JOURNAL OF LEARNING ANALYTICS

SOLAR SOCIETY for LEARNING ANALYTICS RESEARCH

Early Access Articles, 1–23. https://doi.org/10.18608/jla.2022.7465

Analyzing Students' Problem-Solving Sequences: A Human-in-the-Loop Approach

Erica Kleinman[1], Murtuza N. Shergadwala[2], Zhaoqing Teng[3], Jennifer Villareale[4], Andy Bryant[5], Jichen Zhu[6], Magy Seif El-Nasr[7]

Abstract

Educational technology is shifting toward facilitating personalized learning. Such personalization, however, requires a detailed understanding of students' problem-solving processes. Sequence analysis (SA) is a promising approach to gaining granular insights into student problem solving; however, existing techniques are difficult to interpret because they offer little room for human input in the analysis process. Ultimately, in a learning context, a human stakeholder makes the decisions, so they should be able to drive the analysis process. In this paper, we present a human-in-the-loop approach to SA that uses visualization to allow a stakeholder to better understand both the data and the algorithm. We illustrate the method with a case study in the context of a learning game called Parallel. Results reveal six groups of students organized based on their problem-solving patterns and highlight individual differences within each group. We compare the results to a state-of-the-art method run with the same data and discuss the benefits of our method and the implications of this work.

Notes for Practice

- This work demonstrates that a human-in-the-loop approach to analyzing students' problem-solving sequences results in more interpretable insights than purely quantitative methods do.
- Visualizing the output of a clustering algorithm allows stakeholders to identify learning patterns in and across sequences.
- The visualized output also allows stakeholders to better understand how the algorithm interprets the sequences, which can be used to adjust the algorithm's parameters.
- Results suggest that allowing the stakeholder to analyze the visualized sequences and iteratively adjust the algorithm produces clustered sequences that better match an expert's interpretation of the data.

Keywords

Learning analytics, sequence analysis, visualization, human-in-the-loop methods, mixed methods, game-based learning

Submitted: 04/26/2021 — **Accepted:** 12/07/2021 — **Published:** 06/04/2022

Corresponding author [1] Email: emkleinm@ucsc.edu Address: Computational Media, University of California, Santa Cruz, 1156 High Street, Santa Cruz, California, 95064, USA. ORCID ID: https://orcid.org/0000-0002-6269-3848

[2] Email: mshergad@ucsc.edu Address: Computational Media, University of California, Santa Cruz, 1156 High Street, Santa Cruz, California, 95064, USA. ORCID ID: https://orcid.org/0000-0003-2337-4809

[3] Email: zhteng@ucsc.edu Address: Computational Media, University of California, Santa Cruz, 1156 High Street, Santa Cruz, California, 95064, USA.

[4] Email: jmv85@drexel.edu Address: Digital Media, Drexel University, 3141 Chestnut Street, Philadelphia, Pennsylvania, 19104, USA. ORCID ID: https://orcid.org/0000-0002-7315-3601

[5] Email: a@andymbryant.com Address: Computational Media, University of California, Santa Cruz, 1156 High Street, Santa Cruz, California, 95064, USA.

[6] Email: jicz@itu.dk Address: Digital Media, IT University of Copenhagen, Rued Langgaards Vej 7, DK-2300 Copenhagen S, Denmark. ORCID ID: https://orcid.org/0000-0001-6740-4550

[7] Email: mseifeln@ucsc.edu Address: Computational Media, University of California, Santa Cruz, 1156 High Street, Santa Cruz, California, 95064, USA. ORCID ID: https://orcid.org/0000-0002-7808-1686

1. Introduction

Personalized learning requires a detailed understanding of the learner's problem-solving processes, often obtained through the granular analysis of their actions within an educational environment (Shute et al., 2010; Min et al., 2016; Kinnebrew & Biswas, 2012; Baker et al., 2006). By connecting learners' observable actions to the concepts taught, it is possible to infer a student's mastery of skills and personalize their education accordingly (Shute et al., 2010; Vahdat et al., 2015). This is the goal of many approaches to analyzing students' observable actions, including goal recognition (Min et al., 2016; Ha et al., 2014) and Bayesian approaches (Shute et al., 2010; de Klerk et al., 2015). The drawback of these approaches, however, is that they are overly focused on the outcomes of a student's problem solving, rather than the process. As such, the exact strategies employed by a student as they work through a problem, which are necessary to truly identify where they are struggling and personalize education to their needs, are lost.

A promising avenue for analyzing problem-solving processes, and the focus of this work, is the study of sequences of learners' moment-to-moment actions using visualization (Doleck et al., 2015; Hicks et al., 2016; Horn et al., 2016; Bakhshinategh et al., 2018; Malmberg et al., 2017). Sequence-based approaches have demonstrated value as an effective method for identifying individual differences within a group of learners and patterns across a community (Kinnebrew & Biswas, 2012; Kinnebrew et al., 2013). By leveraging visualizations, such techniques can also illuminate the granular details of students' problem-solving strategies (Horn et al., 2016). Unlike goal recognition or Bayesian approaches, these approaches focus specifically on process and are better suited to capturing problem-solving strategies.

The primary drawback of existing sequence-based approaches to studying problem solving, however, is the absence of a human in the loop. We define a human-in-the-loop method as one where a human stakeholder, meaning a person who would use the approach in practice, such as an educator or analyst, can interpret the output of the model and then provide input to the model that impacts how it analyzes, compares, or clusters the sequences. This interaction loop is iterative, with a human reading the output, providing input, reading the new output, and providing new input in a cyclical manner until they feel the output presents an accurate and insightful view of the data. Such an approach can produce results that are more interpretable to a human, especially to the human in the loop, who had a direct hand in producing them. However, existing techniques often do not present sequences in an interpretable manner, either due to lack of visualization or because of the scale of the data. Further, while open-learner models have seen great success with adjustable models (Bodily et al., 2018; Hooshyar et al., 2020; Zhu & El-Nasr, 2021), these models are not often used to analyze students' problem-solving sequences, leaving no opportunity for a human analyst or an educator to provide input to the calculation (Liñán & Pérez, 2015; Papamitsiou & Economides, 2014; Romero & Ventura, 2010).

In education, especially when looking at problem solving, the granular, moment-to-moment actions taken by a student, such as entering and then deleting a potential solution, are critical to understanding the student's thought processes. In this context, we argue that a human-in-the-loop approach to sequence analysis (SA) can produce more interpretable results by allowing stakeholders to understand and correct the model and its outputs. Since a human stakeholder makes the final decisions in an educational setting (Vahdat et al., 2015), it is important to generate not only a system that can justify its conclusions but also one that can allow the human to take charge and correct the system if they believe it is incorrect. Existing literature has demonstrated that approaches where a human works iteratively with a system can result in more interpretable, actionable insights regarding subject behaviour (Ahmad et al., 2019; Kleinman et al., 2020; Horn et al., 2016; Malmberg et al., 2017; Zhu & El-Nasr, 2021).

In this study, we present a human-in-the-loop approach to analyzing students' problem-solving sequences. Similar to most sequence-clustering methods used in the social sciences to group action sequences based on similarities and differences in the ordering of events (Lesnard, 2006; Abbott & Tsay, 2000), we use an algorithm to measure the distance between the sequences in each pair (Lesnard, 2006). Specifically, our method uses dynamic time warping (DTW) (Berndt & Clifford, 1994) and clusters sequences based on their distance to an expert's sequence. However, unlike many existing techniques, which set hyper-parameters based on intuition or standard practices, our method leverages visualization to facilitate a human in the loop. Specifically, through interactive visualization, our method allows stakeholders to analyze and generate their own understanding of the data. This, in turn, allows them to interpret the output of DTW, which further allows them to adjust/optimize the parameter values for the algorithm. Through this iterative conversation between a human expert and an algorithm, our method produces clusters based on learning patterns and individual differences that are human driven and more interpretable and actionable to a human because they were directly involved in its creation.

We present a formalization of our method and demonstrate its use through a case study with a learning game called Parallel. For the case study, we identify six student clusters, each characterized by a different approach to problem solving, revealing valuable details about students' understanding of the educational material. To further demonstrate how our approach differs from one that does not involve a human in the loop, we compare it with a state-of-the-art statistical SA method presented by Sawyer and colleagues (2018). Sawyer and colleagues' (2018) method is similar to ours in that it looks at problem solving

at a granular level, examining the moment-to-moment actions students take while completing a problem-solving task, and clusters them based on a calculated distance to an expert's trace. However, unlike our method, theirs does not feature a human in the loop. Thus, we use this comparison to highlight the advantages of a human-in-the-loop approach, which produced more interpretable, human-driven results that more closely match an expert's understanding of the data.

2. Related Work

In this section, we will discuss previous works that have analyzed problem solving through SA and highlight the absence of approaches that enable a human in the loop to provide input. SA is one of the most effective ways to analyze learning because it offers a detailed and granular examination of student strategies based on the order in which they perform actions (Bakhshinategh et al., 2018; Papamitsiou & Economides, 2014; Romero et al., 2009; Valls-Vargas et al., 2015; Gasevic et al., 2017; Iske, 2008; Kinnebrew & Biswas, 2012; Kinnebrew et al., 2013; Köck & Paramythis, 2011). Looking specifically at problem solving, existing sequence-based approaches focus specifically on clustering and comparing sequences (Sawyer et al., 2018; Paaßen et al., 2018; Eichmann et al., 2020; He et al., 2021). Sequence approaches also encompass hidden Markov models (HMMs), which identify meaningful interaction patterns and infer student problem-solving strategies or predict future actions (Jeong et al., 2008; Balakrishnan & Coetzee, 2013; Boumi & Vela, 2019; Geigle & Zhai, 2017; Doleck et al., 2015).

An HMM is a statistical model of observable events that depend on unobservable, or hidden, factors (Balakrishnan & Coetzee, 2013; Doleck et al., 2015). In the context of problem solving, HMMs are used to identify meaningful interaction patterns and infer student strategies or predict future actions (Jeong et al., 2008; Balakrishnan & Coetzee, 2013; Boumi & Vela, 2019). An illustrative example is the work of Doleck and colleagues (2015), in which they used an HMM to identify different problem-solving strategies in a computer-based learning environment for medical students. The environment logs students' interactions with various screens and tools as they attempt to address a medical case, allowing sequences to be extracted. The authors used an existing HMM tool (Biswas et al., 2010) to analyze these sequences, generate HMMs representing transition probabilities between various states, and identify patterns in how students interacted with the system and solved problems while trying to complete their tasks (Doleck et al., 2015).

The advantage of HMMs is their ability to model behaviour based on observable actions while also considering unobservable states, which makes them well suited to the task of understanding problem solving, which often heavily relies on unobservable states. The drawback, however, is that, like the work of Doleck and colleagues (2015) illustrates, deriving meaning from an HMM relies on human researchers' ability to understand the abstract hidden states, which is not guaranteed, given the statistical nature of the model's creation. This makes it difficult, if not impossible, for HMMs to enable a human in the loop to provide input to the calculations, and the outputs are, instead, completely determined by the model.

More commonly used to study problem solving are sequence-clustering approaches, which group sequences based on similarities, measured using techniques such as optimal matching, k-means, or edit distance (Iske, 2008; Paaßen et al., 2018; Reilly & Dede, 2019). A key example is the work of Eichmann and colleagues (2020), which used qualitatively coded sequences and optimal matching to identify strategies used by students in a complex problem-solving context. More specifically, they analyzed log data from two different problem-solving tasks and generated sequences by coding each set of logged actions as initial goal-directed behaviour, repeated goal-directed behaviour, initial non-targeted exploration, repeated non-targeted exploration, or resetting (Eichmann et al., 2020). They then used optimal matching (Lesnard, 2006) to determine the dissimilarity between sequences and group them into clusters. Through their method, they identified strategic patterns that correlated with successful or failed problem solving, such as those who failed displaying higher percentages of non-targeted exploration and shorter sequences and those who succeeded tending to repeat behaviours and only display non-targeting behaviours toward the beginning of their sequences (Eichmann et al., 2020).

The primary shortcoming of the approach of Eichmann and colleagues (2020), however, is its lack of interpretable clusters. To be more specific, this method benefits from its emphasis on coding the sequences and its use of visualization to depict sequence trends, promoting transparency and interpretability of the sequences themselves. However, there is no point of reference within the method for how the output clusters are to be understood, for example, which one contains the high performers and which the low performers, who took an unexpected approach, and so on. While this may not be an issue with a small population or clearly defined clusters, a large population or more loosely defined clusters may pose a challenge, especially if no external performance data, such as grades, is available to use as a reference. In other words, while the approach of Eichmann and colleagues (2020) may provide interpretable sequences, it does not necessarily provide interpretable clusters, since there is no internal point of reference regarding how each cluster should be understood. Without interpretable clusters, it can be difficult to determine if sequences are clustered correctly and difficult to provide informed input regarding the clustering method.

One way to provide an internal point of reference is to compare sequences against an expert or optimal sequence. This approach makes it easier to ascribe meaning to resulting clusters and identify students who may need aid, even when additional

information about their performance is not available (Sawyer et al., 2018; He et al., 2021). In this approach, problem-solving strategies are analyzed based on the degree to which they differ from an ideal solution. One example of such an approach is the work of Sawyer and colleagues (2018), which uses filtered time series analysis (FTSA) to compare sequences against an expert's sequence. More specifically, they generated sequences from logs of students' interactions with a game-based learning environment where they were trying to solve a medical mystery. Using FTSA, these sequences were compared to the sequence that an expert was expected to take. The distances were then compared against students' performance on a post-test, and Sawyer and colleagues (2018) found that those students who performed poorly on the post-test also had sequences far from the expert trace. Sawyer and colleagues' (2018) technique has also been demonstrated in a non-game context by Reilly and Dede (2019), who additionally found that the number of actions taken correlated with learning gains.

Comparison against an expert trace adds interpretability to the clusters and can be used to identify at a glance which students need aid. Thus, through this method, it is easier to identify struggling students than it is with more general clustering methods, especially when additional performance information (such as grades) is not available to provide meaning to the clusters or if clusters are not clearly defined. The drawback, however, is that this approach has yet to be demonstrated in a manner that also facilitates interpretable sequences. Without interpretable sequences, it is difficult to understand the output of the calculation (the distances) and difficult to adjust the calculation. Further, it is impossible to understand how students are struggling without a clear view of their action sequences, making it difficult to help them.

In summary, existing SA techniques have primarily benefited from interpretable sequences or interpretable clusters, but few demonstrate both. In order to enable a human in the loop to drive the analysis process, we argue that both are necessary. Interpretable clusters make it possible to evaluate the output of an algorithm (i.e., are the right sequences in the right clusters?), while interpretable sequences make it possible to define the characteristics of each cluster. Together they make it possible to provide informed input to the model and identify how students are struggling and how to help them. However, existing techniques also fail to utilize adjustable models, making it impossible to enable a human in the loop to drive the analysis process.

3. The Method

In this section, we formalize our method, shown in Figure 1, which follows the approach taken by Sawyer and colleagues (2018) and He and colleagues (2021) in which student sequences are compared against an expert's sequence to facilitate interpretation of clusters and identification of those students who need aid. However, like that of Eichmann and colleagues (2020), our method also features interpretable sequences. Further, unlike any previous work, we leverage interactive visualization to make it easier for the analyst to connect the sequence patterns to the clusters and an adjustable algorithm that takes input from the human analyst. In this manner, our method facilitates a human in the loop, allowing the stakeholder to interpret the sequences and insert their own expertise into the model.

We consider the stakeholder to be an educator or learning scientist, someone who is familiar with the educational material and the digital learning environment and capable of drawing meaningful and actionable insights from the sequences. The stakeholder may not possess the technical experience necessary to interact with the algorithm, in which case working in tandem with a data scientist is recommended. However, this method is tool agnostic, meaning that it can be implemented with any visualization that meets the usability requirements outlined in Section 3.3, including one with an interface that provides graphical user interface (GUI) inputs for algorithm manipulation.

3.1 Step 1: Data Processing

Because the method is meant to facilitate analysis of students' problem-solving strategies, the learning environment must be instrumented to record students' actions at a granular action-to-action level as they work through a problem or task. Most digital learning environments, including educational games and MOOCs, can be instrumented in this way.

To make the data interpretable, it is recommended that the stakeholders filter and abstract the logged data. To do this, the stakeholders should first identify the actions available within the learning environment that are relevant to studying problem solving, such as submitting or deleting an answer or viewing a help tool tip. We assume that the stakeholders are experienced enough with the learning environment to possess the necessary expertise to recognize these actions. However, one could also work with the developers of the environment or refer to documentation if necessary.

The stakeholders should then work with the developers of the learning environment or its logging system, or refer to available documentation, to map the logged information to the actions. For example, a mouse click in a given location may be mapped to viewing a help tool tip. Any logged data not included in the mapping can be filtered out. If the data is still too complex, that is, there are too many actions, it is recommended that stakeholders develop an abstraction for the data, converting the lower-level actions to higher-level actions or behaviours. This process will help make the data easier for a human to read and understand. Discussing the details of abstraction methods is beyond the scope of this work; however, numerous techniques,

including human labelling and iteratively combining similar states, are discussed in Ahmad and colleagues (2019), Javvaji and colleagues (2020), Kleinman and colleagues (2020), Kinnebrew and colleagues (2013), and Kinnebrew and Biswas (2012).

Finally, once filtering and abstraction have been implemented, student data is converted into sequences of actions. Action sequences can be created based on timing, with actions recorded at set intervals, or they can be based on ordering. This choice is left to the stakeholder, who will know best what information they have and what they need to learn from it. All of these steps can be accomplished through data-processing scripts.

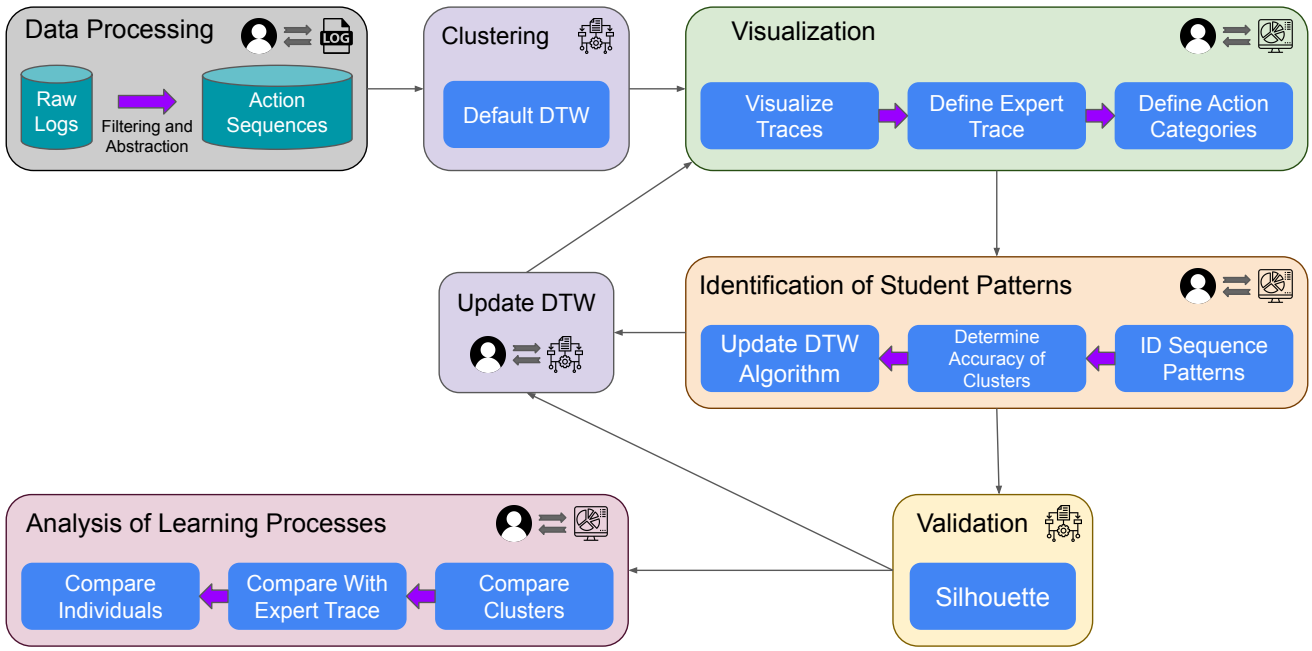

Figure 1. The method for human-driven analysis of learning sequences. In each box is an icon to indicate if the step involves just an algorithm or a human interacting with an algorithm, logs, or a visualization.

3.2 Step 2: Clustering

Once the data has been processed and converted into sequences of actions, the next step is to use the default DTW algorithm (described below) to cluster these sequences. This can be accomplished by setting up a script that takes in the set of sequences as an input and runs the algorithm.

The default DTW algorithm calculates distances as follows: Given a set of sequences S that consists of individual sequences S_1, S_2, S_3, and so on, a distance value d is created for every sequence pair (S_1, S_2). At every given step s_a in sequence S_1 the value at step s_a is compared to the value at every given step s_b in sequence S_2 and a weight w is calculated. If the states or actions at steps s_a and s_b are the same, $w = 0$; otherwise, $w = 1$. In a dynamic programming manner, the algorithm fills in a matrix with values based on these distances as it steps through both sequences. At the end, the distance value d is determined by the value at the bottom right cell of the matrix, which represents the minimum possible distance between the two sequences S_1 and S_2. The pseudocode for the DTW algorithm, originally developed by Berndt and Clifford (1994), can be seen below, in Algorithm 1.

Algorithm 1: Default DTW Algorithm; see Berndt and Clifford (1994)

for *Every sequence pair S_P consisting of sequences S_1 and S_2 in set of sequences S* **do**
  Distance $d = 0$;
  $n =$ Length of S_1;
  $m =$ Length of S_2;
  Initialize Matrix DTW with n rows and m columns;
  for *Every step s_a in Sequence S_1* **do**
    for *Every step s_b in Sequence S_2* **do**
      if $S_1(s_a) == S_2(s_b)$ **then**
        Weight $w = 0$;
      else
        Weight $w = 1$;
      end
      $DTW[s_a][s_b] = w +$ the minimum of $DTW[s_a - 1][s_b]$, $DTW[s_a][s_b - 1]$, and $DTW[s_a - 1][s_b - 1]$
    end
  end
  $d = DTW[n][m]$;
  Return d
end

3.3 Step 3: Visualization

The third step involves setting up the visualization to facilitate the human-driven analysis of the sequences. Our method includes visualization for two reasons. First, while student action sequences can be qualitatively analyzed and clustered using playback videos, visualization permits a holistic view of the entire population, making it easier to identify community-level patterns. Second, through visualization, by syncing a visualization of clusters produced by an algorithm with a visualization of student sequences, it is possible to learn how the algorithm understands the data, which would not be doable through video. With this in mind, the proposed method needs to be able to visualize all action sequences at once and display how these sequences are clustered in a synchronized manner.

Using the visualization, the stakeholder can categorize the actions that appear in the student sequences. First, the stakeholder must identify an expert trace. An expert trace (Sawyer et al., 2018; Javvaji et al., 2020) is the sequence an expert would generate given the educational environment. Such a trace can be created by having an expert interact with an educational environment and then visualizing their sequence. Alternatively, an artificial trace can be generated using expertise or domain knowledge to design the path that an expert would be expected to take.

Second, the stakeholder must leverage their knowledge of the educational environment, and the topic, to sort the actions that make up the sequences into categories based on the higher-level learning strategies that they indicate. For example, the action of re-reading a piece of text may belong to "clarification activities," where a student seeks to reaffirm their understanding of a topic, while editing an answer may belong to "adjustment activities," where a student adjusts their responses based on new information. Categories may represent inferences about the learning strategy that motivated a given action. During this process, those actions that make up the expert trace should become their own category and not be included in other categories to facilitate analysis of sequences based on what actions they take that divert from the expert trace. This process assumes that the stakeholder is familiar enough with the learning environment to either know or reasonably assume that engagement with a given action indicates a given learning strategy (clarification, adjustment, etc.).

3.4 Step 4: Identification of Student Patterns

Once the categories are set, the stakeholder can leverage their expert knowledge to analyze and compare sequences against each other and the expert trace. In doing so, they can identify the general characteristics for each set of clustered student sequences. These characteristics can then be used to identify opportunities to update the calculation and produce more accurate clusters.

Using the DTW output, which should be visualized by the chosen visualization system, the stakeholder identifies a set of sequences that are clustered near each other. Each sequence is then analyzed to identify high-level characteristics of how the student moved between action categories. Analysis at this point does not need to be extensively detailed, but rather just enough to get a high-level idea of the patterns in each sequence. This analysis is repeated for each nearby sequence in an iterative process similar to content analysis until a general characteristic for the clustered sequences is identified.

While the DTW output is used as a starting point, it should not be treated as a ground truth. During this process, the stakeholder may find that some sequences that are clustered near each other by DTW demonstrate different characteristics

based on their expert knowledge or that distanced sequences have similar characteristics. In these moments, the stakeholder's judgment should be prioritized over the algorithm's. These observations will be used in the following step to inform the means by which the stakeholder will edit the algorithm.

3.5 Step 5: DTW Update and Iteration

At the conclusion of the output analysis step, the stakeholder should understand the defining characteristics of each group of students. They should also have a rough understanding of whether or not DTW has clustered the sequences adequately according to these characteristics. At this stage, the stakeholder can update the DTW algorithm based on their knowledge of the patterns in each cluster.

In a process similar to what is described by Javvaji and colleagues (2020), the stakeholder can update the value added by DTW when there is a mismatch. We refer to these values as weights. Specifically, while the default algorithm adds a weight of one to any mismatch, the stakeholder can use their understanding of how students are interacting with the action categories to penalize actions within certain categories or, if necessary, specific actions. This is done by defining a specific, greater value as the weight that should be added if a mismatch includes the designated action. We leave it to the judgment of the stakeholder to decide which action categories should be penalized over others and what the appropriate weights are. In general, it is advised to greater penalize those categories that contain actions deemed to be unlikely to be taken by an expert. Through this process, the stakeholder can tune and correct the DTW algorithm such that it produces more accurate clusters by ensuring that distances between sequences are calculated based on how they display various action patterns.

When using DTW to produce clusters, there is always a risk that sequence length will impact the results. With our method, this can be mitigated by manipulating the weights. For example, if sequence length is a confounding factor in the clustering process, weights can be adjusted such that, in situations where one sequence has ended but the other continues, a reduced weight, such as 0.25, can be applied for each step that the longer sequence continues. Pairing such an arrangement with increased weights, that is, two, three, or four, for disparate sets of actions in situations where neither sequence has concluded will ensure that the differences in actions will have a greater impact on the clusters than the differences in overall length. We leave it to the judgment of the stakeholder to determine if such an arrangement is necessary.

3.6 Step 6: Validation

Once we have satisfactory clusters, we then need to evaluate these clusters. We recommend using the silhouette measure (Rousseeuw, 1987), which measures how similar an object is to its own cluster compared to other clusters, on a scale of -1 (not similar) to 1 (similar), based on distances between objects. Silhouette is a standard for measuring clustering effectiveness in terms of cohesion and separation. If the silhouette result is not satisfactory, the stakeholder can update the DTW algorithm again and repeat the visualization and identification steps.

3.7 Step 7: Granular Analysis of Learning Processes

In this step, we can use the state graph to conduct a detailed analysis of student learning. This analysis follows a three-step process of comparing groups and individual sequences against each other and the expert trace.

First, the stakeholder compares each sequence grouping against the other groupings in terms of their defining characteristics. The goal is to identify and reaffirm how each group of sequences differs from others in terms of which action categories they exhibit and how they move between them. Doing so can help the stakeholder better understand student performance and learning patterns across an entire population holistically.

Second, the stakeholder compares each sequence group against the expert trace. For each sequence group, the stakeholder should focus on how the sequences within tend to converge with, or divert from, the expert trace. The actions and transitions in which the included sequences tend to align with the expert trace may indicate concepts that the students are able to grasp, while actions and transitions that divert from the expert trace may indicate concepts that the students are struggling with.

Third, the stakeholder analyzes in detail each individual sequence within a group to see how a given student's sequence compares to or contrasts with that of others in their group. During this process, it is also important to compare the individual sequences in each group against the expert trace to get an idea of how each student is performing in relation to how an expert is expected to perform. Keeping the defining characteristics of each group in mind, it is then possible to identify individual differences in how students exhibited each characteristic. Understanding these individual differences is critical to designing effective interventions and requires granular details of a student's learning process.

3.8 Bias Mitigation

While the inclusion of a human in the loop can prevent algorithmic bias (O'Neil, 2016), there is the additional risk that the human in the loop may insert human bias into the mix. However, in our human-in-the-loop method, it is easy to identify where such biases may arise and mitigate them through validation approaches borrowed from qualitative research. Specifically, there

are four steps in our method where human bias can be inserted into the analysis. Here, we discuss our recommendations for how to mitigate this risk.

First, during the data-processing step, when selecting the relevant actions, there is a chance that bias will emerge due to relevant actions being overlooked. To mitigate this, we recommend having a second stakeholder of equal knowledge and expertise conduct this task in collaboration with the first. By having multiple people identify the relevant actions separately and then compare their results, measures of percent agreement can be used to determine the validity of the selection. Disagreements can also be discussed and resolved. Depending on the complexity of the learning environment and the range of afforded actions, additional stakeholders may also take part. The identities of these additional stakeholders may vary depending on available human resources, and they may be other educators or teaching assistants.

Second, during the visualization step, when the actions are grouped into categories, there is a risk that the inferences regarding the learning strategies each category represents are incorrect. To mitigate this risk, we recommend member checking with another person who is familiar with the learning environment, ideally with students, who could confirm whether or not the classifications are accurate inferences. If the classifications are deemed inaccurate, or the result of bias from the stakeholder, student input can be used to update them.

Third, during the pattern identification step, we recommend pursuing analysis in an approach similar to inter-rater reliability (IRR). In such an approach, two or more stakeholders, likely an educator and either other educators or teaching assistants, separately analyze the DTW output and identify patterns. They then reconvene to compare their findings. If necessary, Cohen's kappa or a similar measure of reliability can be used and disagreements resolved through discussion until a comprehensive set of patterns is identified.

Fourth, during the final analysis of learning processes, we recommend pursuing analysis following an IRR approach. Again, bias would be mitigated by having separate individuals independently analyze the data and compare the clusters and sequences to identify the student learning processes. Following the IRR approach, a kappa measurement can be used to determine the reliability of the findings, and discussion can be used to resolve disagreements.

4. Case Study: Parallel

To illustrate our approach, we present a case study in which we analyzed student action sequences from a learning game. In this section, we introduce the game, outline the study setup, illustrate the implementation of the method, and discuss the results.

4.1 Parallel: A Game-Based Learning Environment

We chose a learning game due to its demonstrated value in helping students understand new concepts through learn-by-doing experiences (Adams et al., 2012) and because of recent interest in bringing data-driven learning into gameplay (Zhu & El-Nasr, 2021; Villareale et al., 2020).

Parallel is a single-player game-based learning environment that teaches concurrent and parallel programming concepts (Zhu et al., 2019; Kantharaju et al., 2018; Zhu et al., 2020). It does this through 2D puzzles that challenge the player to coordinate arrows, which represent threads, to pick up and deliver packages without interfering with each other. The arrows move along predetermined tracks at random speeds, and, to accomplish this goal, players must use signals and semaphores to control the arrows' movements. This can be done by placing graphical components representing each element on the track. A player wins a level by placing the right number of elements in the correct locations. Through continued gameplay across multiple levels, the game introduces students to concepts such as non-determinism, synchronization, and efficiency.

For this study, a special Parallel build was created containing four levels, where the first three were tutorial levels introducing the game and its mechanics, as well as the parallel programming concepts of non-determinism, multi-threaded processing, critical sections, and race conditions. Level 4, as seen in Figure 2, was unguided, where players practised what they had learned in the previous levels. Level 4 was used as the subject of the case study analysis because it was the first opportunity for players to practise what they learned without guidance.

4.2 Recruitment

Thirteen students were recruited from an undergraduate computer science program. Participants were required to be (1) 18 years of age or older, (2) located in the United States, (3) able to communicate in written and spoken English, and (4) able to play on a Windows machine. We also ensured that participants had not played Parallel before. No other inclusion or exclusion criteria were applied, and players were not required to have any knowledge of parallel programming. Due to the COVID-19 pandemic, this study was carried out remotely. Upon giving informed consent, players were sent detailed instructions on how to download and run Parallel and locate their data logs, which were sent to the research team over an encrypted file-sharing service, along with screen recordings of their gameplay. Gameplay took from 30 to 60 minutes. Participants received a $50 gift card for their time.

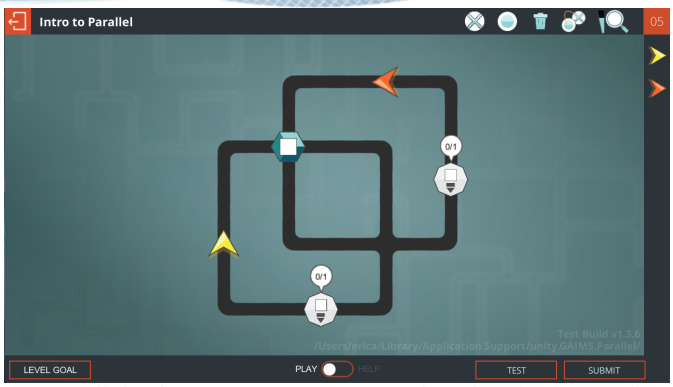

Figure 2. A screenshot of the game Parallel. The two arrows represent threads. Players need to place semaphores and signals to direct arrows, which carry packages and move along predefined tracks, to the designated delivery points. The pictured level is the one that was analyzed for this case study.

4.3 Procedure

For the purposes of this case study, the stakeholder was considered to be an educator teaching parallel programming. However, the role of the stakeholder was played by members of Parallel's design and research teams.

4.3.1 Step 1: Data Processing

Parallel was instrumented in prior work such that a player's actions, including everything from where elements were placed to mouse clicks, were recorded in the log files. This resulted in a great deal more information than what was needed to analyze learning. Thus, the research team worked collaboratively and used knowledge of the game's design and iterative conversations with its designer to identify a set of actions deemed important for recognizing students' learning processes (see Table 1). The logs were then analyzed to identify the corresponding logged events. For example, "place semaphore" (a gameplay action) corresponded to a pair of logged events: "mouse click down" (over semaphore icon in menu) and "mouse click up" (over a track). Once connections were identified, a Python script was used to process each log into a sequence of actions by abstracting the events into the gameplay actions they represent. Logged events that did not correspond to one of the selected gameplay actions were filtered out.

Table 1. The 13 Game Actions Identified to Indicate Learning or Concept Understanding*

Action	Definition
Place semaphore	The player places a semaphore on the board.
Place signal	The player places a signal on the board.
Link signal and semaphore	The player links a signal and a semaphore.
Test passed	The player runs a test and it passes.
Test failed	The player runs a test and it fails.
Stop test	The player stops a test simulation before it completes.
Toggle semaphore	The player locks or unlocks a semaphore.
Move semaphore	The player moves a semaphore to another spot.
Move signal	The player moves a signal to another spot.
Destroyed semaphore	The player destroys a semaphore.
Destroyed signal	The player destroys a signal.
Submission passed	The player submits a solution and it passes.
Submission failed	The player submits a solution and it fails.

*Each log was processed into a sequence comprising some subset of these actions.

4.3.2 Step 2: Visualization

For our case study, we chose the visualization system Glyph (Nguyen et al., 2015) because it meets all of the requirements and the researchers were familiar with the system. However, Glyph is not required to implement this method, and any visualization that meets the requirements stated above would suffice.

Glyph's interface consists of two views: a state graph and a cluster graph. The state graph (Figure 3, right) is a node-link representation of student behaviour in which each node represents an action that can be taken within the environment. A link

between two nodes indicates that at least one student's sequence transitions between these two actions, and the thickness of that link represents the number of students. Repeated actions of the same type are represented by the same node, and the student loops between nodes as they repeatedly perform an action. For example, the sequence of (1) read instructions, (2) answer a question, and (3) read instructions again is represented by two nodes: "read instructions" and "answer a question." The links would loop from "read instructions" to "answer a question" and then back to "read instructions."

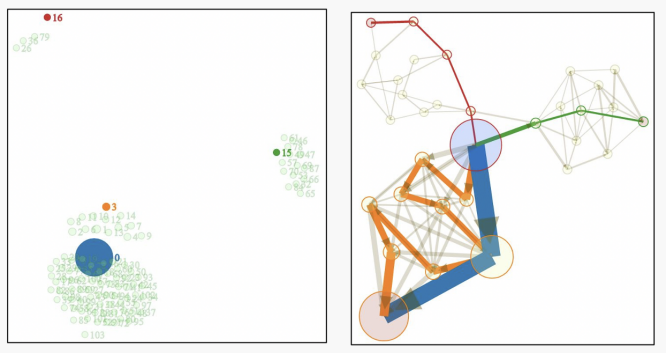

Figure 3. The two graphs of Glyph. On the left is the cluster graph, where the larger node indicates a common sequence and the smaller nodes indicate less common sequences. Distances between nodes indicate similarity. On the right is the state graph. The user has highlighted four different sequences, and the corresponding nodes and links are highlighted as well.

Glyph users can click and drag any node in the state graph to change its position on the screen. The links that connect the node to others will automatically adjust their lengths and angles accordingly. Additionally, selecting any given node in the cluster graph will highlight the sequence of nodes and links in the state graph; multiple cluster graph nodes can be selected at once, as seen in Figure 3.

The cluster graph (Figure 3, left) represents each unique sequence as an individual node. The node's size indicates how many students share it, with larger nodes indicating more students exhibiting that sequence. These nodes are clustered based on the DTW algorithm (Nguyen et al., 2015; Berndt & Clifford, 1994), with similar nodes being near each other. DTW calculation are discussed in Nguyen and colleagues (2015) and Berndt and Clifford (1994). In the default algorithm, for every point in the two sequences in which the two actions are not the same, a value of one is added to the distance. For the initial visualization of the sequences, it is recommended to use this default calculation.

The sequences for all 13 players were uploaded to Glyph. For this case study, the expert trace was generated artificially and added to the visualization. For the expert trace, the researchers leveraged their knowledge of the game and the designer's input to identify the actions that would need to be taken to solve the level. It was decided that the expert trace would exhibit the least number of each of these essential actions in an order that is efficient and exhibits the behaviours the game is intended to teach. The resulting trace was *place semaphore, place signal, place semaphore, place signal, link signal and semaphore, link signal and semaphore, test passed, submission passed*.

The Glyph visualization was then arranged such that this trace was placed diagonally through the centre of the state graph, as can be seen in Figure 4(a). The remaining actions were grouped into three action categories based on knowledge of the game's design and review of its documentation:

- *Adjustment actions* are moves taken to adjust an existing solution (toggle semaphore, move semaphore, move signal).
- *Reset actions* are moves taken to clear a solution (destroy semaphore and destroy signal).
- *Information acquisition actions* are moves taken to get feedback or information (view help, stop submission, submission failed, test failed, and stop test).

The Glyph visualization was arranged such that the nodes belonging to each group were positioned near each other in the state graph. Arranging Glyph in this manner facilitated visual comparison of sequences by allowing researchers to see, quickly, how each sequence traversed the action categories before going into more detailed analysis. The resulting state graph, with the expert trace and action categories highlighted, can be seen in Figure 4(b).

4.3.3 Step 3: Identification of Student Patterns

The researchers then leveraged their domain expertise and analyzed the visualized sequences following the steps outlined in Section 3.4. Specifically, following a protocol similar to content analysis, the researchers identified patterns separately and then reconvened to iteratively discuss similarities and differences before settling on a final set. Based on this analysis, they identified

general action patterns that characterized each cluster produced by DTW. These patterns are outlined in Table 4 and discussed in Section 4.4.

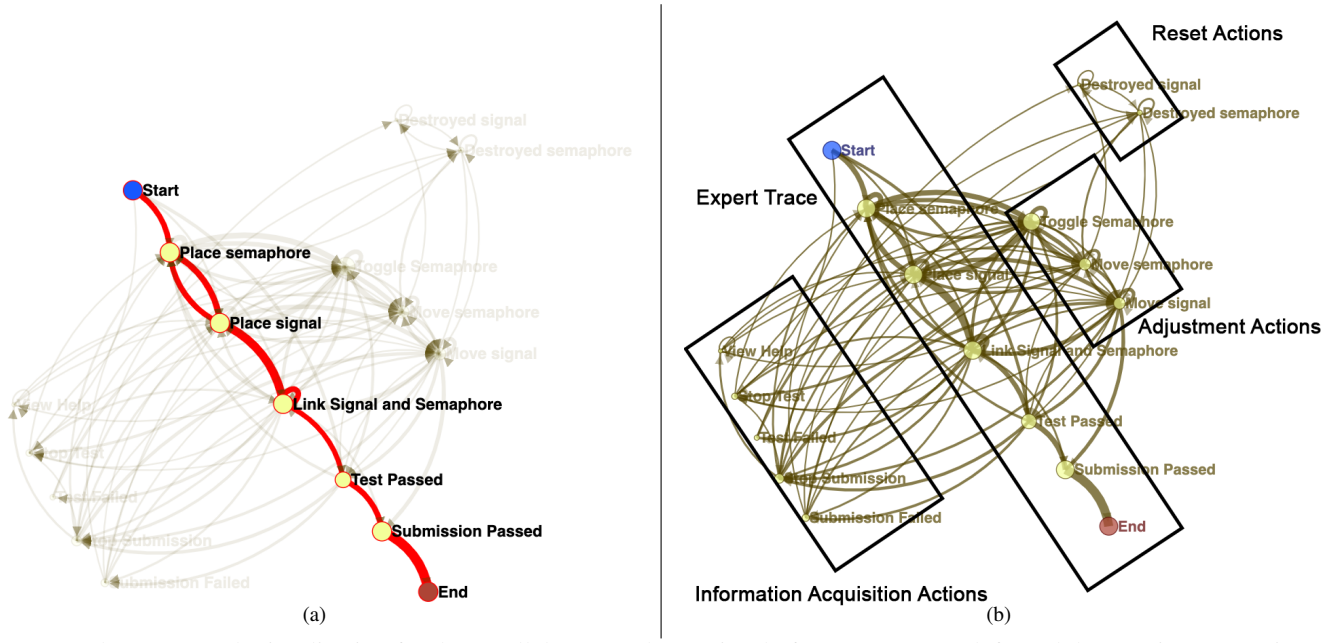

Figure 4. The state-graph visualization for the Parallel case study consisted of an expert trace (left) and three action categories: information acquisition actions, adjustment actions, and reset actions (right).

4.3.4 Step 4: DTW Update and Iteration

The cluster graph produced by the default DTW algorithm can be seen in Figure 5(a). Since no two students had the same sequence, all of the nodes in the cluster graph represent a single student, and all are the same size. The highlighted sequences in this graph indicate those identified to possess similar general action patterns, who should thus be clustered together by DTW. It was determined that the DTW algorithm was not adequately clustering the sequences based on these action patterns. Thus, following the steps outlined in Section 3, the researchers updated the DTW algorithm, specifically deciding to penalize action categories outside of the expert trace, depending on how likely an expert was to take actions in those categories.

The updated algorithm calculated the distances between the sequences based on how they interacted with the action categories that were identified in step 2. It did so as follows: for any given point in any two sequences, if the two actions at that point were the same, then a zero was added to the distance. Otherwise, a value was added based on what action categories the two actions belonged to. If they were in the same category, a value of one was added. If they were in different categories, a value between two and four was added accordingly. The value for each category pair can be seen in Table 2.

Table 2. Values for the Updated DTW Algorithm*

	Expert trace	Adjustment	Info acquisition	Reset
Expert trace	1	2	3	4
Adjustment		1	2	3
Info acquisition			1	2
Reset				1

*If the two actions were not the same, then a value was added to the distance based on what action categories they belonged to.

This produced an updated cluster graph that more closely matched the human's judgment of how the students should be grouped. At this point, it was determined that "test passed" should be considered an information acquisition action by DTW instead of an expert trace action (see Table 2). Passing a test is a form of information acquisition. It is considered a part of the expert trace, since Parallel teaches students to always test before submitting. However, it is essentially the only information acquisition action that an expert should take within the context of the given level. The DTW distances were calculated again with this additional adjustment, and the output, seen in Figure 5(b), produced satisfactory clusters. The state graph did not change during this process.

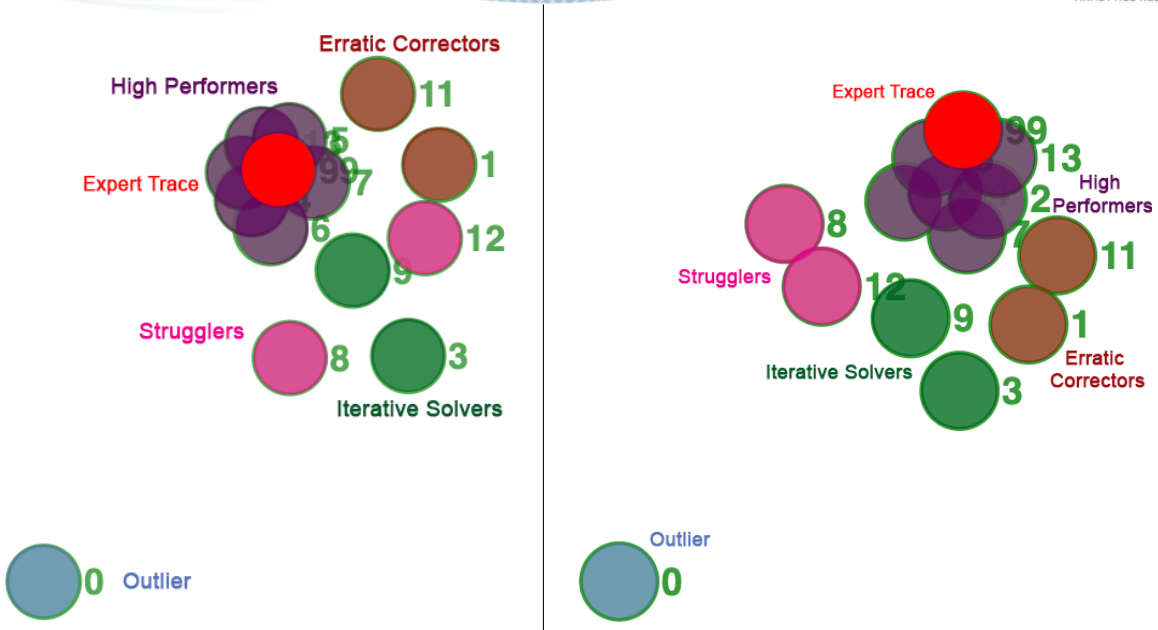

Figure 5. The initial cluster graph for the Parallel case study based on the default DTW algorithm (left) and the final cluster graph for the Parallel case study after the DTW algorithm was adjusted based on human insight (right). The colour-coding indicates the student clusters that were identified by the human analyst. Between the original graph and the updated graph, the DTW algorithm was updated to reflect how the human understood the data.

4.3.5 Step 5: Validation

Silhouette (Rousseeuw, 1987) was used to validate the five resulting clusters, and the results can be seen in Table 3. As can be seen from the resulting silhouette values, sequences 2, 4, 5, 6, 7, and 13 are all appropriately clustered, as indicated by their positive silhouette values, which are all greater than 0.4. These six sequences correspond to the largest and most prominent cluster, dubbed the "high performers," seen in Figure 5(b) and discussed in Section 4.4. Sequences 6 and 7, which have the lowest values of the six, are on the outer edges of the cluster, as can be seen in Figure 5(b). Likewise, sequence 0, which is notably far from any other sequence, as can be seen in Figure 5(b), is clustered appropriately on its own.

Table 3. The Silhouette Values Calculated for Each Student Sequence Using the DTW Distances and the Researcher-Derived Clusters*

Player	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P11	P12	P13
Silhouette	1	−0.06	0.52	0	0.557	0.605	0.449	0.453	0.056	−0.174	−0.179	−0.03	0.538

*Numbers closer to one indicate that the sequence is in the correct cluster, while numbers closer to zero indicate that the sequence is between clusters.

The remaining six sequences, however, are all between clusters according to the silhouette calculation, as indicated by their values, which are close to zero. The lack of a strong cluster for these points is due to the nature of the data and the abstraction used, which resulted in six sequences that do not exhibit prominent enough behavioural similarities or differences to be clustered. While statistically these points didn't form a coherent cluster, through qualitative analysis, expert insight was able to identify similarities between these sequences, and thus we retained these groupings.

4.3.6 Step 6: Granular Analysis of Learning Processes

The final visualization was then analyzed in detail to identify key characteristics and differences in student learning processes within and across groups. The results of this process are discussed in detail next.

4.4 Results

All 13 students successfully completed the level, but the paths they took varied. DTW sorted the 13 student-generated sequences into five clusters (see Table 4) based on notable characteristics in terms of what actions they took and in what order (see Figure

6(a)). Comparison of sequences permitted the identification of learning patterns and individual differences that can be used to identify points of struggle and design potential interventions.

Table 4. The Six Identified Sequence Groups and Their Defining Characteristics

Group	Defining Characteristic
High performers	Close to the expert trace but differ in terms of ordering and because they all exhibit at least one adjustment behaviour and, in two cases, one or two information acquisition behaviours
Erratic correctors	Perform information acquisition actions followed by a large number of actions, with no obvious logical progression, before reaching the solution
Iterative solvers	Pass or stop tests and submissions multiple times, often followed by minimal changes and then additional tests or submissions
Strugglers	Exhibit multiple characteristics in random and inconsistent manners, along with visits to reset actions
The outlier	A single sequence isolated from the rest

4.4.1 Group 1: The High Performers

The first and largest group contained six student sequences, dubbed "high performers" because they were characterized as being close to the expert trace, but differed from the expert trace in terms of ordering of actions and because they all exhibited at least one, but never more than two, adjustment actions and, in two cases, a single information acquisition action. The sequences belonging to this group can be seen in Figure 6(b). As high performers, these students do not require extensive intervention from an educator. However, we can still glean insights into how they are performing by comparing their sequences against one another and the expert trace.

Four of these traces differ from the expert trace by visiting adjustment nodes (moving or toggling elements), and, thus, it was possible to conclude that these students may be a bit less certain of the exact ordering of elements than an expert would be. However, this is the only extra action category that these four sequences visit; that is, they still reach the solution in a single attempt, indicating that the students understood the concepts enough to recognize and correct errors on their own.

The remaining two traces differ from the expert trace in both visits to adjustment actions and information acquisition actions. Unlike the expert trace, the students did not initially have a correct solution and needed to make adjustments to reach a solution. However, unlike the previous four students, they did not realize that this was the case and needed the system to inform them through a failed submission. These sequences differ from sequences that come later because they only exhibited these behaviours once and because they arrived at the correct solution after a minimal number of corrective actions. This indicates that they understood their mistakes and how to correct them, implying that they are still high performers. Thus, they are clustered with this group. Of note is that these are sequences 6 and 7, which are on the edges of the cluster and had lower silhouettes, as can be seen in Table 3.

By comparing the sequences to each other, it was possible to identify some noteworthy individual differences. The most prominent observation was that three students began level 4 with a test (which passed for two and was stopped for the third). However, these students did not follow the test with a submission attempt, which may imply that they *do* understand non-determinism, or that a solution that works once will not automatically work every time. From here, the students converge with the rest of the group in terms of what actions are taken and in what general order, suggesting that the unusual testing action does not indicate a lack of understanding of learning concepts but perhaps a lack of understanding of the nature of the puzzle.

When comparing the two students who performed information acquisition activities, we found that one failed a submission immediately after passing a test, implying that they do *not* have a strong grasp of non-determinism. The second student attempted to solve the solution with only a single signal and semaphore pair, which was not possible. It may be that this student was thinking about the problem in terms of optimization and believed that only one pair should be necessary.

4.4.2 Group 2: Erratic Correctors

The next group consisted of two students whose sequences can be seen in Figure 6(c). Both sequences, after taking an information acquisition action, exhibited large numbers of seemingly undirected or random actions, implying that the students may not have properly understood how to get to a correct solution and did so randomly rather than through a plan or strategy. This characteristic is in sharp contrast to the expert trace, which did not include any information acquisition or adjustment actions. It is also in contrast to the high-performers group, who made minimal adjustments to reach a solution, with or without information acquisition. These minimal moves indicate an understanding of the error and a directed movement to correct it, whereas the erratic correctors' sequences indicate the opposite.

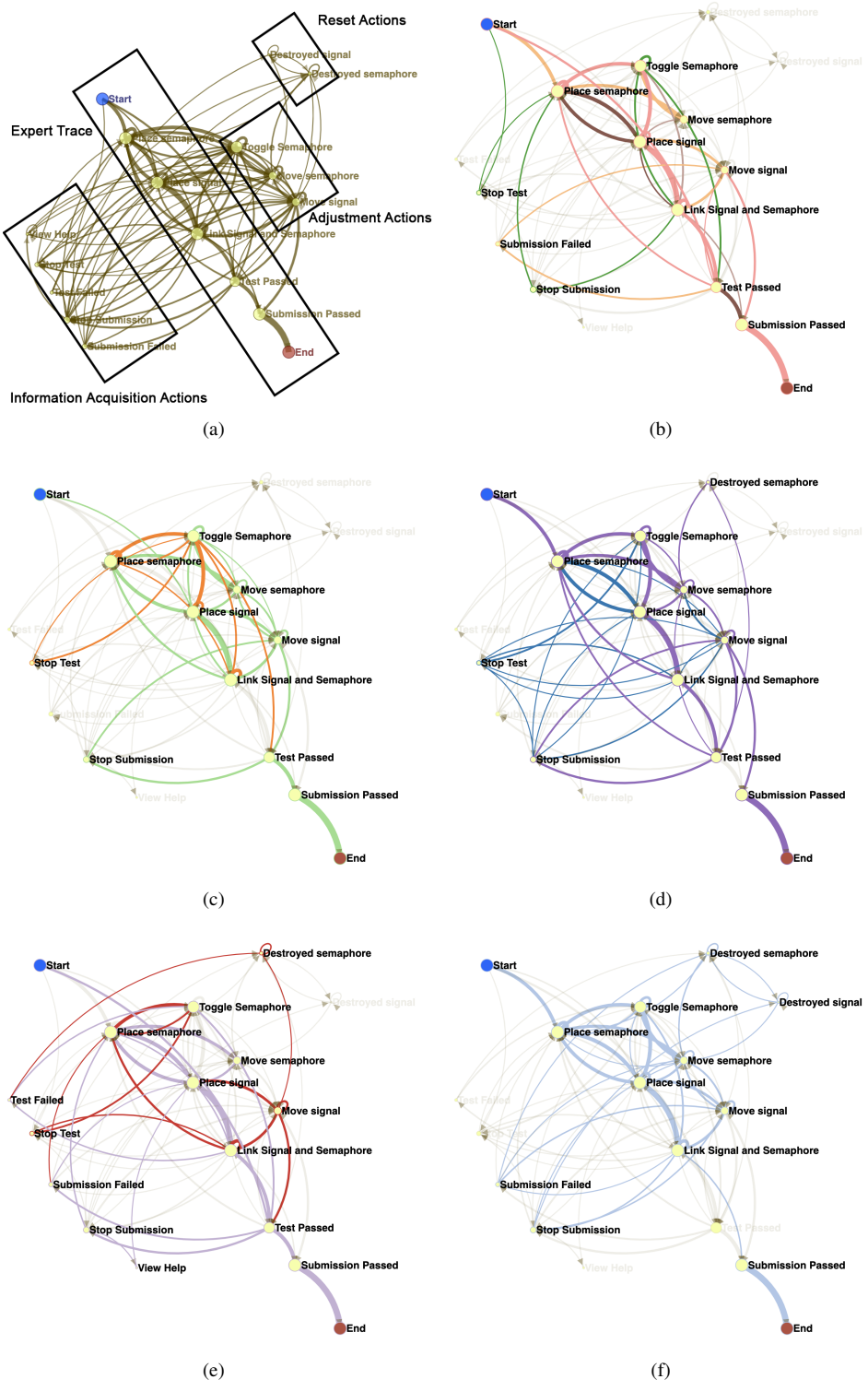

Figure 6. The five groups for the Parallel case study: (b) high performers, (c) erratic correctors, (d) iterative solvers, (e) strugglers, and (f) the outlier. A key indicating which actions belong to which category is shown in (a).

When the individual sequences were compared, further individual differences were identified. Both sequences began by placing the correct number of signals and semaphores and then ultimately stopped at a test or submission. At this point, one student moved their board elements (signals and semaphores) around seven times. This indicates that they understood the

number of elements they needed to solve the level (because they did not add more) but did not understand the placement. This may indicate that the student understands the logic behind parallel programming but struggles with the order of operations. By contrast, after stopping the test, the other student placed two more signal-semaphore pairs on the board and did not move any elements. This indicates that the student was confident in their placement but believed that more elements were needed to solve the level. This may imply that the student was struggling with understanding the functionality of semaphores, leading them to believe that more than necessary were needed to solve the puzzle.

4.4.3 Group 3: Iterative Solvers

The next group also consisted of two sequences (see Figure 6(d)) characterized by how they differ from other clusters in their overall approach. Specifically, these sequences passed and stopped tests and submissions multiple times. However, these actions were rarely followed by lengthy adjustments. Instead, they were often followed by minimal changes and then additional tests or submissions. This implies that the students used tests and submissions to get feedback on parts of the solution before building more. In other words, they were building their solution iteratively.

When compared against the expert trace and the other groups, the primary differences were the number of times information was sought and the number of actions that happened in between that seeking. The expert trace included no information acquisition, while all previous groups, including the erratic correctors, successfully reached the solution with minimal information acquisition actions. By contrast, the sequences in the iterative solvers group exhibited more information acquisition. Because they sought information more often, these sequences featured shorter bursts of adjustment actions than the erratic correctors, who did not test often and instead performed a high number of adjustments without seeking feedback. Looking closely at each sequence in this group permitted identification of individual differences. For example, one student passed tests multiple times. Each time, they added more to their solution or made a minor adjustment. The first time they passed a test, they had already placed elements on the board but had not created any pairs, but after passing the test, they created two pairs and then tested again. After passing this test, they moved an element three times and then tested again. This pattern of passing a test and making a small adjustment before testing again continued until they eventually reached the solution. The fact that this student never went from a passed test to a stopped or failed submission indicates (a) that they understand non-determinism and (b) that they did not have the entire, correct solution. This implies that their regular testing was not due to a lack of understanding but to gain feedback, which may imply that the student was not confident in their understanding of the problem and solution or perhaps needed a little extra help in seeing where they were wrong, despite already knowing they were wrong.

The other sequence is a bit different. This student moved from a passed test to a stopped submission, implying that they may not understand non-determinism or failed to recognize the errors in their solution. For example, they stopped a test, created a link, stopped another test, moved a signal, and then stopped another test. The presence of stopped, rather than passed, tests may indicate that this student is further from understanding the correct answer than the other student.

4.4.4 Group 4: Strugglers

The next group consisted of two sequences (see Figure 6(e)). Both included a number of the behavioural patterns exhibited by the previous groups but switched between them in a somewhat random manner. The lack of consistency and direction may imply that these students struggled to grasp the concepts taught by the game, recognize the solution, or develop a clear path to reach that solution.

When compared to the expert trace, there are notable divergences. Perhaps the most jarring is that both players passed a test with an empty board and then attempted to submit the empty board (which failed). This may imply that these players do not have a strong grasp of the concept of non-determinism, leading them to believe that the empty board, which passed once, would pass every time. When comparing them to the other groupings, it was noted that there was a lack of consistency to their approach, with both students alternating between small changes with regular information acquisition (similar to iterative solvers) and large changes with little information acquisition (similar to erratic correctors).

When compared against each other, one student built an entire solution (with both necessary pairs) and failed their next test. The student then alternated between small changes and subsequent stopped or failed tests and large, erratic changes with few tests. The other student followed their first failed submission with the creation of one pair, a second passed test, and a second failed submission. Their repetition of the passed test–failed submission sub-sequence indicates that they may not have a good understanding of the problem or how to solve it. They then entered into the same alternating pattern as the other player, with the primary difference being that they often passed their tests and always followed these with a failed submission.

4.4.5 Group 5: The Outlier

The final group consisted of only a single sequence (see Figure 6(f)). Although the student eventually reached the solution, it took them almost four times as many actions as other students. In addition, they exhibited behavioural patterns similar to those exhibited by the erratic correctors, though with far more submission attempts, with substantial, seemingly erratic, and undirected changes between submissions, which included frequent use of reset actions. They also never tested, suggesting

false confidence in their solution and an inability to recognize errors. This information implies that this student struggled to understand the concepts taught by the game but may not have been consciously aware of this struggle.

4.4.6 Summary and Takeaways

In summary, our method resulted in five distinct student clusters, each characterized by certain sequence patterns that indicate what the students are, or are not, understanding. Notably, our method produces information about the ordering of actions, not just the number of actions, which provides critical information about students' understanding of the problem and their approach to solving it. For example, with the high performers, knowing that students made adjustments only after failing tells us that they were confident in their understanding and only prompted to question their ideas by failure. Similarly, knowing that students immediately submitted after passing a test may indicate a lack of understanding of non-determinism. This information is critical to designing effective personalized learning because it allows the stakeholder to pinpoint exactly what concepts each student is struggling with and design targeted lesson plans. For example, in the case of the iterative solvers, the first student was closer to the solution and seemed to be able to reason their way through the problem with the aid of feedback from the game. This may imply that the student had a decent, though not perfect, grasp of parallel programming concepts but lacked confidence in their knowledge of the topic. The stakeholder could thus design a personalized lesson plan to help this student build confidence while further enhancing their already steady understanding of the subject matter. Pinpointing the student's areas for improvement would not be possible without a granular understanding of their problem-solving process, which is afforded by our method through interpretable SA.

5. Generalizability of the Approach

While we demonstrate the method in a single case study, we have been able to apply the same approach to several other digital and learning environments, such as E-sports (Ahmad et al., 2019), alternative reality games (Javvaji et al., 2020), and May's Journey, a puzzle game that teaches programming (Jemmali et al., 2020). In our experience working with this approach, data abstraction is key to ensuring that the data remains interpretable even at large scales. While our case study used the technique to examine only 13 students, the visualization methods we leverage have been demonstrated in previous work with significantly larger numbers (such as hundreds and even thousands of users or sessions) (Javvaji et al., 2020; Ahmad et al., 2019; Nguyen et al., 2015). In all cases, interpretability was maintained through an abstraction scheme that generalized lower-level actions into fewer higher-level actions and ensured that the visualized traces were manageable for a human stakeholder. Thus, it is reasonable to assume that the method presented here will also be scalable to large numbers of students and more complex levels (i.e., levels allowing for multiple solutions). However, we acknowledge that we have not empirically confirmed that with the current case study.

In particular, in future work, we aim to apply the approach to different and more complex levels and to bigger numbers of students. Expanding this analysis to more complex levels will reveal many opportunities for us to evaluate or investigate learning. In these cases, each possible correct solution would be an expert trace, and the analysis of the individual sequences would be conducted taking the multiple traces into account. For example, perhaps a group of students is clustered around one trace and a second group is clustered around the other. Perhaps the two expert traces represent two ways to solve the level that indicate certain learning gains versus others. Thus, conclusions can be drawn about students' learning and problem-solving strategies based on their relative proximity to the multiple expert traces. Alternatively, suppose the expert traces are all conceptually similar. In that case, they can be clustered together and treated as a sequence category, and individual traces and their categories can be analyzed based on proximity to the category as a whole.

In addition to using this approach within games, we can also see an application of this approach to other educational environments beyond games. Game-based learning environments are, essentially, made up of a chain of problems that students need to solve, making them well suited to the study of problem solving at a granular action-by-action level. That being said, other digital learning environments, such as MOOCs or LMSs, can also be developed and instrumented so that they can capture the same granular action-by-action problem-solving data. The granularity of the data, capturing a sequence of individual actions that a student takes while solving a problem, is the key element. The method is generalizable to any environment capable of facilitating the collection of such information. Even within a game-based environment, the method can be executed on a single level, or it can be executed across multiple levels, as long as the game has been instrumented appropriately and the stakeholder has developed a sufficient abstraction scheme.

6. FTSA: An Existing SA Technique

To better demonstrate the value of our approach, we compare it to the FTSA method proposed by Sawyer and colleagues (2018) and replicate their approach with data from the Parallel game. We selected this particular technique because, like our method, it

compares sequences based on their distance to an expert trace, creating interpretable clusters. However, unlike our method, it does not facilitate a human in the loop due to the lack of interpretable sequences and adjustable model, meaning there is no opportunity for a human analyst or stakeholder to provide input to the calculations. Further, we selected this method for comparison because it has been demonstrated specifically in the context of solving a game level. Thus, it has been used to explore problem solving at a granular action-to-action level within a goal-based interactive task.

FTSA leverages lock-step Euclidean distance (Ding et al., 2008) to compare gameplay sequence data (referred to as *trajectories*) between the players in a pair. Specifically, Sawyer and colleagues (2018) calculated the lock-step Euclidean distance between each student trajectory and an expert trajectory. They demonstrated that students with trajectories further from the experts had lower learning gains.

We chose this particular method for comparison due to its similarity with our work; that is, it uses actions, it compares sequences based on a distance measure, it emphasizes the comparison of each sequence to an expert trace, and it has been demonstrated in the context of a game-based learning environment. However, unlike our method, it is a statistical technique with little to no human input in the process and results that are more difficult to interpret.

6.1 The FTSA Method

Sawyer and colleagues (2018) consider the cumulative counts of student in-game actions as the components of a multi-dimensional action vector $\mathbf{x}$. The action vector $\mathbf{x}$ depends on time, and the authors use a time interval of 10 seconds for cumulative counts. Thus, for every player, at $t = 0$ the action vector $\mathbf{x}$ is a zero vector.

Principal component analysis (PCA) (Abdi & Williams, 2010) is used to convert the cumulative counts of actions into a single value describing student progress until a particular moment in time. The authors justify the use of PCA due to the correlations between cumulative action counts (the components of the action vector $\mathbf{x}$) at specific time intervals. Lowering the dimensions used in the calculations reduces noise in the distance measurements between a student and an expert trajectory.

The authors define filtration as a function, f, that converts the multi-dimensional action vector $\mathbf{x}$ to a scalar value, c, using the first principal component vector, $\mathbf{p}$. This function is shown in Equation (1) for cumulative action vector $\mathbf{x}$ of player i at time t,

$$f(\mathbf{x}_i^t) = (\mathbf{x}_i^t)^T \mathbf{p} = c_i^t. \tag{1}$$

Thus, the cumulative action vector $\mathbf{x}$ is reduced to a scalar value c_i^t for each 10-second time interval (Figure 7), generating a univariate time series for each player. The distance d_{ij} between two players i and j is the average Euclidean distance between their filtered time series over all time steps:

$$d_{ij} = \frac{1}{n} \sum_{t=1}^{n} \|c_i^t - c_j^t\|_2. \tag{2}$$

The value n is calculated as $\max(n_i, n_j)$, where n_i and n_j are the lengths of the two time series. Moreover, the shorter series is padded to the length of the longer series by repeating the final filtered value of that series.

In summary, the lock-step Euclidean distance is calculated in two steps. First, the gameplay data is filtered through PCA by converting a multivariate time series to a univariate time series. Second, the univariate time series generated by the PCA is leveraged to calculate the lock-step Euclidean distance.

6.2 FTSA Results for Parallel

In Parallel, we create a 13-dimensional vector representing the 13 actions of Parallel gameplay. Each vector dimension is calculated by the cumulative action counts of a user's gameplay. The principal components are calculated on the final action counts of each player (not including the expert trace), and the first principal component projects the cumulative action vectors onto a single dimension. We report the summary statistics of the PCA in Table 5.

The output of the FTSA method for Parallel, generated using the same 13 players as we used for our analysis above, is seen in Table 6. The "ET" stands for the expert trace. Each value in the table is the Euclidean distance, as discussed in Equation (2).

Based on the work of Sawyer and colleagues (2018), the distance between a student's trace and an expert trace is correlated with a student's learning gains and may be used to identify struggling students. Thus, by referencing the distances from the expert trace, we can identify those with notably large distances, such as player 1, as students who require assistance in their learning.

To validate the accuracy of the FTSA output, we enlisted the aid of a domain expert who is not from our research group and who is experienced in parallel programming and familiar with the Parallel game. The expert watched screen recordings of each

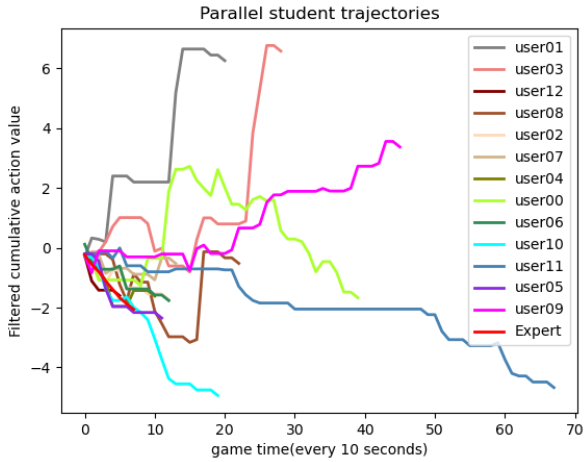

Figure 7. "Action trajectory" of the players using the FTSA method. The scalar values (y-axis) at a 10-second time interval (x-axis) represent the filtered cumulative action vector using the first principal components.

Table 5. Summary Statistics of the Principal Component Used to Filter Cumulative Action Vector of Players in Parallel

Gameplay Action	First Principal Component	Mean (SD)	Expert Trace Cumulative Action Vector
Place signal	−0.312	2.923 (1.639)	2
Link signal and semaphore	−0.281	3.538 (2.763)	2
Place semaphore	−0.242	3.308 (2.398)	2
Testing	−0.202	1.615 (1.273)	1
Submission	−0.187	1.769 (1.25)	1
Toggle semaphore	−0.087	3.615 (3.913)	0
View help	−0.032	0.077 (0.266)	0
Destroy signal	0.005	0.462 (1.599)	0
Destroy semaphore	0.047	0.923 (2.401)	0
Stop submission	0.096	0.923 (1.269)	0
Stop test	0.119	0.615 (1.146)	0
Move semaphore	0.342	2.231 (3.262)	0
Move signal	0.734	3.154 (3.57)	0

Table 6. The Output of the FTSA Method Run with the Data from the Parallel Game*

Player	P5	P4	P13	P12	P2	P6	P8	P7	P11	P0	P9	P3	P1
d_{iET}	0.077	0.127	0.13	0.138	0.147	0.156	0.238	0.292	0.381	0.439	0.473	0.731	1.287

*Each table entry corresponds to the Euclidean distance d_{iET} between player i in a column to the expert trace (ET). The players are organized based on their calculated distances, with the sequence most similar to the ET being furthest to the left.

player's gameplay and ranked each player based on how near to or far from an expert's action trajectory they deemed their actions to be. The resulting ranking can be seen in Table 7. The expert judged distance based on whether or not the players appeared to have an idea or plan for how to solve the level, as inferred by the actions they took and the order they took them in. They specifically identified players 13, 7, 5, 4, and 2 as taking deliberate action as if they had an idea, while players 12, 9, and 0 took action as if they were guessing and checking.

Table 7. A Comparison of How Each Method Ranked Each Sequence's Distance from the Expert Trace and How an Expert Ranked Them Based on Gameplay Videos

Expert ranking	P13	P7	P5	P4	P2	P6	P3	P8	P1	P11	P12	P9	P0
Our method's ranking	P13	P5	P2	P4	P7	P6	P9	P11	P1	P12	P8	P3	P0
FTSA ranking	P5	P4	P13	P12	P2	P6	P8	P7	P11	P0	P9	P3	P1

Complete sequence (30 states): Place signal(1), Place semaphore(1), Move semaphore(1), Toggle Semaphore(2), Move semaphore(3), Place signal(1), Link Signal and Semaphore(1), Place semaphore(1), Move semaphore(1), Place signal(1), Move signal(1), Link Signal and Semaphore(1), Toggle Semaphore(1), Move signal(3), Test Passed(1), Stop Submission(1), Move signal(3), Move semaphore(2), Move signal(2), Test Passed(1), Submission Passed(1)

Figure 8. The sequence for student 1, who was furthest from the expert trace according to the FTSA results.

Similar to the FTSA output, the expert identified players 5, 4, 13, and 2 as being close to expert play, although the ordering was somewhat different. However, the expert identified player 7 as being close to an expert's actions, while FTSA did not. Similarly, FTSA identified player 12 as being close to expert play, while the expert explicitly stated that player 12 was far from expert play.

6.3 Comparison

In this section, we compare FTSA to our technique and highlight the advantages of facilitating a human in the loop. Previous work has demonstrated that FTSA can identify students with lower learning gains. However, the statistical method does the calculations without any input from the stakeholder. As a result, if there is an error in the comparison, it is difficult for the stakeholder to identify or correct it.

By contrast, our technique leverages stakeholder input to fine-tune the DTW algorithm. Thus, the stakeholder is aware of how the model understands the data and can guide or correct it toward a result that more closely matches their own understanding. In order to enable the human in the loop to take such actions, our technique leverages visualization, which makes the sequences transparent and interpretable so that the stakeholder can make informed decisions when updating the algorithm. The interpretable sequences, however, additionally allow the stakeholder to understand the details of how a given student's sequence differed from the expert trace, which provides a stronger understanding of what the student was struggling with and permits more effective personalized learning.

By contrast, FTSA does not depict the sequences, only the calculated distances between them and the expert trace. As a result, it is impossible to know *how* the student sequence differs from the expert trace based on the results shown in Table 6. In other words, FTSA provides no insight into what the student is struggling with, which is necessary to design an appropriate and effective personalized learning environment.

To illustrate the value of the sequence information that FTSA does not capture, we examine the sequences of the student who is furthest from the expert trace according to the FTSA results: student 1. As seen in Figure 8, student 1's sequence reveals several placement and movement actions, too many signals at the time of submission, an unsuccessful submission after a passed test, and several adjustments made afterwards before successfully submitting. This result implies that this student did not have a concrete understanding of the problem they were trying to solve. They did not appear to understand non-determinism (they followed a successful test with an unsuccessful submission). They adjusted multiple pieces around the board before testing again, indicating that they may not have recognized the source of their mistake. This detailed information regarding the student's problem-solving process is critically important to addressing their needs as a learner and is absent from the FTSA results.

With the human-in-the-loop approach, this information can be used to adapt the model to ensure that its output more closely matches the stakeholder's understanding. This is illustrated by the expert order that was done to validate the FTSA method. The six sequences ranked closest to expert actions by the expert, seen in Table 7, correspond to the six sequences in the high-performers cluster according to our method. Similarly, player 0 was identified by the expert as furthest from an expert's actions and is singled out as an outlier by our method, but this player was not ranked furthest by the FTSA method. This indicates that our method, which enables a human in the loop to insert their knowledge into the learning model, results in a model that better aligns with how experts would understand the data. Although there are some differences in ordering between our method and the expert's ranking, these are likely the result of the researchers who used our method prioritizing different actions than the expert who analyzed the videos. Because a human was directly involved in generating our method's ranking, these differences could be identified and resolved through discussion and iteration.

By contrast, the FTSA approach is a statistical technique, and thus human analysts can't control the output or embed their own expertise within the system. Thus, the disagreements between the expert's ranking and the FTSA ranking cannot be discussed or resolved. While human-set parameters are a standard of statistical methods, our technique, specifically, leverages visualization to allow stakeholders to make informed decisions about how these parameters should be set, which FTSA does not allow.

7. Limitations

We acknowledge several limitations of our work. First, our approach looks only at actions and disregards states, which is different from other approaches, such as Bayesian networks. States, which could include such information as the layout of the

board when a particular action was taken, could provide additional context that would be valuable to understanding the students' intentions. Thus, our approach may be limited due to the loss of context that would be provided by the state information. This limitation can be addressed by abstracting the data to include both states and actions, and such abstraction will be explored in future work.

Second, we acknowledge that our approach takes more time than a purely statistical technique. However, as demonstrated through our analysis, we argue that the human-in-the-loop approach provides greater interpretability and more actionable insight than a purely statistical technique. Further, while analysis of gameplay videos would result in the greatest amount of contextually derived insight, we argue that such an approach would take significantly longer and not easily provide a holistic view of the data, since each video would need to be analyzed individually. Thus, we position our technique as a mix of quantitative and qualitative that aims to bridge the gap between the benefits of the two.

Third, we acknowledge that, in its current form, our method requires the educator to work with someone who has algorithmic experience to make adjustments to the algorithm. However, because we present this method in a tool-agnostic manner, we argue that one could reasonably create an interface that allows a stakeholder to make the DTW adjustments through GUI elements, eliminating the need to work with a programmer. The development and evaluation of such an interface is the subject of future work, and the contribution of this work is to present the method itself in a tool-agnostic manner.

Finally, we acknowledge that the current procedures for bias mitigation rely on additional people who may not have much available time. We are considerate of the time of educators and academics and acknowledge this as a practical limitation of this method. In future work, we hope to explore options for mitigating bias without consuming more of the educator's time than necessary.

8. Conclusion

SA, specifically sequence clustering and comparison, is a promising avenue to the process-focused study of student problem solving, informing more effective personalized learning. The problem, however, is that many SA techniques cannot be interpreted, so that it can be difficult for a stakeholder to understand the data or how to act on the results. Further, without an understanding of the data, it can be difficult for a stakeholder to infer how an algorithm or statistical method is understanding the data or why a statistical technique resulted in what it did. This, in turn, makes it difficult to critique or adjust the calculations.

In this paper, we presented a human-in-the-loop approach to SA. As demonstrated through the case study in Section 4, through visualization of sequences, our method allows a stakeholder to analyze the sequences and develop their own understanding of the data. Further, through the synchronization of the clusters and sequences in Glyph, the stakeholder can also build an understanding of how the algorithm has understood the data. These two understandings can then allow them to determine if the algorithm produces accurate clusters and correct and adjust it accordingly.

This human-in-the-loop process results in a scenario where the stakeholder better understands the resulting model because they had a direct hand in creating it and are, therefore, better able to act upon it in an informed manner. We argue that such an approach is beneficial to learning environments in general but is especially so in the context of online and hybrid learning, where it allows stakeholders to understand student learning in detail even when they are unable to observe the students directly. While we acknowledge that the time cost of our method compared to a purely statistical or automated technique may render it inappropriate in certain contexts, we believe that the benefits make a strong case for its use.

Acknowledgements

We would like to thank all members of the Parallel research and development teams for their hard work and contribution to this work. We would also like to thank Danial Hooshyar for his valuable input into our work.

Declaration of Conflicting Interest

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The publication of this article received financial support from the National Science Foundation under Grant Number 1917855, and the authors would like to thank all current members of the project.

References

Abbott, A., & Tsay, A. (2000). Sequence analysis and optimal matching methods in sociology: Review and prospect. *Sociological Methods & Research*, *29*(1), 3–33. https://doi.org/10.1177/0049124100029001001

Abdi, H., & Williams, L. J. (2010). Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, *2*(4), 433–459. https://doi.org/10.1002/wics.101

Adams, D. M., Mayer, R. E., MacNamara, A., Koenig, A., & Wainess, R. (2012). Narrative games for learning: Testing the discovery and narrative hypotheses. *Journal of Educational Psychology*, *104*(1), 235–249. https://doi.org/10.1037/a0025595

Ahmad, S., Bryant, A., Kleinman, E., Teng, Z., Nguyen, T.-H. D., & Seif El-Nasr, M. (2019). Modeling individual and team behavior through spatio-temporal analysis. *Proceedings of the Annual Symposium on Computer-Human Interaction in Play* (CHI PLAY 2019), 22–25 October 2019, Barcelona, Spain (pp. 601–612). ACM. https://doi.org/10.1145/3311350.3347188

Baker, R. S., Corbett, A. T., & Wagner, A. Z. (2006). Human classification of low-fidelity replays of student actions. *Proceedings of the Educational Data Mining Workshop at the Eighth International Conference on Intelligent Tutoring Systems,* 26–30 June 2006, Jhongli, Taiwan (pp. 29–36). https://educationaldatamining.org/ITS2006EDM/baker.pdf

Bakhshinategh, B., Zaiane, O. R., ElAtia, S., & Ipperciel, D. (2018). Educational data mining applications and tasks: A survey of the last 10 years. *Education and Information Technologies*, *23*(1), 537–553. https://doi.org/10.1007/s10639-017-9616-z

Balakrishnan, G., & Coetzee, D. (2013). *Predicting student retention in massive open online courses using hidden Markov models* (Technical Report No. UCB/EECS-2013-109). Electrical Engineering and Computer Sciences, University of California at Berkeley. https://www2.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-109.pdf

Berndt, D. J., & Clifford, J. (1994). Using dynamic time warping to find patterns in time series. *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining* (AAAIWS'94), 31 July–1 August 1994, Seattle, Washington, USA (pp. 359–370). ACM. https://dl.acm.org/doi/10.5555/3000850.3000887

Biswas, G., Jeong, H., Kinnebrew, J. S., Sulcer, B., & Roscoe, R. (2010). Measuring self-regulated learning skills through social interactions in a teachable agent environment. *Research and Practice in Technology Enhanced Learning*, *5*(2), 123–152. https://doi.org/10.1142/S1793206810000839

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. *Proceedings of the Eighth International Conference on Learning Analytics and Knowledge* (LAK 2018), 7–9 March 2018, Sydney, Australia (pp. 41–50). ACM. https://doi.org/10.1145/3170358.3170409

Boumi, S., & Vela, A. (2019). Application of hidden Markov models to quantify the impact of enrollment patterns on student performance. *Proceedings of the 2019 Conference on Educational Data Mining* (EDM 2019), 2–5 July 2019, Montréal, Québec, Canada. https://drive.google.com/file/d/1B8tZ8BIDrx-mg840dDiemY_DybiRIt7n/view

de Klerk, S., Veldkamp, B. P., & Eggen, T. J. (2015). Psychometric analysis of the performance data of simulation-based assessment: A systematic review and a Bayesian network example. *Computers & Education*, *85*, 23–34. https://doi.org/10.1016/j.compedu.2014.12.020

Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., & Keogh, E. (2008). Querying and mining of time series data: Experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, *1*(2), 1542–1552. https://doi.org/10.14778/1454159.1454226

Doleck, T., Basnet, R. B., Poitras, E. G., & Lajoie, S. P. (2015). Mining learner–system interaction data: Implications for modeling learner behaviors and improving overlay models. *Journal of Computers in Education*, *2*(4), 421–447. https://doi.org/10.1007/s40692-015-0040-3

Eichmann, B., Greiff, S., Naumann, J., Brandhuber, L., & Goldhammer, F. (2020). Exploring behavioural patterns during complex problem-solving. *Journal of Computer Assisted Learning*, *36*(6), 933–956. https://doi.org/10.1111/jcal.12451

Gasevic, D., Jovanovic, J., Pardo, A., & Dawson, S. (2017). Detecting learning strategies with analytics: Links with self-reported measures and academic performance. *Journal of Learning Analytics*, *4*(2), 113–128. https://doi.org/10.18608/jla.2017.42.10

Geigle, C., & Zhai, C. (2017). Modeling MOOC student behavior with two-layer hidden Markov models. *Proceedings of the Fourth ACM Conference on Learning @ Scale* (L@S 2017), 20–21 April 2017, Cambridge, Massachusetts, USA (pp. 205–208). AMS. https://doi.org/10.1145/3051457.3053986

Ha, E. Y., Rowe, J. P., Mott, B. W., Lester, J., Sukthankar, G., Goldman, R., Geib, C., Pynadath, D., & Bui, H. (2014). Recognizing player goals in open-ended digital games with Markov logic networks. In G. Sukthankar, C. Geib, H. H. Bui, D. Pynadath, & R. P. Goldman (Eds.), *Plan, Activity and Intent Recognition: Theory and Practice* (pp. 289–311). Morgan Kauffman. https://dl.acm.org/doi/10.5555/2671144

He, Q., Borgonovi, F., & Paccagnella, M. (2021). Leveraging process data to assess adults' problem-solving skills: Using sequence mining to identify behavioral patterns across digital tasks. *Computers & Education*, *166*, 104170. https://doi.org/10.1016/j.compedu.2021.104170

Hicks, D., Eagle, M., Rowe, E., Asbell-Clarke, J., Edwards, T., & Barnes, T. (2016). Using game analytics to evaluate puzzle design and level progression in a serious game. *Proceedings of the Sixth International Conference on Learning Analytics and Knowledge* (LAK 2016), 25–29 April 2016, Edinburgh, UK (pp. 440–448). ACM. https://doi.org/10.1145/2883851.2883953

Hooshyar, D., Pedaste, M., Saks, K., Leijen, Ä., Bardone, E., & Wang, M. (2020). Open learner models in supporting self-regulated learning in higher education: A systematic literature review. *Computers & Education*, *154*, 103878. https://doi.org/10.1016/j.compedu.2020.103878

Horn, B., Hoover, A. K., Barnes, J., Folajimi, Y., Smith, G., & Harteveld, C. (2016). Opening the black box of play: Strategy analysis of an educational game. *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play* (CHI PLAY 2016), 16–19 October 2016, Austin, Texas, USA (pp. 142–153). ACM. https://doi.org/10.1145/2967934.2968109

Iske, S. (2008). Educational research online: E-learning sequences analyzed by means of optimal-matching. *Proceedings of EdMedia+ Innovate Learning,* 30 June 2008, Vienna, Austria (pp. 3780–3789). Association for the Advancement of Computing in Education (AACE). https://www.learntechlib.org/primary/p/28909/

Javvaji, N., Harteveld, C., & Seif El-Nasr, M. (2020). Understanding player patterns by combining knowledge-based data abstraction with interactive visualization. *Proceedings of the Annual Symposium on Computer-Human Interaction in Play* (CHI PLAY 2020), 2–4 November 2020, online. ACM. https://doi.org/10.1145/3410404.3414257

Jemmali, C., Kleinman, E., Bunian, S., Almeda, M. V., Rowe, E., & Seif El-Nasr, M. (2020). MAADS: Mixed-methods approach for the analysis of debugging sequences of beginner programmers. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (SIGCSE 2020), 11–14 March 2020, Portland, Oregon, USA (pp. 86–92). https://doi.org/10.1145/3328778.3366824

Jeong, H., Gupta, A., Roscoe, R., Wagster, J., Biswas, G., & Schwartz, D. (2008). Using hidden Markov models to characterize student behaviors in learning-by-teaching environments. In B. P. Woolf, E. Aïmeur, R. Nkambou, & S. Lajoie (Eds.), *Proceedings of the Ninth International Conference on Intelligent Tutoring Systems* (ITS 2008), 23–27 June 2008, Montréal, Québec, Canada (pp. 614–625). https://doi.org/10.1007/978-3-540-69132-7_64

Kantharaju, P., Alderfer, K., Zhu, J., Char, B., Smith, B., & Ontanón, S. (2018). Tracing player knowledge in a parallel programming educational game. *Proceedings of the 14th Artificial Intelligence and Interactive Digital Entertainment Conference,* 13–17 November 2018, Edmonton, Alberta, Canada (pp. 173–179). https://ojs.aaai.org/index.php/AIIDE/article/view/13038

Kinnebrew, J. S., & Biswas, G. (2012). Identifying learning behaviors by contextualizing differential sequence mining with action features and performance evolution. *Proceedings of the Fifth International Conference on Educational Data Mining* (EDM 2012), 19–21 June 2012, Chania, Greece (pp. 57–64). https://educationaldatamining.org/EDM2012/uploads/procs/EDM_2012_proceedings.pdf

Kinnebrew, J. S., Loretz, K. M., & Biswas, G. (2013). A contextualized, differential sequence mining method to derive students' learning behavior patterns. *Journal of Educational Data Mining*, *5*(1), 190–219. https://doi.org/10.5281/zenodo.3554617

Kleinman, E., Ahmad, S., Teng, Z., Bryant, A., Nguyen, T.-H. D., Harteveld, C., & Seif El-Nasr, M. (2020). "And then they died": Using action sequences for data driven, context aware gameplay analysis. *Proceedings of the 15th International Conference on the Foundations of Digital Games* (FDG 2020), 15–18 September 2020, Bugibba, Malta (pp. 1–12). https://doi.org/10.1145/3402942.3402962

Köck, M., & Paramythis, A. (2011). Activity sequence modelling and dynamic clustering for personalized e-learning. *User Modeling and User-Adapted Interaction*, *21*(1), 51–97. https://doi.org/10.1007/s11257-010-9087-z

Lesnard, L. (2006). *Optimal Matching and Social Sciences* (tech. rep.). HAL Open Science. https://halshs.archives-ouvertes.fr/halshs-00008122/document

Liñán, L. C., & Pérez, Á. A. J. (2015). Educational data mining and learning analytics: Differences, similarities, and time evolution. *International Journal of Educational Technology in Higher Education*, *12*(3), 98–112. https://doi.org/10.7238/rusc.v12i3.2515

Malmberg, J., Järvelä, S., & Järvenoja, H. (2017). Capturing temporal and sequential patterns of self-, co-, and socially shared regulation in the context of collaborative learning. *Contemporary Educational Psychology*, *49*, 160–174. https://doi.org/10.1016/j.cedpsych.2017.01.009

Min, W., Mott, B. W., Rowe, J. P., Liu, B., & Lester, J. C. (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* (IJCAI 2016), 9–15 July 2016, Palo Alto, California, USA (pp. 2590–2596). https://www.ijcai.org/Proceedings/16/Papers/368.pdf

Nguyen, T.-H. D., El-Nasr, M. S., & Canossa, A. (2015). Glyph: Visualization tool for understanding problem solving strategies in puzzle games. *Proceedings of the 10th International Conference on the Foundations of Digital Games* (FDG 2015), 22–25 June 2015, Pacific Grove, California, USA. http://www.fdg2015.org/papers/fdg2015_paper_64.pdf

O'Neil, C. (2016). *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Crown.

Paaßen, B., Hammer, B., Price, T. W., Barnes, T., Gross, S., & Pinkwart, N. (2018). The continuous hint factory—Providing hints in vast and sparsely populated edit distance spaces. *Journal of Educational Data Mining*, *10*(1), 1–35. https://doi.org/10.5281/zenodo.3554697

Papamitsiou, Z. K., & Economides, A. A. (2014). Learning analytics and educational data mining in practice: A systematic literature review of empirical evidence. *Educational Technology & Society*, *17*(4), 49–64. https://drive.google.com/file/d/1iLb4Mf3uClTltv0JAUpYbBWOQ-CjeMt-/view

Reilly, J. M., & Dede, C. (2019). Differences in student trajectories via filtered time series analysis in an immersive virtual world. *Proceedings of the Ninth International Conference on Learning Analytics and Knowledge* (LAK 2019), 4–8 March 2019, Tempe, Arizona, USA (pp. 130–134). https://doi.org/10.1145/3303772.3303832

Romero, C., & Ventura, S. (2010). Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *40*(6), 601–618. https://doi.org/10.1109/TSMCC.2010.2053532

Romero, C., Ventura, S., Zafra, A., & De Bra, P. (2009). Applying web usage mining for personalizing hyperlinks in web-based adaptive educational systems. *Computers & Education*, *53*(3), 828–840. https://doi.org/10.1016/j.compedu.2009.05.003

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

Sawyer, R., Rowe, J., Azevedo, R., & Lester, J. (2018). Filtered time series analyses of student problem-solving behaviors in game-based learning. *Proceedings of the 11th International Conference on Educational Data Mining* (EDM 2018), 15–18 July 2018, Buffalo, New York, USA (pp. 229–238). https://educationaldatamining.org/files/conferences/EDM2018/EDM2018_Preface_TOC_Proceedings.pdf

Shute, V. J., Masduki, I., & Donmez, O. (2010). Conceptual framework for modeling, assessing and supporting competencies within game environments. *Technology, Instruction, Cognition & Learning*, *8*(2), 137–161. http://www.oldcitypublishing.com/pdf/2347

Vahdat, M., Ghio, A., Oneto, L., Anguita, D., Funk, M., & Rauterberg, M. (2015). Advances in learning analytics and educational data mining. *Proceedings of the 23rd European Symposium on Artificial Neural Networks* (ESANN2015), 22–24 April 2015, Bruges, Belgium (pp. 297–306). https://www.esann.org/sites/default/files/proceedings/legacy/es2015-18.pdf

Valls-Vargas, J., Ontanón, S., & Zhu, J. (2015). Exploring player trace segmentation for dynamic play style prediction. *Proceedings of the 11th Annual Conference on Artificial Intelligence and Interactive Digital Entertainment* (AAAI 2015), 14–18 November 2015, Santa Cruz, California, USA (pp. 93–99). https://ojs.aaai.org/index.php/AIIDE/article/view/12782

Villareale, J., F. Biemer, C., Seif El-Nasr, M., & Zhu, J. (2020). Reflection in game-based learning: A survey of programming games. In G. N. Yannakakis, A. Liapis, V. V. Penny Kyburz, F. Khosmood, & P. Lopes (Eds.), *Proceedings of the 15th International Conference on the Foundations of Digital Games* (FDG 2020), 15–18 September 2020, Bugibba, Malta (pp. 1–9). https://doi.org/10.1145/3402942.3403011

Zhu, J., Alderfer, K., Furqan, A., Nebolsky, J., Char, B., Smith, B., Villareale, J., & Ontañón, S. (2019). Programming in game space: How to represent parallel programming concepts in an educational game. *Proceedings of the 14th International Conference on the Foundations of Digital Games* (FDG 2019), 26–30 August 2019, San Luis Obispo, California, USA (pp. 1–10). https://doi.org/10.1145/3337722.3337749

Zhu, J., Alderfer, K., Smith, B., Char, B., & Ontañón, S. (2020). Understanding learners' problem-solving strategies in concurrent and parallel programming: A game-based approach. *arXiv:2005.04789*. https://arxiv.org/abs/2005.04789

Zhu, J., & El-Nasr, M. S. (2021). Open player modeling: Empowering players through data transparency. *arXiv:2110.05810*. https://arxiv.org/abs/2110.05810