# Robust Player Plan Recognition in Digital Games with Multi-Task Multi-Label Learning

**Alex Goslen[1], Dan Carpenter[1], Jonathan Rowe[1], Roger Azevedo[2], James Lester[1]**
[1] North Carolina State University, Raleigh, NC, USA
[2] University of Central Florida, Orlando, FL, USA
{amgoslen, dcarpen2, jprowe, lester}@ncsu.edu, roger.azevedo@ucf.edu

**Abstract**

Plan recognition is a key component of player modeling. Player plan recognition focuses on modeling how and when players select goals and formulate action sequences to achieve their goals during gameplay. By occasionally asking players to describe their plans, it is possible to devise robust plan recognition models that jointly reason about player goals and action sequences in coordination with player input. In this work, we present a player plan recognition framework that leverages data from player interactions with a planning support tool embedded in an educational game for middle school science education, CRYSTAL ISLAND. Players are prompted to use the planning tool to describe their goals and planned actions in CRYSTAL ISLAND. We use this data to devise data-driven player plan recognition models using multi-label multi-task learning. Specifically, we compare single-task and multi-task learning approaches for both goal prediction and action sequence prediction. Results indicate that multi-task learning yields significant benefits for action sequence prediction. Additionally, we find that incorporating automated detectors of plan completion in plan recognition models improves predictive performance in both tasks.

## Introduction

Recent years have seen growing interest in player modeling in games. Data-driven approaches to player modeling provide an unobtrusive way to adapt games to individual player's needs and intentions (Hooshyar, Yousefi and Lim 2018). An important player modeling task is player plan recognition, which is the process of inferring players' goals and plans through observations of player interactions with a game (Albrecht, Zukerman and Nicholson 1997). Goal setting and planning are critical to how players approach digital games. Players will often develop plans about how to approach challenging tasks or puzzles. In educational games, setting goals and building plans is central to becoming a self-regulated learner (Dever et al. 2022). Devising computational models of player plan recognition enables the creation of player-adaptive games that can assess and support goal setting and planning processes to improve players' gameplay experiences and engagement. For example, if a player plan recognition model can accurately predict what a player is planning, the game can provide feedback or hints or tailor components of the game scenario based upon the player's plans.

Player plan recognition focuses on utilizing lower-level observations of individual players' strategies to infer high-level goals and plans for achieving them. Goal recognition falls under the umbrella of plan recognition, where only high-level goals are predicted (Blaylock and Allen 2003). While there has been considerable work on player goal recognition for player modeling purposes, very few applications of plan recognition have been explored in open-world game environments. Additionally, little work has been done to understand how to best externalize and leverage players' goal setting and planning processes for player plan recognition modeling.

This paper presents a player plan recognition framework that uses long-short term memory (LSTM) networks to predict players' goals and the action sequences players identify as helping to achieve their goals. The testbed for the framework is gameplay data from an open-world game designed to teach middle school microbiology, CRYSTAL ISLAND. In this game, players are prompted to construct plans with an embedded planning support tool. Plans in this case consist of high-level goals and sets of low-level in-game actions the player can enact in the game. Using these plans, we formalize the two prediction tasks as multi-label multi-task learning problems. We compare the performance of this framework to a single-task LSTM classifier. Additionally, we incorporate automated detectors for goal and action sequence completion and compare results to both single and multi-

task performance. Our aim is to investigate the effectiveness of using multi-task techniques and plan completion detectors to enhance player plan recognition.

## Related Work

The ability to recognize player goals and plans in digital games provides insight into how to adapt games to player behaviors, performance, and interests (Duarte et al. 2020; Sukthankar et al. 2014). Recent work has investigated using theory of mind (ToM) to inform plan recognition models for plan intervention (Weerawardhana, Whitley, and Roberts 2021), multi-agent cooperation (Boeda 2021), and intention supporting planners (Ware and Siler 2021). ToM is the ability to understand and predict intent, mental models and other cognitive characteristics, which is important when applied to player modeling (Shergadwala, Teng, and El-Nasr 2021). There has been a wide variety of methodologies for constructing such recognition problems that use ToM for plan and goal recognition tasks, such as recursive neural networks (Bisson, Larochelle, and Kabanza 2015), combinatory categorial grammars (Rabkina et al. 2022), and hierarchical task networks (Rabkina et al. 2021). While these methods have been shown to work well in digital games with pre-defined states, little work has been done on the potential of machine learning-based plan recognition in open-world digital games.

Player action sequences are highly idiosyncratic and exploratory in open-world games. LSTMs are broadly effective at handling noisy, probabilistic data. Prior work on player goal recognition in open-world games has found that LSTMs outperform several non-LSTM baselines (e.g., non-recurrent deep neural networks, conditional random fields, Markov logic networks, n-grams) across a range of evaluation metrics (Min et al. 2016, Min et al. 2017). We extend this work by formalizing player plan recognition in terms of two complimentary prediction tasks: (1) goal recognition of high-level player goals and (2) action sequence recognition of low-level actions players enact in the game environment. Another contribution of this work is the use of a planning support tool to construct labels for player plan recognition. Leveraging these labels, we translate players' gameplay into action sequences to sequentially model student plans with LSTMs.

Recent research has investigated techniques to improve plan recognition models in finite-state environments that have predetermined goals and states. Massardi, Gavel, and Beaudry (2019) examine the use of a particle filter to reduce noise in the low-level observations provided as input to the prediction model and subsequently reduce error in the keyhole plan recognition task. This approach is shown to be efficient but requires a plan library specific to the environment. Another approach utilizes parsing techniques to verify and predict plans constructed in a hierarchical task network (Bartak, Ondrkova and Maillard 2019). Additionally, deleting action sequences from invalid plans has been shown to aid in correcting hierarchical plans and help with the explainability verifying plans (Bartak et al. 2021). Although these examples rely on defined maps of the environment and appropriate plans, these approaches demonstrate the usefulness of preprocessing steps and utilizing in-game action sequences to enhance plan recognition models. Our work also utilizes the concept of plan verification to enhance plan recognition models' predictions by detecting when players complete goals they have externalized with the planning support tool. We incorporate a form of action deletion in the label set using this technique, which is an extension of prior work and a novel contribution of our plan recognition framework.

## Plan Recognition Framework

Our plan recognition framework utilizes low-level game events as inputs for two prediction tasks: player goal prediction and action sequence prediction. Additionally, we leverage players' interactions with a planning support tool to generate labels for each prediction task.

### CRYSTAL ISLAND Testbed

CRYSTAL ISLAND is an open-world game-based learning environment for middle school science in which students investigate a mysterious outbreak on a remote island research station (Figure 1). During the game, players have a first-person view of the island as they converse with non-playable characters (NPCs), read virtual books and posters, test items in a virtual laboratory and explore different locations on the island.

Figure 1. CRYSTAL ISLAND open-world environment.

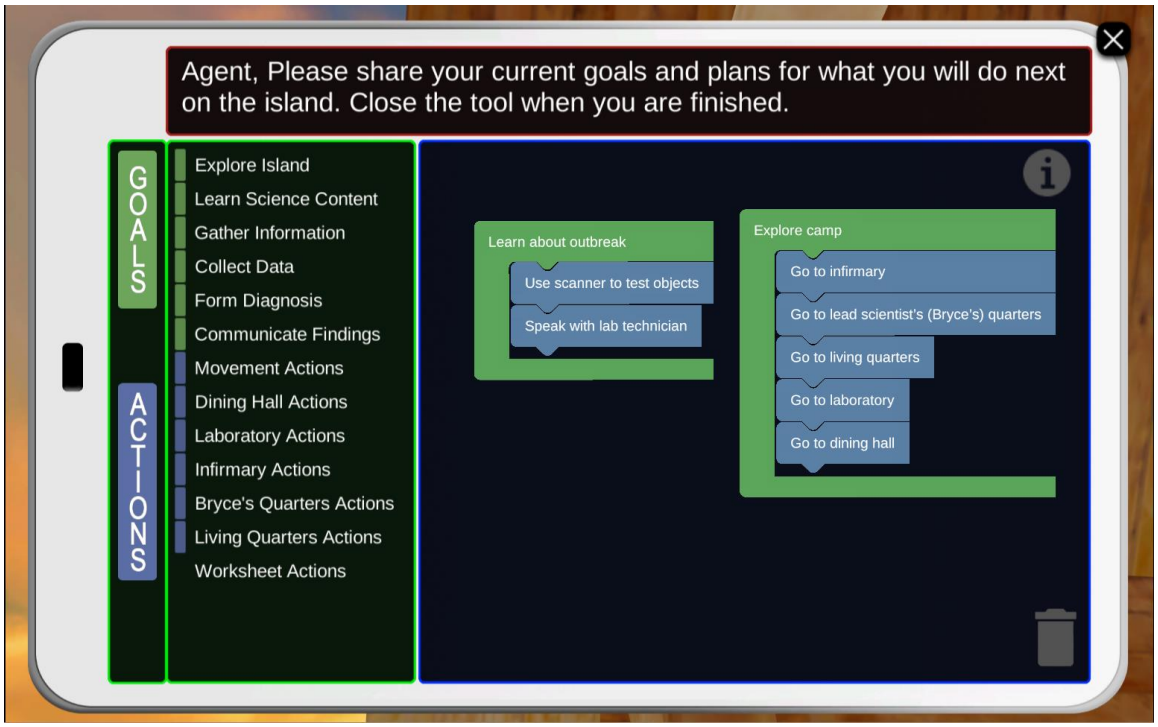


Figure 2. Example planning support tool interaction.

Throughout gameplay, players are prompted to build plans using a drag and drop, block-based visual interface inspired by visual programming languages (Figure 2). Each plan consists of a goal clamp that represents a high-level goal in the game and a series of nested actions that represents low-level trace events that can be enacted in the game. The size of plans is not restricted, and players can access the tool voluntarily throughout gameplay. All in-game actions, including planning support tool usage are logged and available for offline analysis.

The dataset used for this analysis was collected from 144 eighth grade students (60% female, 40% male). Students played CRYSTAL ISLAND over a two-day span asynchronously during remote science class time due to the COVID-19 pandemic. Students were not given a time limit to complete the game and averaged 94.7 minutes (SD = 47.7) of gameplay. They were also given an introductory video describing the game mechanics and planning support tool and asked to complete pre- and post-tests.

## Framework Input

The trace logs generated from student gameplay represent sequences of actions taken while interacting with CRYSTAL ISLAND. We refer to these as in-game event sequences. Each in-game event contains three types of features: event type, event argument and location.

- **Event type.** Event types were derived from the various activities a player can take in the game. There were 9 total event types: moving to another location, reading a book or article, completing questions about a book or article, filling in items in the diagnosis worksheet, viewing a poster, having a conversation with a NPC, submitting a final diagnosis, completing a plot point in the game, and scanning items for disease.
- **Event argument.** The event arguments generated are specific to the event type. For example, if the event is reading a book, the event argument will be the title of the book. This feature is used to provide more information about the event type. 108 unique event arguments were derived from the gameplay data.
- **Location.** The location feature represents the area of the island the event took place. The game environment contains 24 unique locations. If the event type is movement, the location feature represents where the player moved to.

CRYSTAL ISLAND's data logging system produces a single event sequence for each player that captures key actions they performed in the game. These complete event sequences need to be broken up to construct smaller event sequences that correspond to the goals and plans players had at different points during gameplay. To construct these smaller event sequences, we split the complete event sequences at each interaction with the planning support tool. Since players are asked to use the planning support tool to externalize their goals and plans, we assume that the in-game events that occur after an interaction with the tool are steps to enact the externalized goals and plans. When a player interacts with

the planning support tool again, we assume that the updated contents of the tool represent their current goals and plans. Thus, an event sequence begins with the event occurring directly after a player's planning tool interaction and ends with the event immediately before the next planning tool interaction. There was a wide range of event sequence lengths (min=1, max=454), meaning there was a wide variety of planning support tool interactions for each player. To account for this, we set the maximum event sequence length to be the median across players: 30. Sequences of less than 30 events were zero-padded to provide a fixed-length input. We constructed these event sequences cumulatively for action-level prediction. Once event sequences were segmented by planning support tool use, we created a vector representation of these sequences using one-hot encoding vectors. These steps have been shown in prior work to be the most effective for goal recognition tasks (Goslen, et al. 2022; Min et al., 2017). There were 385 event sequences after processing the data across all players. Once constructed cumulatively, we had 11,550 total sequences.

## Framework Prediction Tasks

We constructed both goal prediction and action sequence prediction tasks as multi-label classification problems in which a trained classifier predicts which selected goals a player has chosen, as well as the set of planned actions they indicate they intend to take to achieve that goal.

### Goal Recognition

The planning support tool provides 20 possible goals for players to select, which fall in five categories. We utilized these five categories as labels for the goal recognition task: Collect Data (22%), Communicate Findings (4%), Form Diagnosis (13%), Learn Science Content (22%), and Gather Information (40%). Because Crystal Island is an open-world game, we cannot assume a player will work towards only one goal at a time. Thus, we formalized the problem as a multi-label classification task, where each event sequence is assigned a binary label vector of length five that corresponds to the given goal categories.

### Action Sequence Recognition

The planning support tool provides 55 possible actions from which students can construct plans for their goals. Six action categories were designed to provide broader context to the planned actions. Because plans often contained more than one planned action (mean=2.58, SD=1.96) per selected goal, we used these categories as well as the following steps to formalize the action sequence recognition task into a multi-label classification problem. First, we concatenated all planned actions together for each selected goal. Then, we applied SpaCy word embeddings to each of these sets of planned actions (Levy and Goldberg 2014; Srinivasa-Deskan 2018). Next, we averaged the word embeddings across each word and applied k-means clustering to find patterns in the player constructed plans. We used the Elbow

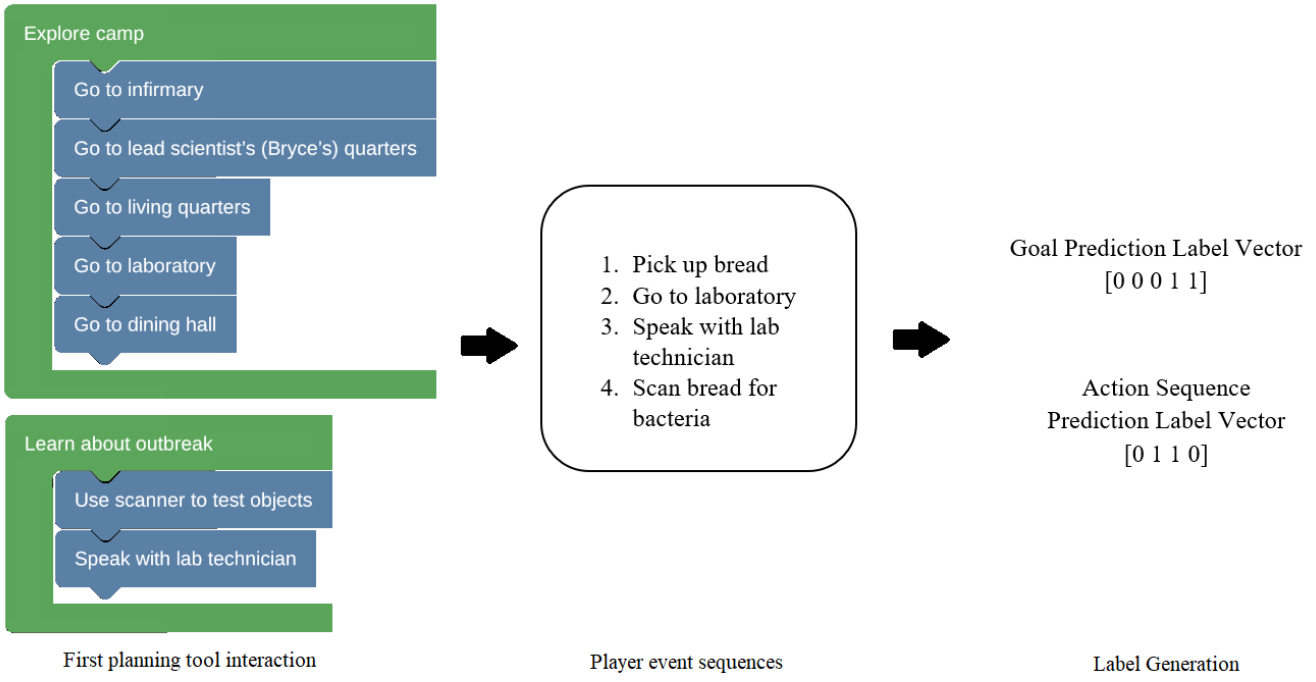


Figure 3. Representation of the label assignment process. The label vectors shown above are generated from the player's first planning tool interaction. In this case, the model would take in the one-hot encoded vector representation of the player event sequences and be trained on the given labels.

method to determine 4 to be the appropriate number of clusters for the dataset (Bholowalia and Kumar 2014). Similar to goal label construction, we used these clusters of class labels in the form over a binary vector of length 4. The resulting clusters appeared to align with the most used action category in each plan. "Read Science Content" was primarily found in Cluster 0 (9%). "Explore" was primarily used in Cluster 1 (30%). Plans mostly contained "Gather and Scan Items" in Cluster 2 (33%), and plans mostly contained "Speak with Characters" in Cluster 3 (28%).

### Automatic Plan Completion Detection

The original framework this work extends assumes players frequently update their goals and plans to what they want to achieve next (Goslen et al. 2022). A large component of this is marking plans as being completed once the player has enacted all appropriate events to achieve a given goal. If a plan is not marked as being complete, it is left as a label in the plan recognition framework. This creates a problem when training the models, as it is being trained on event sequences that might not be representative of that plan. To alleviate this issue, we incorporated a preprocessing step to automatically identify when a plan has been completed and remove it from the label set.

Because CRYSTAL ISLAND is an open-world game environment, there is not one specific way that a player could achieve a selected goal. For instance, one possible goal a player can select is "Explore Island". There are 24 locations in the game, so deciding when a player has explored the island enough to complete the goal is not a simple task. To solve this problem, we used players' planned actions to determine if the goal was complete. That is, if a player completed the entire set of actions in a plan in their previous planning instance and that plan was still present in the next planning instance, both the goal and set of planned action sequences would be removed from the label set. For example, consider the plans and event sequences from Figure 2. As we can see in the player's in-game events executed after the first planning instance, the player enacted all the steps in the bottom plan, "Learn about outbreak." If in the next planning interaction, the player kept the "Learn about outbreak" plan in their planning tool, this goal and action sequence label would not be included in the respective label vectors. The resulting dataset had the following distribution of goals labels: (1) Collect Data: 22%, (2) Communicate Findings: 5%, (3) Form Diagnosis: 6%, (4) Learn Science Content: 22%, and (5) Gather Information: 46% and the following distribution of plan labels (0) Read Science Content: 10%, (1) Explore: 28%, (2) Gather and Scan Items: 34%, (3) Speak with Characters: 28%.

## Evaluation

To evaluate both goal and action sequence recognition tasks, we investigated three different types of computational models that all shared the same event sequence representation as input: (1) single-task multi-label classification, which trains and predicts goal and action sequence prediction models separately (Figure 4), (2) multi-task multi-label classification, which allows models for each task to be informed by the other (Figure 5), and (3) enhanced multi-task evaluation, which incorporates automatic detection of plan completion into player plan recognition.

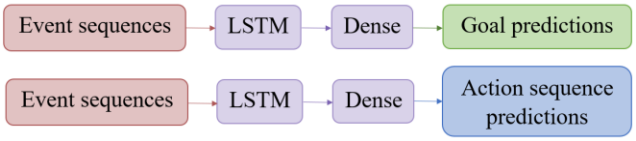


Figure 4. Single-task model architecture.

Long short-term memory (LSTM) networks were used for all three types of models. Both the single-task and multi-task models were trained on one hidden layer with 100 units. We used nested 5-fold cross validation, with iterative grid search applied to the inner fold for hyperparameter tuning of batch size (64 and 128) and number of training epochs (50 and 100). We also used a stratified player-level split within folds to ensure similar label distribution and eliminate data leakage between training and test splits. Since the hyperparameters were tuned as part of a nested 5-fold cross-validation procedure, the optimal hyperparameters chosen for each fold differed.

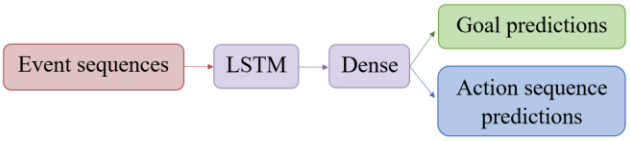


Figure 5. Multi-task model architecture.

Macro-average F-measure was used to evaluate the models' predictive performance. Macro-average F-measure performs well on imbalanced datasets because it calculates the average F-measure for each class label individually before aggregating the averages together (Pereira et al. 2018). Both plan and goal label distributions are imbalanced, making macro-average F-measure an appropriate choice for evaluation. Additionally, F-measure works well in multi-label classification because its calculations utilize false positives and false negatives, emphasizing incorrectly classified labels (Liu and Chen 2015; Madjarov et al. 2012). Evaluating performance based on this type of calculation is useful for designing models for player-adaptive games.

| | Collect data | Communicate 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** |

Table 1. F-measure goal recognition results for all three experiments. Distribution of labels represents the distribution of the test set from the 5-fold cross validation.

| | 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** |

Table 2. F-measure action sequence results for all three experiments. Distribution of labels represents the distribution of the test set from the 5-fold cross validation.

## Results

This section presents the results for both goal and action sequence prediction tasks. We compared the performance of a single-task LSTM to a multi-task multi-label LSTM classification task, as well as an enhanced multi-task model that included detection of plan completion.

### Goal Recognition Results

Table 1 shows that the overall performance of the trained goal recognition models was the same for single and multi-task goal recognition. Individual performance across classes did not differ significantly either. However, we did see an improvement in predictive performance when removing completed goals from the label set. This preprocessing removed 77 total goals from the label set.

Although the distribution of labels did not change much, a 3% improvement in macro F-measure implies that detecting plan completion reduced noise in the dataset. The enhanced multi-task model that incorporated plan completion performed best for two out of the five goal categories, with the highest improvement in F-measure being seen in "Communicate findings". These results imply that using the plan completion logic helped to boost performance.

### Action Sequence Recognition Results

Results in Table 2 show an improvement in macro F-measure for multi-task action sequence recognition compared with the single-task model's performance. This implies that players' selected goals help to inform the action sequence prediction models.

Like in the goal recognition task, there is an improvement in overall F-measure performance after including the automated plan completion detectors. The plan completion detection preprocessing removed 58 sets of action sequences from the label set because they were already completed in gameplay. The enhanced multi-task model performed best for two out of four of the sets of plans, with "Gather and scan items" showing the most improvement in F-measure.

Notably, there was a decrease in F-measure performance for the least represented set of plans, "Read science content," in both multi-task models. Based on the multi-task model architecture, we can infer that the goal categories

used in players' plans might have caused this performance drop. Players can access reading material in all locations in CRYSTAL ISLAND, and reading science content could aid in achieving almost all goals found in the planning support tool, meaning that players could use "Read science content" action sequences in a wide variety of ways. More investigation into how players used this action sequence category in relation to goal categories is needed to fully understand this decrease in performance.

## Discussion

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 player plan recognition models in game-based learning environments, as does accounting for player goal and action completion in player plan recognition models. While these results indicate improvement in player plan recognition models, there were some limitations with the framework.

Using players' goal setting and planning processes provides a new way to construct plan recognition models, but it also relies heavily on how players utilize the embedded planning support tool in CRYSTAL ISLAND. Event sequences are segmented when players open and close the tool and there was considerable variance in the number of times players opened the planning tool. If a player does not open the planning support tool until the end of the game, the presented framework could train player 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 players choose to interact with the planning support tool.

Additionally, our framework partially assumes players 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 players 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 player strategies to solve the mystery of the game. Further analysis could be done to understand the relationship between selected goal categories and action sequences with where they occur in the game. Aligning science problem solving logic to goal categories and action sequences might help to inform player plan recognition models in this context, since the environment is a narrative scenario based on science problem-solving. This could provide further insight into player strategies.

## Conclusion

This work presents a player plan recognition framework that leverages players' interactions with a planning support tool in an open-world learning environment to predict player goals and planned action sequences for achieving that goal. The presented framework takes gameplay observations as input and uses players' selected goals and action sequences to construct a multi-label multi-task formalization of player plan recognition. Specifically, the framework is centered on two complementary prediction tasks: player goal prediction and player action sequence prediction. Models for both tasks were evaluated as single tasks as well as multi-tasks using LSTMs. These techniques proved to be beneficial for action sequence prediction, with an overall macro-average F-measure improvement. Additionally, automatic detection of plan completion was incorporated into the multi-task LSTM model for further analysis as an enhancement to the player plan recognition framework. In both tasks, we saw improvement in macro-average F-measure, indicating that this preprocessing step is beneficial for the prediction models.

These results highlight the potential of player plan recognition models in player-adaptive digital games. Future work could be done to explore plan verification techniques in player-adaptive environments to help inform the plan recognition models, as well as players' strategies throughout gameplay. Investigating how to devise models that can identify goal abandonment throughout gameplay would be beneficial for enhancing goal and action sequence recognition. Furthermore, incorporating run-time plan recognition models into player-adaptive games to enhance players' gameplay experiences has significant promise as a future direction.

## Acknowledgements

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

## References


Albrecht, D. W., Zukerman, I., & Nicholson, A. E. 1998. Bayesian models for keyhole plan recognition in an adventure game. *User modeling and user-adapted interaction*, *8*(1), 5-47.

Barták, R., Ondrckova, S., & Maillard, A. 2019. Parsing-based Approaches for Verification and Recognition of Hierarchical Plans. *In ICAPS 2019 Workshop on Hieararchical Planning*.

Barták, R., Ondrčková, S., Behnke, G., & Bercher, P. 2021, September. Correcting Hierarchical Plans by Action Deletion. *In Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, *18*(1), 99-109.

Bholowalia, P., & Kumar, A. 2014. EBK-means: A clustering technique based on elbow method and k-means in WSN. *International Journal of Computer Applications*, *105*(9).

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*.

Blaylock, N., & Allen, J. 2003, August. Corpus-based, statistical goal recognition. *In IJCAI*, *3*, 1303-1308.

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.

Dever, D. A., Amon, M. J., Vrzáková, 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*.

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

Goslen, A., Carpenter, D., Rowe, J. P., Henderson, N., Azevedo, R., & Lester, J. 2022. Leveraging Student Goal Setting for Real-Time Plan Recognition in Game-Based Learning. *In International Conference on Artificial Intelligence in Education*, 78-89. Springer, Cham.

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

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.

Liu, S. M., & Chen, J. H. 2015. A multi-label classification based approach for sentiment classification. *Expert Systems with Applications*, *42*(3), 1083-1093.

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.

Massardi, J., Gravel, M., & Beaudry, E. 2019. Error-tolerant anytime approach to plan recognition using a particle filter. *Proceedings of the International Conference on Automated Planning and Scheduling*, *29*, 284–291.

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.

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.

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*.

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.

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.

Wang, L., Chen, Z., Lin, N., & Huang, X. 2021. An Interdisciplinary Literature Classifier Based on Multi-task Multi-label Learning. *2021 International Conference on Asian Language Processing (IALP)*, 183–188.

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.*