogy. *Proceedings of the AAAI 2021 Workshop on Plan, Activity, and Intent Recognition*.

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.

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098.

Schunk, D. H., & Zimmerman, B. J. (Eds.). (2012). *Motivation and self-regulated learning: Theory, research, and applications.* Routledge.

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.

Sinatra, G. M., & Taasoobshirazi, G. (2017). The self-regulation of learning and conceptual change in science: Research, theory, and educational applications. In *Handbook of self-regulation of learning and performance* (pp. 153-165). Routledge.

Shabrina, P., Mostafavi, B., Chi, M., & Barnes, T. (2023, June). 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). Cham: Springer Nature Switzerland.

Shergadwala, M. N., Teng, Z., & El-Nasr, M. S. (2021). Can We Infer Player Behavior Tendencies from a Player's Decision-Making Data? Integrating Theory of Mind to Player Modeling. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, *17*(1), 195–202.

Squire, K. (2008). Open-Ended Video Games: A Model for Developing Learning for the Interactive Age. *The Ecology of Games: Connecting Youth, Games, and Learning*, 167–198.

Srinivasa-Desikan, B. (2018). *Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras*. Packt Publishing Ltd.

Sukthankar, G., Geib, C., Bui, H., Pynadath, D., & Goldman, R. P. (Eds.). (2014). *Plan, activity, and intent recognition: Theory and practice*. Newnes.

Sutskever, I. (2014). Sequence to Sequence Learning with Neural Networks. arXiv Preprint arXiv:1409.3215.

Taub, M., & Azevedo, R. (2019). How does prior knowledge influence eye fixations and sequences of cognitive and metacognitive SRL processes during learning with an intelligent tutoring system?. *International Journal of Artificial Intelligence in Education*, *29*, 1-28.

Taub, M., & Azevedo, R. (2023). Teachers as self-regulated learners: The role of multimodal data analytics for instructional decision making. *New Directions for Teaching and Learning, 2023*(174), 25–32. https:// doi.org/10.1002/tl.20545

Taub, M., Sawyer, R., Smith, A., Rowe, J., Azevedo, R., & Lester, J. (2020). The agency effect: The impact of student agency on learning, emotions, and problem-solving behaviors in a game-based learning environment. *Computers & Education*, 147, 103781.

Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H. T., ... & Le, Q. (2022). Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., … Scialom, T. (2023). Llama 2: Open Foundation and Fine-Tuned Chat Models (arXiv:2307.09288). arXiv. http://arxiv.org/abs/2307.09288

Van Mil, M. H. W., Postma, P. A., Boerwinkel, D. J., Klaassen, K., Waarlo, A. J.: Molecular mechanistic reasoning: Toward bridging the gap between the molecular and cellular levels in life science education. *Science Education, 100*(3), 517–585 (2016)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Venkatesh, V., & Davis, F. D. (2000). A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management science*, *46*(2), 186-204.

Wallace, C. S., & Coffey, D. J. (2019). Investigating elementary preservice teachers' designs for integrated science/literacy instruction highlighting similar cognitive processes. *Journal of Science Teacher Education*, 30(5), 507-527.

Wang, T., & Lajoie, S. P. (2023). How does cognitive load interact with self-regulated learning? A dynamic and integrative model. *Educational Psychology Review, 35*(3), Article 69. https://doi.org/10.1007/s10648- 023-09794-6

Ware, S. G., & Siler, C. (2021). Sabre: A Narrative Planner Supporting Intention and Deep Theory of Mind. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, *17*(1), 99–106.

Weerawardhana, S., Whitley, D., & Roberts, M. (2021). Domain-independent Plan Intervention. *Proceedings of the AAAI 2021 Workshop on Plan, Activity, and Intent Recognition*.

Weintrop, D., & Wilensky, U. (2019). Transitioning from introductory block-based and text-based environments to professional programming languages in high school computer science classrooms. *Computers & Education*, 142, 103646.

Wiedbusch, M. D., Dever, D. A., Goslen, A., Carpenter, D., Marano, C., Smith, K., & Azevedo, R. (2024). Contextualizing Plans: Aligning Students Goals and Plans during Game-Based Inquiry Science Learning. To appear in Proceedings of 26th International Conference on Human-Computer Interaction.

Wiedbusch, M., Kite, V., Yang, X., Park, S., Chi, M., Taub, M., & Azevedo, R. (2021). A theoretical and evidence-based conceptual design of MetaDash: An intelligent teacher dashboard to support teachers' decision making and students' self-regulated learning. *Frontiers in Education, 6*, Article 570229. https://doi.org/10.3389/feduc.2021.570229

Winne, P. (2018). Cognition and metacognition within self-regulated learning. In D. Schunk & J. Greene (Eds.), *Handbook of self-regulation of learning and performance* (2nd ed., pp. 36–48). Routledge.

Winne, P., & Azevedo, R. (2022). Metacognition and self-regulated learning. In R. Sawyer (Ed.), *The Cambridge handbook of the learning sciences* (3rd ed., pp. 93-113). Cambridge University Press. https://doi.org/10.1017/9781108888295.007

Winne, P., & Hadwin, A. (1998). Studying as self-regulated learning. In D. J. Hacker, J. Dunlosky, & A. C. Graesser (Eds.), *Metacognition in educational theory and practice* (pp. 227–304). Erlbaum.

Winne, P., & Hadwin, A. (2008). The weave of motivation and self-regulated learning. In D. Schunk & B. Zimmerman (Eds.), *Motivation and self-regulated learning: Theory, research, and applications* (pp. 297–314). Taylor & Francis.

Wirth, J., Stebner, F., Trypke, M., Schuster, C., & Leutner, D. (2020). An interactive layers model of self-regulated learning and cognitive load. *Educational Psychology Review*, *32*(4), 1127-1149.

Woolley, J. S., Deal, A. M., Green, J., Hathenbruck, F., Kurtz, S. A., Park, T. K., ... & Jensen, J. L. (2018). Undergraduate students demonstrate common false scientific reasoning strategies. *Thinking skills and creativity, 27,* 101-113.

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