Psychometrika (2026), **91**, 839–867
doi:10.1017/psy.2026.10096

PSYCHOMETRIC SOCIETY

APPLICATION AND CASE STUDIES - ORIGINAL

Analyzing Complex Educational Data: A Data Analytic Framework for Integrating Structured and Unstructured Eye-Tracking Data

Luyang Fang[1], Shiyu Wang[2], Yinghan Chen[3], Susu Zhang[4], Zichu Liu[2] and Wenxuan Zhong[1]

[1]Statistics, University of Georgia, USA; [2]Educational Psychology, University of Georgia, USA; [3]Mathematics and Statistics, University of Nevada Reno, USA; [4]Psychology; Statistics, University of Illinois at Urbana-Champaign, USA

Corresponding author: Shiyu Wang; Email: swang44@uga.edu

(Received 31 January 2025; revised 31 October 2025; accepted 4 February 2026)

This manuscript is part of the special section on Integrating and analyzing complex high-dimensional data in social and behavioral sciences research. We thank Drs. Eric F. Lock and Katrijn Van Deun for serving as co-Guest Editors.

Abstract

The growing use of computer-based assessments has produced complex process data that capture learners' cognitive and behavioral processes in real time. Among these, eye-tracking data provide rich temporal information on how individuals attend to and process visual information during problem solving. Yet, analyzing such high-dimensional, temporally dependent, and multimodal data remains a methodological challenge. This study introduces a two-component data-analytic framework (DAK) for integrating and interpreting structured and unstructured data in educational assessments. The first component employs a time-aware long short-term memory Autoencoder to extract latent features representing dynamic visual attention patterns. The model extends conventional architectures by incorporating fixation duration and elapsed time between actions, using a data-driven temporal decay function, and optimizing a multi-target reconstruction objective. The second component integrates these extracted features through clustering, categorical data analyses, and mixed-effects modeling to generate construct-relevant validity evidence for test-taking and learning behaviors. We demonstrate the DAK using structured scores and unstructured eye-tracking data from a spatial rotation learning program. Results reveal distinct behavioral patterns linked to test performance and intervention effectiveness, highlighting the potential of multimodal process data to advance psychometric modeling and instrument design.

Keywords: eye-tracking data; feature extraction; high-dimensional data; pattern recognition; process data

1. Introduction

The increasing use of computer-based assessments and digital technologies has produced extensive behavioral data for educational and psychological research. Among these, eye-tracking has become a valuable method for examining cognitive and behavioral processes. Prior studies have used eye-tracking data to uncover cognitive mechanisms during testing (Kaczorowska et al., 2021; Zhu & Feng, 2015), describe test-taking behaviors (Man & Harring, 2023), and gather validity evidence for item and test design (Yaneva et al., 2021, 2022). In particular, applications to mental rotation tasks have provided important insights into spatial reasoning, problem-solving strategies, and gender-related differences in cognitive processing (e.g., Nazareth et al., 2019; Xue et al., 2017).

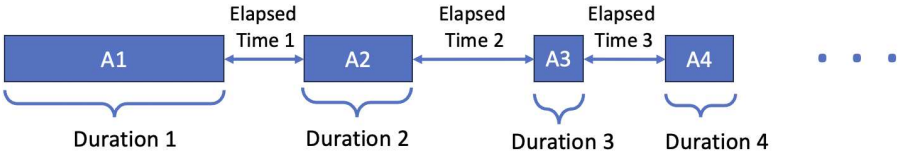

Figure 1. The example of the fixation sequence for one participant on one question.

Our motivation data set stems from an experiment that used a spatial rotation learning program (Wang et al., 2026, 2020) to assess participants' mental rotation skills. The program includes two test blocks and two learning blocks, with learning blocks placed between the test blocks. Each block has 10 multiple-choice questions, with the learning blocks featuring additional interactive elements. Eye-tracking equipment captured participants' eye-movement data during the experiment, which generated a rich array of data that capture both static and dynamic aspects of visual behavior. Typically, eye-tracking systems record raw gaze coordinates over time, which can then be processed into summary eye-movement metrics, such as fixation duration and count, saccade length and velocity, number of glances, and visit frequency or duration within predefined areas of interest (AOIs).

Beyond these aggregated indicators, eye-tracking devices also provide temporal sequences of gaze events—including the order, duration, and transitions between fixations and saccades—offering fine-grained representations of an individual's eye-movement trajectory during task performance. An example is shown in Figure 1 for one participant's fixation sequence in one test question. The action A_t represents the fixation location on an area of interest at time point t, and the duration of each fixation and the elapsed time between two consecutive fixations are also available. Such sequential data enable researchers to explore dynamic patterns of visual attention and information processing that cannot be captured by summary statistics alone (e.g., Liu & Cui, 2025).

Compared with aggregated indicators, unstructured eye-tracking sequence data retain richer, fine-grained information about individual differences. Although our focus is on eye-tracking sequences, their analytic value can be understood within the broader process-data framework, where prior studies have demonstrated how process data can be used to model latent traits, improve proficiency scoring, identify behavioral prototypes, and predict performance outcomes (Chen, 2020; Fang & Ying, 2020; He et al., 2019, 2023; LaMar, 2018; Liu et al., 2018; Tang, 2023; Ulitzsch, Molter, et al., 2022; Zhang, Wang, et al., 2022). These findings suggest that unstructured eye-movement sequences likewise offer rich diagnostic potential for understanding how individuals engage with tasks and solve problems.

The goal of this case study is to analyze both structured data (binary response scores) and unstructured data (eye-movement sequences) from the testing and learning blocks to provide construct validity evidence about participants' testing and learning behaviors. Specifically, we address six research questions (RQ): For testing behaviors, we ask (RQ1) what participants' testing behaviors are in Test Blocks 1 and 2, (RQ2) how these behaviors change between the two blocks and relate to different question types, and (RQ3) how testing behaviors are associated with overall test performance. For learning behaviors, we examine (RQ4) what participants' learning behaviors are in Learning Blocks 1 and 2, (RQ5) how these behaviors vary across question types, and (RQ6) how learning behaviors are associated with testing behaviors to better understand the linkage between learning and assessment processes.

The research questions on testing behaviors (RQ1–RQ3) aim to examine participants' problem-solving processes and explore how variations in response patterns relate to question types and performance outcomes. These analyses are expected to yield construct validity evidence for the test items and provide deeper insights into participants' testing performance. Likewise, the research questions on learning behaviors (RQ4–RQ6) focus on how participants engage with learning materials, how these behaviors vary by question type, and how learning behaviors influence subsequent testing behaviors. Findings from these analyses will contribute validity evidence for the learning design and inform potential refinements to the integration of learning and assessment components.

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

To answer these questions, we must address the challenges inherent in analyzing dynamic, high-dimensional, and multimodal data. The dynamic nature arises from the platform's design, which records data across multiple time points, while the high-dimensional and multimodal aspects stem primarily from the eye-tracking sequences that capture fixation durations and inter-action intervals across four blocks. To tackle these challenges, we propose a data analytic framework (DAK) comprising two integrated components that address two key issues: (1) analyzing unstructured, high-dimensional sequence data and (2) integrating and interpreting evidence across multiple analytic results.

The first component of the DAK proposes a new time-aware long short-term memory (T-LSTM) Autoencoder framework for feature extraction. This framework builds on existing LSTM research (Min et al., 2019; Yin et al., 2021) by incorporating the duration of actions and the elapsed time between consecutive actions, along with the original eye-movement sequence, providing a more comprehensive analysis of eye-tracking data. We demonstrated the effectiveness of this new feature extraction method and provided guidance on how to select the best Autoencoder model based on different hyperparameters. The second component of our DAKs provides systematic guidelines for interpreting the patterns captured in extracted features through an integrated suite of data-analytic tools. When interpreted appropriately, process data (e.g., log files, eye-tracking sequences, and response times) can provide construct-relevant evidence by revealing whether the strategies and behaviors observed during problem solving align with those anticipated in the evidence model (He et al., 2021; Wang et al., 2023). However, a major challenge in applying feature extraction methods to educational assessment tasks lies in the limited interpretability of the extracted features, which often hinders their direct use in addressing substantive research questions. Specifically, we employed a series of statistical analyses, including (a) clustering analyses of extracted features to identify distinct patterns of test-taking and learning behaviors, (b) categorical data analysis techniques to summarize and compare changes in these patterns, thereby providing evidence for the quality of assessment design and the effectiveness of learning interventions, and (c) generalized mixed-effects models to examine the relationships between test-taking behaviors and test performance.

The remainder of this article begins with a description of the motivating dataset and an overview of the proposed DAK. We then detail the two components of the DAK and demonstrate their application in addressing the six research questions using the motivating data. Finally, we conclude with a discussion of key findings, limitations, and directions for future research.

2. Motivation data and overview of the proposed DAK

The structured and unstructured data are from a sample of 70 participants with their interaction of a spatial rotation learning platform, which are summarized in Table 1.[1] For each participant, the structured data include binary response scores for all items in both the test and learning blocks, while the unstructured data consist of eye-movement sequences recorded during task performance. Among various eye-tracking metrics, we focus on fixation sequences, as fixation patterns are well-established indicators of how individuals allocate attention and process information. Longer or more frequent fixations often reflect greater cognitive effort or uncertainty in decision-making (e.g., Rayner, 1998, 2009). In this study, our primary goal is to enhance the interpretability of test-takers' testing and learning behaviors using features derived from eye-tracking sequences, rather than focusing on predictive tasks such as score prediction. An earlier work has conducted a multidimensional exploratory analysis on using 100 eye-tracking summary variables and participants' response score and response time in one test block from this experiment and discovered that the latent structures extracted from fixation-related variables can offer meaningful insights into individuals' test-taking processes (Wang et al., 2026). Based on these considerations, we include only fixation sequence data in the following analyses.

[1]The experiment procedure and data cleaning procedure can be found in Wang et al. (2026).

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

Table 1. Structured and unstructured data from each participant

Block	# Questions	Structured data	Unstructured data
TB1,TB2	10 for each block	Binary score	Fixation sequence
LB1,LB2			(action location, duration, and
			elapsed time)

Note: TB represents the test block, and LB denotes the learning block.

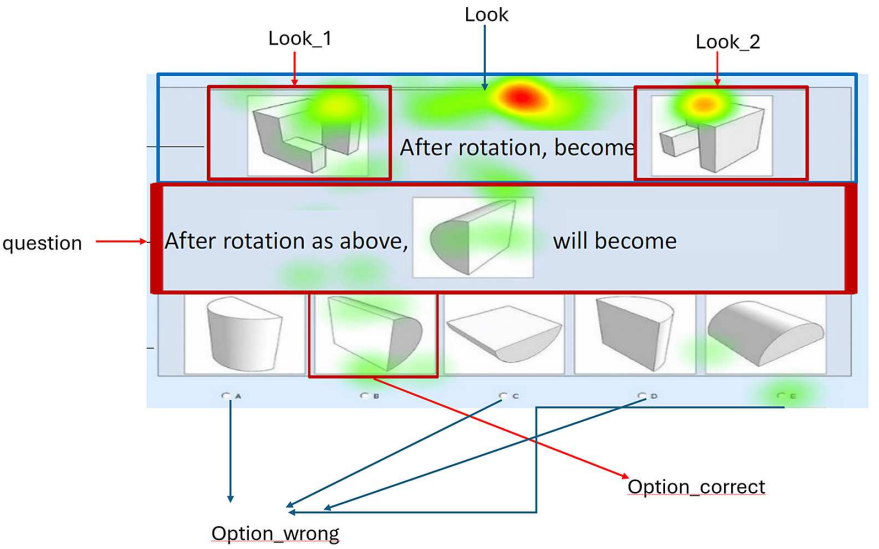

Figure 2. Example of a fixation sequence from one participant to one question.

2.1. The complexity of unstructured eye fixation sequence data

An eye fixation sequence with temporal and sequential dependencies is shown in Figure 1. For this experiment, a participant's fixation sequence on a question contains six unique locations as described by Figure 2. *Look_1* denotes location on the first object (initial position) in the example rotation; *Look_2* represents the second object (ending position) in the example rotation; and *Look* denotes other parts of the Look area. *Question* denotes the central part (circled in red), which represents the new object to be mentally rotated; *option_correct* denotes the location of the correct answer in the multiple-choice options; and *option_wrong* is the location on any of incorrect options.

The learning platform includes seven types of questions across two test blocks and two learning blocks, measuring rotations in one or two directions along the x or y axis for 90 or 180 degrees ($x90$, $x180$, $y90$, $y180$, $x90y90$, $x90y180$, $x180y90$). Details of the design can be found in Table 2 of Wang, Zhang, et al. (2020). The questions in four blocks follow the same format (Figure 2), with the key difference being that participants were not informed of their response accuracy in the test blocks, while in the learning blocks, they were given feedback on correctness and had the opportunity for additional learning interventions.

2.1.1. High dimensionality

The first challenge to address fixation sequences is their high dimensionality and nonstandard format. Consider the fixation sequence data from 70 participants answering 10 questions in test block 1. Treating each participant's sequence for one question as a single observation gives 700 observations, each with a varying number of variables. As shown in Figure 3, the length of eye-tracking sequences varies widely. For instance, the average sequence length is 430, but some exceed 1,000, reaching up to 2,000, far

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

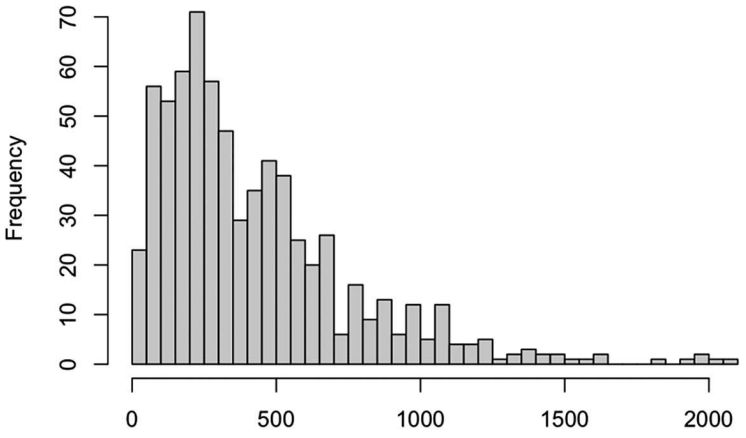

Figure 3. The distribution of fixation sequence lengths for test block 1.

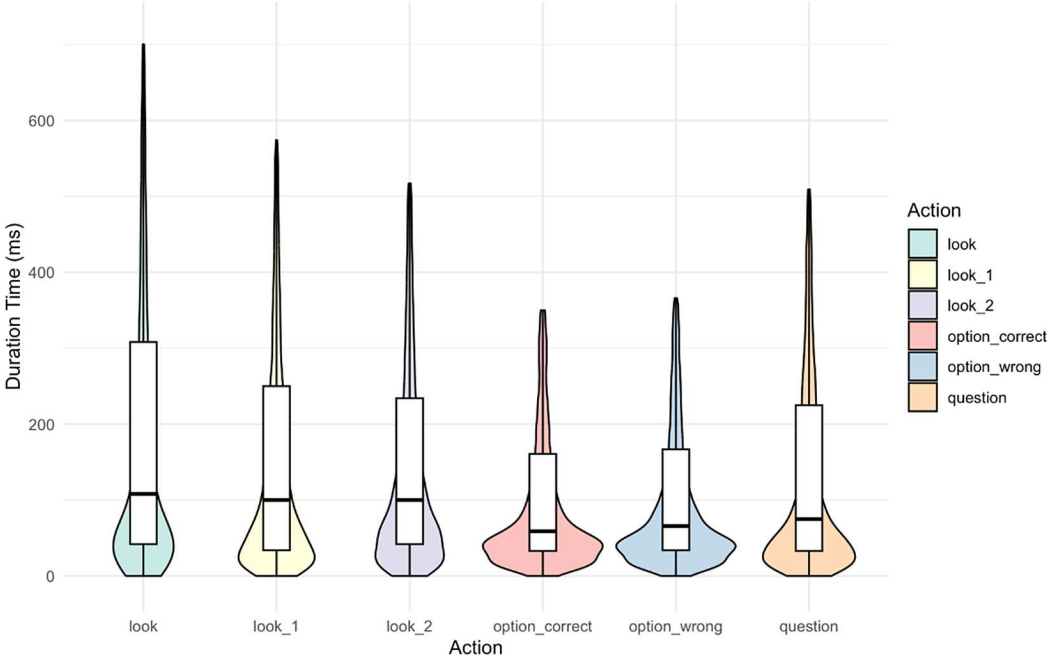

Figure 4. Distribution of fixation duration for each action (test block 1).

surpassing the sample size. This high-dimensional sequence data presents challenges for traditional statistical methods.

2.1.2. Motivation of considering fixation duration

To preliminarily examine whether integrating fixation duration with fixation location provides a more informative basis for distinguishing testing and learning patterns, we analyzed the distribution of fixation durations in Test Block 1 (Figure 4). The results show that fixation duration distributions vary

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

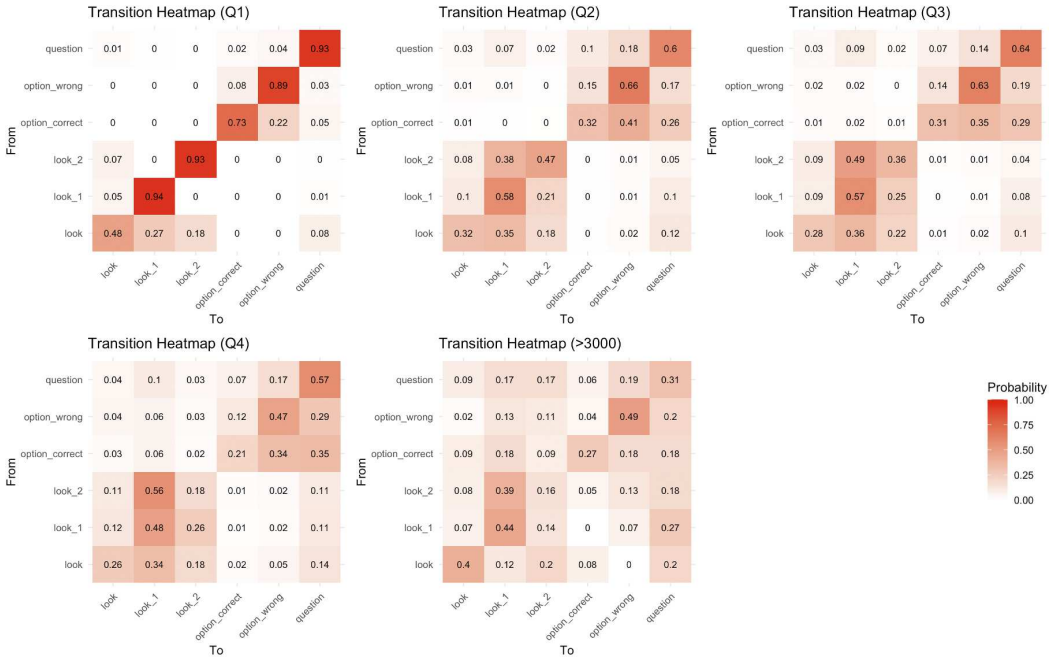

Figure 5. Five transition patterns of consecutive fixations for test block 1 sequences.

notably across locations. For instance, locations, such as "look" and "question," display wider ranges and longer tails, indicating greater variability, whereas "option_correct" and "option_wrong" exhibit more compact distributions. These findings suggest that fixation duration captures meaningful differences across action types and should be examined in conjunction with fixation location sequences when analyzing testing and learning behaviors.

2.1.3. Motivation of considering elapsed time

To examine how elapsed time between consecutive fixations influences transitions, we divided all fixation pairs into five subsets based on quartiles of the elapsed time distribution: Q1 (elapsed times below the 1st quartile), Q2 (between the 1st quartile and median), Q3 (median to 3rd quartile), Q4 (above the 3rd quartile but below 3000 ms), and a final subset for extremely long elapsed times (>3,000 ms). For each subset, we constructed a 6×6 transition matrix to represent the probabilities of transitioning between locations. Each element (i,j) in the matrix represents the probability of transitioning from action i to action j, estimated using the proportion of observed transitions in this cell relative to all transitions. Figure 5 displays these matrices from test block 1. In Q1, participants primarily remain within the same location, indicating that shorter elapsed times strengthen the influence of the current fixation. As elapsed time increases (up to Q4), location transitions become more evenly distributed, reflecting a reduced fixation influence. This trend is most noticeable in the final matrix, where transition probabilities approximate a uniform distribution for very long elapsed times (>3,000 ms). These findings highlight the importance of considering elapsed time when analyzing fixation transitions.

2.2. Overview of the two-component DAK

We propose a comprehensive two-component DAK for complex, high-dimensional structured and unstructured data, as shown in Figure 6.

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

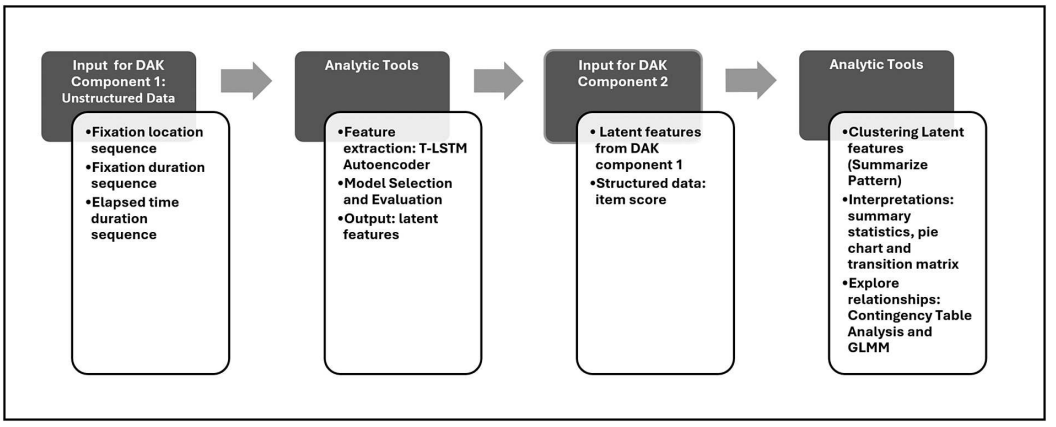

Figure 6. The proposed data analytic framework.

The first component addresses the challenges of analyzing unstructured data with temporal and sequential dependencies. Specifically, it examines two key questions: (1) how to effectively extract features from high-dimensional fixation sequence data by incorporating both fixation durations and elapsed times between consecutive fixations and (2) how to identify the optimal feature extraction model using an appropriate evaluation mechanism. The second component focuses on integrating and interpreting evidence from multiple analytic results to answer educationally meaningful research questions in the context of the spatial rotation experiment. Sections 3 and 4 provide details of these two components.

3. DAK component 1

3.1. Rationale of the DAK component 1

Existing approaches for analyzing unstructured, unsegmented sequence data have been studied in the psychometrics (He & Cui, 2025), process mining (Bogarín et al., 2018), and natural language processing literature. They can generally be categorized into several types. One category relies on summary variables or similarity measures derived from raw sequences (Hahn & Klein, 2025; Ivanová et al., 2022; Liu & Cui, 2025; Schnipke & Scrams, 1997; Ulitzsch et al., 2021). The features or pairwise sequence similarity measures are often used to identify clusters of sequential patterns (Križanić, 2020; Song et al., 2008; van Dongen & Adriansyah, 2009; Yang et al., 2022) that reflect meaningful individual differences. However, the outcomes of such approaches are typically sensitive to how these summary features or dissimilarity measured are defined.

Another category of approaches focuses on data-driven feature extraction from the raw sequences, preserving temporal and sequential pattern information in a data-driven manner. Without pre-specifying a subset of patterns/characteristics to attend to, these methods could yield more faithful representations of underlying individual differences in behavioral or cognitive processes, but at the same time can be more difficult for interpretation. Accordingly, the first component of our analytic framework is designed to extract data-driven features from unstructured data while retaining their temporal and sequential dependencies.

Although various feature extraction techniques have been applied to process data across educational and behavioral domains (e.g., He et al., 2021; Min et al., 2019; Pardos et al., 2014; Peña-Ayala, 2014; Tang et al., 2020; Zhang et al., 2022, 2023), only a few incorporate timestamp sequences alongside process data (Tang et al., 2021; Ulitzsch et al., 2021). Moreover, even when temporal information is considered,

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

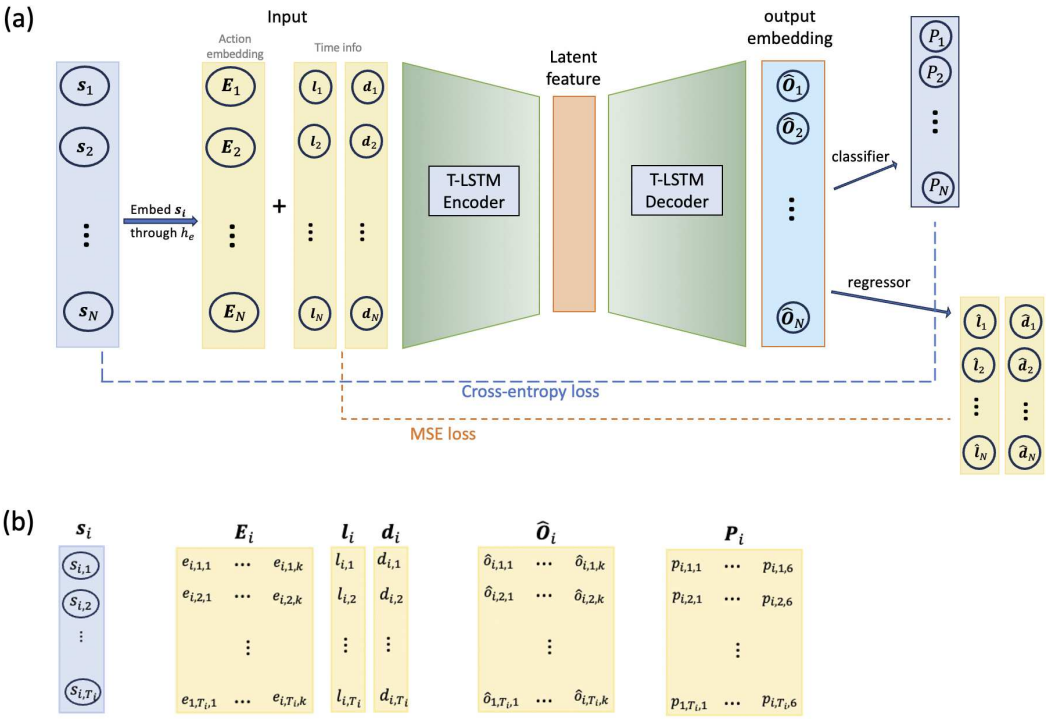

Figure 7. Autoencoder model structure. (a) Overall architecture incorporating temporal information. (b) Notation and data representation at the sequence level.

different types of timing information, such as action duration and time intervals between actions, are rarely differentiated.

In the context of eye-tracking fixation sequences, accounting for different types of time-related information, such as fixation durations and the elapsed time between consecutive fixations, is essential (see Figure 1). Two participants may focus on the same locations in the same order, yet their problem-solving strategies can differ substantially based on their temporal patterns. Variations in fixation duration and inter-fixation intervals can reveal meaningful differences in attention allocation, cognitive load, and decision-making strategies.

3.2. The proposed Autoencoder with T-LSTM

We now introduce an Autoencoder model (Hinton & Salakhutdinov, 2006) to extract hidden features from fixation sequence data. The model incorporates both fixation sequences and their corresponding time and elapsed time data, integrating temporal information into feature extraction. The encoder and decoder use a novel T-LSTM framework, which captures the impact of elapsed time on subsequent actions.

Let $\mathcal{S} = \{\mathbf{s}_i\}_1^N$ denote the fixation sequences, where N is the total number of sequences. The i^{th} sequence, $\mathbf{s}_i = (s_{i,t}, \ldots, s_{i,T_i})^T$, represents the sequence of actions, with $s_{i,t}$ being the t^{th} action and T_i the total number of actions. Each action corresponds to one of six fixation locations: $\mathcal{A} = \{$Look, Look_1, Look_2, Question, Option_correct, and Option_wrong$\}$. Additionally, the time duration sequence for the i^{th} sequence is denoted as $\mathbf{d}_i = (d_{i,t}, \ldots, d_{i,T_i})^T$, where $d_{i,t}$ represents the duration of the t^{th} action. The elapsed time sequence is $\mathbf{l}_i = (0, l_{i,2}, \ldots, l_{i,T_i})^T$, where $l_{i,t}$ denotes the time between actions $s_{i,t-1}$ and $s_{i,t}$. The Autoencoder model, illustrated in Figure 7, extracts representative features from the eye-tracking sequence $\mathbf{s}_i$ with the help of both time information $\mathbf{l}_i$ and $\mathbf{d}_i$. The following sections detail key components of this framework.

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

3.2.1. Input

The categorical actions in $\mathbf{s}_i$ pose a challenge for neural networks due to the absence of inherent numerical relationships. To overcome this, we convert each action into a numerical embedding vector of size k using a learnable embedding layer (Bengio et al., 2003; Hancock & Khoshgoftaar, 2020; Mikolov, 2013). Mathematically, the embedding layer is parameterized by a learnable matrix $\mathbf{W}_e \in \mathbb{R}^{6\times k}$, where 6 is the number of unique actions and k is the embedding size. For each action $s \in \mathcal{A}$, the embedding vector $\mathbf{e} \in \mathbb{R}^k$:

$$\mathbf{e} = \mathbf{W}_e^T \mathbf{u}_s, \tag{1}$$

where $\mathbf{u}_s \in \mathbb{R}^6$ is a one-hot encoded vector representing the action s, such that the s-th entry is 1 and all others are 0. For each sequence $\mathbf{s}_i$, the embedding is represented as a matrix $\mathbf{E}_i = (\mathbf{e}_{i,1}, \ldots, \mathbf{e}_{i,T_i})^T$ with shape $T_i \times k$, where $\mathbf{e}_{i,t} = (e_{i,t,1} \ldots, e_{i,t,k})^T$. By embedding the actions, we map the discrete categories into a continuous vector space, enabling the neural network to better capture semantic relationships between actions.

To capture how time influences these behaviors, we include not only the action sequence $\mathbf{s}_i$ but also the corresponding time duration $\mathbf{d}_i$ and elapsed time $\mathbf{l}_i$, that is, feeding all of them into the Autoencoder model for feature extraction. Mathematically, we concatenate $\mathbf{E}_i$,$\mathbf{l}_i$, and $\mathbf{d}_i$ along the column dimension to form $\mathbf{X}_i = [\mathbf{E}_i, \mathbf{l}_i, \mathbf{d}_i]$, which serves as the input to the Autoencoder.

3.2.2. Encoder and decoder with T-LSTM framework

The Autoencoder model includes two primary parts: an encoder and a decoder. The encoder compresses the input $\mathbf{X}_i$ into a latent representation $\mathbf{z}_i$, formulated as

$$\mathbf{z}_i = f_{encoder}(\mathbf{X}_i), \tag{2}$$

where $\mathbf{z}_i \in \mathbb{R}^h$ is intended to capture high-level features of the input sequence i with dimension h. That is, $\mathbf{z}_i$ represents the key feature of the original input $\mathbf{X}_i$ and serves as the compressed feature representation used for subsequent analysis. Ideally, if $\mathbf{z}_i$ fully captures the information hidden in $\mathbf{X}_i$, then we should be able to reconstruct $\mathbf{X}_i$ based solely on $\mathbf{z}_i$. Thus, the decoder reconstructs the original $\mathbf{X}_i$ from $\mathbf{z}_i$ through

$$\hat{\mathbf{O}}_i = f_{decoder}(\mathbf{z}_i), \tag{3}$$

confirming that $\mathbf{z}_i$ effectively encodes the relevant information. In the Autoencoder model, the encoder block $f_{encoder}$ and decoder block $f_{decoder}$ play a critical role in effectively extracting hidden features.

To effectively capture the underlying structure in the high-dimensional sequential eye-tracking data, we design $f_{encoder}$ and $f_{decoder}$ as extensions of the long short-term memory (LSTM) architecture (Hihi & Bengio, 1995; Hochreiter, 1997), referred to as T-LSTM, which incorporates duration and elapsed time information into the fixation action sequences. The standard LSTM framework is widely used in fields like patient subtyping, anomaly detection, and speech recognition (Baytas et al., 2017; Irie et al., 2016; Lindemann et al., 2021; Yu et al., 2019), effectively capturing sequential dependencies, preserving long-term memory, and mitigating the vanishing gradient problem (Hihi & Bengio, 1995; Pascanu, 2013).

LSTMs consist of a sequence of memory blocks, each containing a cell state and three gates: the input gate, forget gate, and output gate. The cell state carries the information of the current memory block, while the gates regulate the flow of information into and out of the cell state. The forget gate decides which information to discard, allowing the LSTM to forget irrelevant data. The input gate controls the addition of new information to the current cell state, enabling the network to update its memory with new inputs. Lastly, the output gate determines which information to pass to the output, based on the current contents of the cell state and the network's learned parameters. This architecture enables LSTMs to remember important information over long sequences and to discard irrelevant data.

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

In the implementation of an LSTM memory block, the input gate i_t, forget gate f_t, candidate value of the memory cell $\tilde{c}_t$, and output gate o_t at time t are computed with Equations 4–7, respectively,

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right), \tag{4}$$

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right), \tag{5}$$

$$\widetilde{c_t} = \tanh\left(W_c x_t + U_c h_{t-1} + b_c\right), \tag{6}$$

$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right), \tag{7}$$

where W and U are weight matrices, b is the bias vector of each unit, and σ and *tanh* are the logistic sigmoid and hyperbolic tangent activation functions, respectively.

The current memory cell's state c_t is calculated by modulating the current memory candidate value $\tilde{c}_t$ via the input gate i_t and the previous memory cell state c_{t-1} via the forget gate f_t, and $c_t = i_t\tilde{c}_t + f_t c_{t-1}$. Through this process, a memory cell decides whether to keep or forget the previous memory state and regulates the candidate of the current memory state via the input gate. The memory cell state c_t is then controlled by the output gate o_t to compute the memory cell output h_t of the LSTM block at time t,

$$h_t = o_t \tanh\left(c_t\right). \tag{8}$$

To incorporate the crucial temporal information, we develop T-LSTM by employing the elapsed time $\mathbf{l}_i$ to weight the short-term memory built on the work from Baytas et al. (2017). The observations in Section 2.1 show that the passage of time reduces the influence of a previous action on the current one, thereby diminishing how past memory affects the current output. To address this, we propose adding a weight, determined by the elapsed time and pre-defined function $g(\cdot)$, to adjust the previous memory cell state c_{t-1} in the LSTM framework. Specifically, the adjusted previous memory cell state c^*_{t-1} is calculated by

$$\begin{aligned} c^S_{t-1} &= \sigma\left(W_d c_{t-1} + b_d\right) c_{t-1}, \\ \hat{c}^S_{t-1} &= c^S_{t-1} * g(l_t), \\ c^T_{t-1} &= c_{t-1} - c^S_{t-1}, \\ c^*_{t-1} &= c^T_{t-1} + \hat{c}^S_{t-1}. \end{aligned} \tag{9}$$

Accordingly, the current memory cell's state c_t will be

$$c_t = i_t\tilde{c}_t + f_t c^*_{t-1}. \tag{10}$$

An illustration of the T-LSTM memory block is provided in Figure 8. Our T-LSTM autoencoder stacks these blocks to form an encoder–decoder. Given the input sequence $\mathbf{X}_i = [\mathbf{E}_i, \mathbf{l}_i, \mathbf{d}_i]$, the encoder processes it in time order and produces the final hidden states h_{T_i}. We take this last hidden state as the sequence representation, $\mathbf{z}_i := h_{T_i}$ in Equation (2), summarizing the full fixation trajectory together with its timing (durations and elapsed times). The decoder is initialized with the encoder's terminal states (h_{T_i}, c_{T_i}). To ensure that the representation alone carries the information needed for reconstruction, the decoder receives a sequence of zero vectors as input; at each step, it updates its state solely from its internal memory. This "zero-input" design creates a tight bottleneck: if the decoder accurately reconstructs actions and times, then the encoder has indeed captured the essential temporal–sequential structure in $\mathbf{z}_i$.

3.2.3. Output

Each decoder output $\hat{\mathbf{o}}_{i,t}$ is mapped to class probabilities over the six actions by a single-layer perceptron (Almeida, 2020; Rosenblatt, 1958) followed by softmax:

$$\mathbf{p}_{i,t} = \mathrm{softmax}\left(\mathbf{W}_0\hat{\mathbf{o}}_{i,t} + \mathbf{b}_0\right) \in \mathbb{R}^6, \tag{11}$$

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

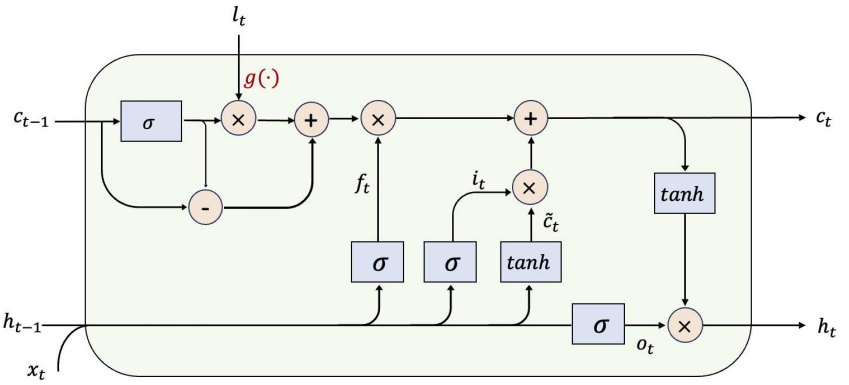

Figure 8. An illustration of a T-LSTM memory block.

where $\mathbf{W}_0$ and $\mathbf{b}_0$ are learnable parameters of the one-layer perception. The resulting probability vector $\mathbf{p}_{i,t}$ contains six elements $(p_{i,t,1},\ldots,p_{i,t,6})^T$, each representing the likelihood of the corresponding action class. The predicted action for timestep t in sequence i is, $\hat{s}_{i,t} = \text{argmax}\,\mathbf{p}_{i,t}$, that is, the action class with the highest probability. For simplicity in notation and implementation, the six distinct actions in action set $\mathcal{A}$ are represented by integers from 1 to 6. We have the reconstructed fixation sequences $\hat{\mathcal{S}} = \{\hat{\mathbf{s}}_i\}_i^N$ with $\hat{\mathbf{s}}_i = (\hat{s}_{i,t},\ldots,\hat{s}_{i,T_i})^T$.

Alongside action reconstruction, we also predict the time sequences $\mathbf{l}_i$ and $\mathbf{d}_i$. Two separate regression layers operate on the decoder outputs $\hat{\mathbf{O}}_i$ to estimate $\hat{l}_{i,t}$ and $\hat{d}_{i,t}$ for each time t:

$$\hat{l}_{i,t} = \text{ReLU}\left(\mathbf{w}_1^\top \hat{\mathbf{o}}_{i,t} + b_1\right), \hat{d}_{i,t} = \text{ReLU}\left(\mathbf{w}_2^\top \hat{\mathbf{o}}_{i,t} + b_2\right), \tag{12}$$

where $\mathbf{w}_1$ and $\mathbf{w}_2$ are learnable weight vectors, and b_1 and b_2 are scalar biases.

Thus, at each step, the decoder yields (1) action probabilities, (2) a non-negative elapsed time, and (3) a non-negative duration, enabling joint reconstruction of sequence content and timing from $\mathbf{z}_i$. The accurate reconstructions of these three components from the extracted feature/representation h_T can validate that we successfully extract the essential information from the original sequences.

To achieve the accurate reconstruction, we optimize the model's learning parameters using the traditional reconstruction loss, that is, the discrepancy between the original and reconstructed data through gradient descent (Ruder, 2016).

The learnable parameters include $\mathbf{W}_e^T$ in the embedding layer, the parameters in the encoder block $f_{encoder}$ and decoder layer $f_{decoder}$ that we will discuss in the following section, as well as $\mathbf{W}_0$, $\mathbf{b}_0$, $\mathbf{w}_1$, $\mathbf{w}_2$, b_1, and b_2 in the classifier and regressors added after the decoder layer. Collectively, these parameters are denoted by $\boldsymbol{\theta}$.

For the fixation action sequences, the discrepancy between $\mathcal{S}$ and the reconstructed $\hat{\mathcal{S}}$ is measured using the cross-entropy loss

$$L_{\text{CE}}(\boldsymbol{\theta} \mid \mathcal{S}) = -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{T_i}\sum_{l=1}^{T_i}\sum_{k=1}^{6}\mathbb{I}(s_{i,t}=k)\log(p_{i,t,k}), \tag{13}$$

where $\mathbb{I}(\cdot)$ is the indicator function. This loss function is a widely used criterion for measuring the difference between the true action and the predicted probability. It ensures that the model is penalized more heavily when it assigns a low probability to the correct action. Considering the reconstruction performance on temporal information, we apply the widely used mean squared error (MSE) loss, defined as follows:

$$L_{\text{MSE}}(\boldsymbol{\theta} \mid \mathcal{T},\mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{T_i}\sum_{t=1}^{T_i}\left[\left(l_{i,t}-\hat{l}_{i,t}\right)^2 + \left(d_{i,t}-\hat{d}_{i,t}\right)^2\right]. \tag{14}$$

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

We combine the cross-entropy loss L_{CE} for action prediction and the MSE loss L_{MSE} for time sequence reconstruction into a unified objective function:

$$L = L_{\text{CE}} + \lambda L_{\text{MSE}}, \tag{15}$$

where λ is a hyperparameter balancing the contributions of two parts. By minimizing this combined loss, the Autoencoder effectively learns to extract meaningful hidden representations from the input, enabling accurate reconstruction of both the action sequence and its associated temporal information from the hidden space.

3.3. Implementation issues and discussion

In this work, we adopt a one-layer T-LSTM structure for both the encoder and decoder blocks. Under this structure, four key issues are considered: the selection of function $g(\cdot)$ in Equation (9), the determination of three hyperparameters: weighting parameter λ, embedding size k, and hidden feature size h. The strategy we use is to first identify potential pools for each function/hyperparameter and then select the optimal values through several evaluation criteria. Below, we provide guidance on how to set reasonable pools for each function/hyperparameter.

3.3.1. Selection of $g(\cdot)$

Function $g(\cdot)$ essentially captures how elapsed time affects subsequent actions, which can be proposed based on evidence from a data set. As shown in Figure 5, longer elapsed times correspond to a diminished influence of the current fixation on the next one. Thus, a decreasing function is a natural choice in our case. We observed from Figure 5 that as elapsed time increases, the transition pattern changes rapidly at first but stabilizes over longer intervals. For instance, when the elapsed time is around 800 ms, the transition pattern closely resembles that at very long intervals (e.g., > 3,000 ms). This empirical pattern supports the use of an exponential decay function, since it decays quickly at the beginning and then gradually levels off, consistent with the observed transition dynamics. The exponential form is also mathematically simple, differentiable, and widely used in modeling memory decay and time-dependent influence in sequential data, making it both interpretable and computationally efficient. Consequently, we set $g(l) = \exp(-\alpha l)$, with $\alpha = 0.003$, a value carefully selected to match the empirical transition patterns observed in the data (see Figure 9(a)).

For comparison, three additional functions are presented in Figure 9(b)–(d), illustrating common possible patterns in how elapsed time impacts subsequent actions. Specifically, the constant function $g(l) = 1$ in Figure 9(b) represents the standard LSTM, where elapsed time has no effect on subsequent actions. In contrast, the other two functions exhibit different trends: one shows an initial decrease in influence followed by an increase, while the other demonstrates a consistently increasing trend.

3.3.2. Weighting parameter λ

To determine the weighting parameter λ in Equation (15), which balances the action and time sequence contributions, we ensure that their losses are on a similar scale during early training. This prevents one part from dominating. In our dataset, L_{CE} is on the scale of 1, while L_{MSE} is on the scale of 0.1. Since our primary goal is to reconstruct the action sequences, with time information as supplementary, we set the maximum λ to 10 to align L_{CE} and λL_{MSE} on the same scale. We also test values of 0.1 and 1 to emphasize L_{CE}.

3.3.3. Embedding size k

Theoretical and empirical work suggests that for a large number of unique actions, the optimal embedding size is proportional to the number of actions, with a small scaling factor, such as 0.2 (Guo & Berkhahn, 2016; Yin & Shen, 2018). In our case, with only six unique actions, we consider embedding

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

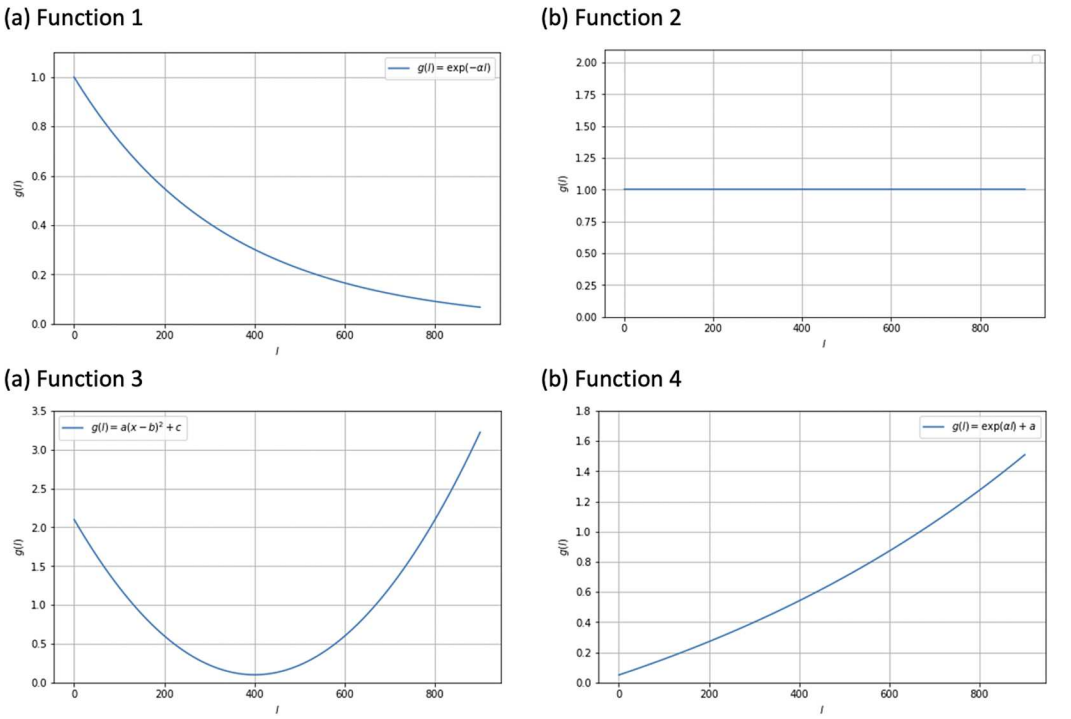

Figure 9. (a) The proposed data-driven function. (b)–(d) Three potential alternative functions.

sizes from the pool $5, 10, 15$, capping at 15 to prevent over-fitting. A smaller size of 5 is suitable for simple action sequences, while 15 offers greater capacity to capture complex patterns.

3.3.4. Hidden size h

The selection of the hidden size, that is, the dimension of the extracted features, is closely related to the nature of the input data. A smaller hidden size may result in the loss of important information, while a larger one risks the model learning an identity function, thereby failing to extract meaningful features. Given that the average sequence length of our data is around 400, and the belief that students' problem-solving behaviors can be effectively represented by features of relatively low dimensionality, we set the largest hidden size to be 20, which is 5% of the average sequence length. The possible pool for the hidden size is defined as $\{5, 10, 20\}$, which allows for experimentation to determine the optimal balance between compression and reconstruction fidelity. For more complex datasets or sequences containing richer information, we recommend considering higher dimensions, such as 30 or 50. However, the hidden size should generally remain modest to ensure effective information compression and to help subsequent analyses.

3.3.5. Evaluation

To evaluate the Autoencoder model's performance with a set of hyperparameters, we focus on its ability to accurately reconstruct the fixation action sequences $\mathcal{S}$, emphasizing the reconstruction of action sequences using temporal information. Performance is assessed based on three criteria. The first is *Accuracy*, which measures the proportion of correctly reconstructed actions compared to the original

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

input sequence and is defined as

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{L_i}\sum_{l=1}^{L_i}\mathbb{I}(s_{i,l}=\hat{s}_{i,l}). \tag{16}$$

The second is *BLEU (Bilingual Evaluation Understudy Score)*, which evaluates the similarity between the original and reconstructed sequences based on overlapping n-grams. For each sequence pair $(\mathbf{s}_i, \hat{\mathbf{s}}_i)$, the BLEU score is computed as

$$\text{BLEU} = \exp\left(\frac{1}{M}\sum_{m=1}^{M}\log p_m\right), \tag{17}$$

where $p_m = \frac{\text{Number of overlapping m-grams between } \mathbf{s}_i \text{ and } \hat{\mathbf{s}}_i}{\text{Total number of m-grams in } \hat{\mathbf{s}}_i}$ is the precision of m-grams of length m. In this study, we set $m = 4$ and compute the final BLEU score by averaging the BLEU scores across all sequences.

The last is *ROUGE (Recall-Oriented Understudy for Gisting Evaluation)*, which evaluates the quality of the reconstructed sequence by comparing m-gram recall and the longest common subsequence (LCS). For m-grams, the ROUGE score for a sequence pair $(\mathbf{s}_i, \hat{\mathbf{s}}_i)$ is defined as

$$\text{ROUGE-M} = \frac{\text{Number of overlapping m-grams between } \mathbf{s}_i \text{ and } \hat{\mathbf{s}}_i}{\text{Total number of m-grams in } \mathbf{s}_i}. \tag{18}$$

For the LCS, ROUGE-L is defined as

$$\text{ROUGE-L} = \frac{\text{LCS length}}{\text{Length of } \mathbf{s}_i}. \tag{19}$$

In this work, we calculate the ROUGE score for each sequence by $\text{ROUGE} = \frac{\text{ROUGE-1+ROUGE-2+ROUGE-L}}{3}$, and average it over all sequences. These three metrics provide a comprehensive evaluation of the Autoencoder's performance by assessing both token-level reconstruction (accuracy) and sequence-level quality (BLEU and ROUGE). Higher values indicate better model performance.

4. DAK Component 2

The primary output of our T-LSTM Autoencoder framework is the set of learned feature representations $\{\mathbf{z}_i\}_1^N$ (Equation (2)). These features serve as compressed representations of the eye-tracking sequences, capturing both the sequential patterns of fixations and their temporal dynamics (duration and elapsed time). Using the extracted features, we apply a set of statistical methods to analyze the features and structured response data to address the six research questions described in the Introduction section.

4.1. Clustering analysis

To explore participants' testing and learning behaviors, we apply hierarchical clustering to features extracted from fixation sequence data during testing and learning blocks. The clustering uses pairwise dissimilarities, computed with the Euclidean distance metric. We implement hierarchical clustering using the *Ward.D* method in the *hclust* function from the *stats* R package (Ward Jr., 1963). To determine the optimal number of clusters, we use both the hierarchical structure in the dendrogram and silhouette analysis (Rousseeuw, 1987; Shutaywi & Kachouie, 2021). The silhouette index evaluates both intra-cluster cohesion and inter-cluster separation. For each point i, the intra-cluster distance a_i is the average Euclidean distance to other points in the same cluster, while the nearest-cluster distance b_i is the minimum average distance to points in any other cluster. The silhouette coefficient s_i is calculated as

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}, \tag{20}$$

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

with values ranging from −1 to 1, where values closer to 1 indicate better clustering. The average silhouette width is computed for different cluster numbers, and the number of clusters that maximizes this value is chosen as optimal.

4.2. Statistical tools facilitating interpretation

4.2.1. Descriptive statistics and visualization

To address research questions RQ1 and RQ4, we calculate descriptive statistics for each cluster, including the number of sequences, average response accuracy, and the distribution of fixation locations, presented via pie charts. We also adjust the transition matrix for each cluster to highlight deviations in transition probabilities from the overall pattern, aiding our interpretation of testing and learning behaviors based on clusters from the testing and learning blocks.

4.2.2. Contingency table analysis

To address research questions RQ2, RQ5, and RQ6, we perform contingency table analysis between testing clusters across test blocks 1 and 2, and across different item types in testing blocks (RQ2), as well as between learning clusters across the two learning blocks and different item types (RQ5), and between learning and testing clusters RQ6. We use mosaic plots, where the area of each rectangular tile represents the conditional relative frequency for a cell in the contingency table, to summarize the patterns of association. Cramer's V is used to calculate the association between testing clusters and item types, and between learning clusters and item types. Additionally, χ^2 tests of independence, using likelihood ratio statistics (G^2), assess the significance of associations between the two sets of clusters.

4.2.3. Generalized mixed effect model

To explore the relationship between testing clusters and the test performance (RQ3), we fit the generalized linear mix-effect model (GLMM) (Breslow & Clayton, 1993) where the outcome variable is the binary score to each question (1 indicating correct and 0 otherwise), the explanatory variables include the testing cluster ID, the type of items, and the block identifier. Additionally, we include a person-level random effect to account for individual-level variability, capturing the within-person correlation in test performance. Note that we reduce the item types to levels of 2, indicating an item measures one direction rotation or two direction rotations to simplify the interpretation. The GLMM is conducted via *lme*4 R package (Bates, 2015).

5. Application of the two-component DAK

As our six research questions are related to testing and learning behaviors, in this section, we provide the feature extraction results for the testing and learning blocks separately. The feature extraction procedure is executed on an NVIDIA Tesla V100 Tensor Core, and the clustering procedure is executed on a Mac with a 10-Core M1 Max processor and 32 GB memory, utilizing the CPU.

5.1. The best fitted T-LSTM Autoencoder model for testing blocks

As discussed in Section 3.2, we define the function g as $g(l) = \exp(-0.003l)$. For comparison, we evaluate three alternatives: $g(l) = 1$ (comp1), $g(l) = 2(\frac{l}{400} - 1)^2 + 0.1$ (comp2), and $g(l) = \exp(0.001l) - 0.95$ (comp3), as shown in Figure 9. We also note that comp1 represents the standard LSTM, where elapsed time has no effect on subsequent actions. For each $g(.)$ function, we fit different T-LSTM Autoencoder models using all combinations of hyperparameters: the weighting parameter $\lambda \in \{0.1, 1, 10\}$, embedding size $k \in \{5, 10, 15\}$, and hidden size $h \in \{5, 10, 20\}$, as outlined in Section 3.2.

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

Table 2. Comparison of model performance across different metrics (Accuracy, BLEU, and ROUGE) for the testing blocks, with their chosen hyperparameters

	Accuracy				BLEU				ROUGE			
Model	Embed	Hidden	λ	Score	Embed	Hidden	λ	Score	Embed	Hidden	λ	Score
comp1	5	20	10.0	0.393	15	5	0.1	0.226	15	5	0.1	0.366
comp2	15	20	0.1	0.394	15	5	0.1	0.215	15	5	0.1	0.360
comp3	10	20	0.1	**0.398**	10	10	0.1	0.207	10	10	0.1	0.347
our	10	10	0.1	0.393	15	5	0.1	**0.233**	15	5	0.1	**0.383**

Note: The best score for each metric is highlighted in bold.

After data cleaning, we obtained 1,387 fixation sequences from the two testing blocks, which were split into training, validation, and test sets (70%:15%:15%). We systematically evaluated all hyperparameter combinations using grid search (Liashchynskyi & Liashchynskyi, 2019), training each configuration on the training set and validating it on the validation set to identify the optimal configuration. The best configuration for each $g(l)$ was then tested on the test set for an unbiased measure of generalization. This three-way split and validation-driven hyperparameter selection helped prevent overfitting and ensured reliable performance estimates.

Table 2 compares the performance of the best T-LSTM Autoencoder model under different $g(\cdot)$ functions using accuracy, BLEU, and ROUGE scores. The comp2 function performs best in accuracy (0.398), while our proposed function is slightly lower (0.393). However, our function outperforms others in BLEU (0.233) and ROUGE (0.383). While accuracy differences between the functions are small, BLEU and ROUGE scores vary more significantly, ranging from 0.207 to 0.233 for BLEU and 0.347 to 0.383 for ROUGE. BLEU and ROUGE measure sequence-level quality, while accuracy focuses on token-level reconstruction. These results confirm that our data-driven function performs better in sequence-level quality.

We observe that the optimal hyperparameter configuration based on BLEU and ROUGE is the same for both our proposed function and the three competitors. However, the configuration selected based on accuracy differs from this. Given the minimal differences in accuracy and the consistency between BLEU and ROUGE, we choose the hyperparameter configuration selected by BLEU and ROUGE and use it to extract the features. That is, for the testing blocks, the extracted feature has a dimension of $h = 5$.

5.2. The best fitted T-LSTM Autoencoder model for learning blocks

Following the same procedure used for the testing blocks, we evaluate the T-LSTM Autoencoder model for the learning block fixation sequence data. For the learning blocks, we design the function $g(\cdot)$ as $g(l) = 1.5\exp(-0.004l)$. To compare its performance, we also consider three alternative functions: $g(l) = 1$ (comp1), $g(l) = 1.5(\frac{l}{300} - 1)^2 + 0.1$ (comp2), and $g(l) = 0.5\exp(0.003l) - 0.45$ (comp3), as illustrated in Figure A1 in the Supplementary Material. The performance of the model with different $g(\cdot)$ functions is evaluated using accuracy, BLEU score, and ROUGE score, with results presented in Table 3.

We observe similar trends between the learning blocks and the testing blocks. First, different choices for the $g(\cdot)$ function show comparable results in terms of accuracy, with the highest value being 0.409 and the lowest 0.391, while showing significant variation in BLEU and ROUGE scores. Second, the function we designed achieves the best performance in BLEU (0.147) and ROUGE (0.292). Third, the optimal hyperparameter configuration based on BLEU and ROUGE is consistent for both our proposed function and comp1, whereas the configuration selected based on accuracy differs. Given the small variations in accuracy and the strong alignment between BLEU and ROUGE metrics, we use the configuration identified by BLEU and ROUGE to extract features. Specifically, for the learning blocks, the extracted features have a dimension of $h = 20$.

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

Table 3. Comparison of model performance across different metrics (Accuracy, BLEU, and ROUGE) for the learning blocks, with their chosen hyperparameters

	Accuracy				BLEU				ROUGE			
Model	Embed	Hidden	λ	Score	Embed	Hidden	λ	Score	Embed	Hidden	λ	Score
comp1	5	5	0.1	0.409	5	10	1	0.139	5	10	1	0.268
comp2	10	20	0.1	0.392	15	20	1	0.073	5	5	0.1	0.214
comp3	5	10	0.1	0.391	15	10	1	0.075	15	5	1	0.204
our	5	20	0.1	0.400	10	20	0.1	**0.147**	10	20	0.1	**0.292**

Note: The best scores for each metric are highlighted in bold.

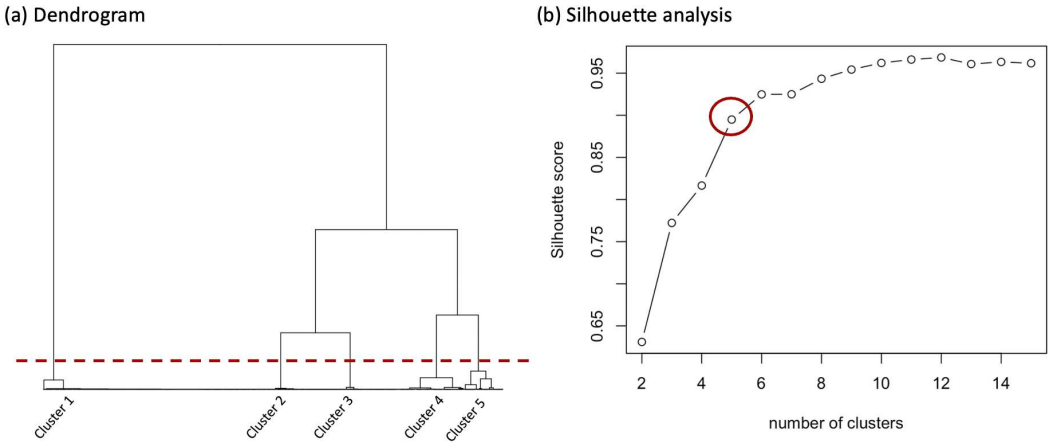

Figure 10. (a) Hierarchical clustering dendrogram. (b) Silhouette analysis.

5.3. Results for testing behavior research questions (RQ1–RQ3)

5.3.1. Hierarchical clustering results

The dendrogram and Silhouette analysis of the hierarchical clustering results using features from test blocks 1 and 2 are shown in Figure 10. The Silhouette analysis suggests an optimal cluster number of 5, where the silhouette score peaks and then plateaus with minimal variation. The dendrogram also supports this choice, with clear separation between clusters at this level. Beyond this, merging occurs at higher heights, indicating weaker cohesion within larger clusters.

5.3.2. RQ1: Interpretation of testing clusters

In Table 4, we present basic information for each cluster, including the number of observations, average accuracy, and the proportion of sequence originating from test block 1. These basic statistics highlight notable differences among the clusters. For example, Cluster 1 has the highest average accuracy (0.74), whereas Cluster 2 shows the lowest accuracy (0.59). A detailed analysis of each cluster and their corresponding problem-solving strategies is provided in the following sections.

Figure 11 shows the proportion of each fixation location in each cluster. Across all clusters, the highest proportions are for "look_1" and "question," indicating participants focus on these areas during problem-solving. All clusters have low fixation proportions for "option_correct," suggesting less gaze effort for correct options. "Look" consistently has the lowest fixation proportions, indicating limited attention to non-key areas.

Cluster 1 has a relatively balanced distribution of fixations on these different locations, with moderate focus on Look_1 (23%) and Question (26%); the highest exploration of incorrect options and correct

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

Table 4. Descriptive statistics for five testing clusters

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Number	699	204	117	154	213
Avg. acc	0.74	0.59	0.68	0.64	0.65
Prop of block 1	68%	22%	26%	41%	35%

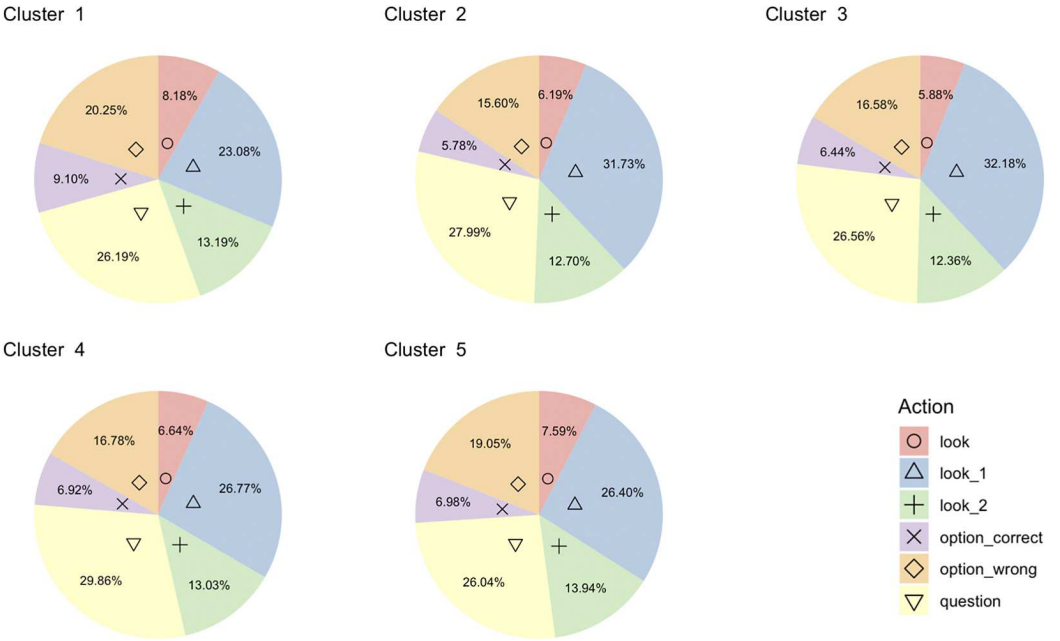

Figure 11. The distribution of six fixation locations for each cluster.

Table 5. Summary statistics of sequence length distribution for each cluster

Cluster	Min	1st Q	Median	Mean	3rd Q	Max
1	2.00	29.00	48.00	53.55	70.00	167.00
2	29.00	96.75	141.50	171.69	225.00	635.00
3	25.0	78.0	123.0	142.6	178.0	404.0
4	4	80.25	115.00	135.44	176.50	543.00
5	21.0	75.0	117.0	134.9	161.0	458.0

options among the five clusters. The fixation location distributions from Clusters 2 and 3 are similar, especially the proportions of Look 1 are similar and higher than the other three clusters. Cluster 4 has the highest proportion for the Question (30%) among the five clusters and relatively balanced focus on Look_1 (27%) and Option_Wrong (17%). Cluster 5 has balanced proportions between Look_1 (26%), Look_2 (14%), and Question (26%).

Table 5 provides the summary statistics for the distribution of sequence length for each cluster. Cluster 1 has the shortest average sequence length compared to the other four clusters. The remaining four clusters have similar sequence length distributions, with Cluster 2 exhibiting the longest sequence lengths among them.

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

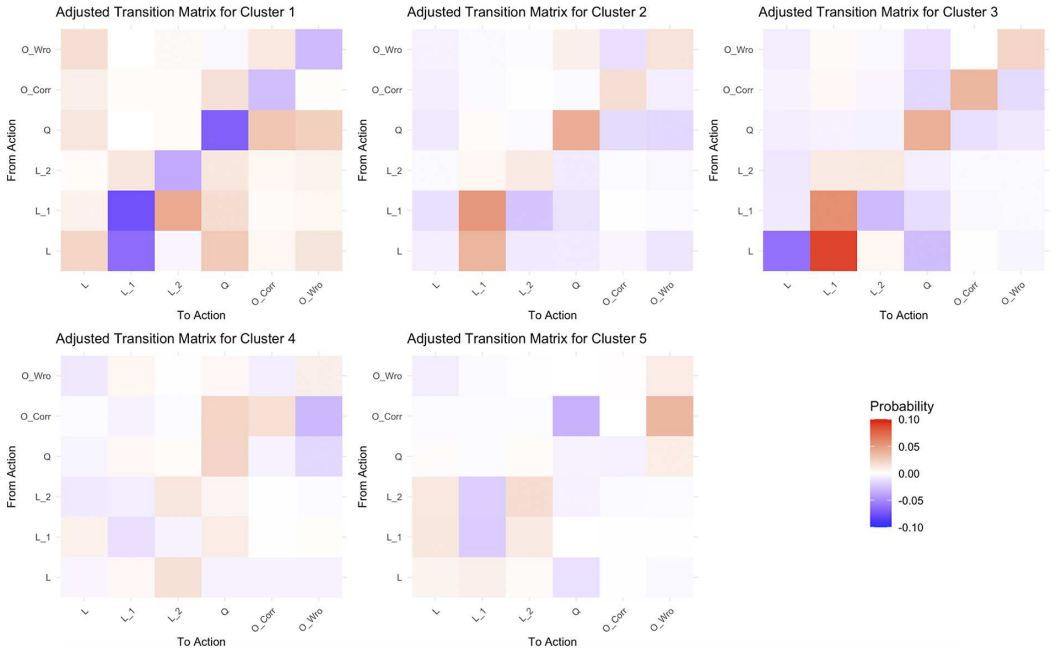

Figure 12. The adjusted transition matrix for five clusters.

Table 6. Problem solving (PS) behaviors interpretation

Cluster	Characteristics
TC1 (Quick Navigation PS)	Exploratory and less focused on screen
TC2 (Example-Driven PS)	Highly rely on example questions and choice they choose
TC3 (Targeted PS)	Focused and deliberate engagement with specific areas
TC4 (Less decisive PS)	Moderate change, with slightly cautious tendencies
TC5 (Balanced PS)	Balance explorations and answering questions

The adjusted transition matrix for each cluster in Figure 12 highlights how transition probabilities deviate from the overall pattern. Positive values (red) indicate transitions that occur more frequently within the cluster compared to the overall data, while negative values (purple) indicate transitions that occur less frequently. Cluster 1 shows fewer self-transitions, suggesting participants avoid focusing on the same areas or lingering too long. Cluster 2 has more self-transitions (L1–L1, Q–Q) and fewer L1–L2 transitions, indicating less systematic exploration. Cluster 3 mirrors Cluster 2 but with fewer transitions from Wrong to Q, Q to Wrong, and Q to Correct, suggesting a more focused problem-solving approach. Cluster 4's pattern is similar to the overall average, with slight increases in transitions like L to L1 and Q to Q, reflecting a more cautious strategy. Slight decreases in transitions like L to Q and Correct to Wrong suggest reduced guessing or incorrect answers. Cluster 5 resembles Cluster 4 but shows fewer repetitive movements, such as L1 to L1 and Correct to Q, and more transitions from Correct to Wrong.

Based on the above analysis, we interpret the five testing clusters as representing five distinct problem-solving strategies, summarized in Table 6.

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

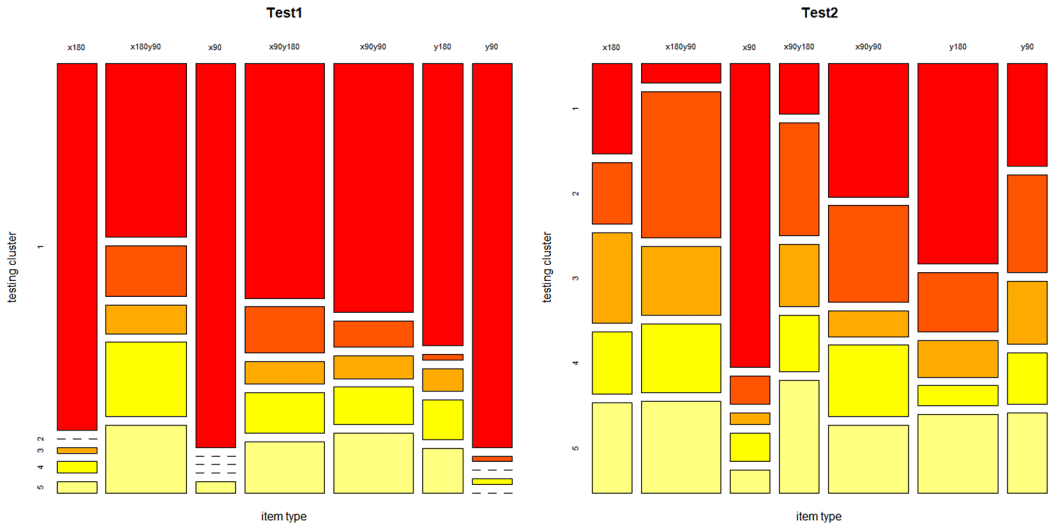

Figure 13. The mosaic plot of contingency table between item type and testing cluster in two tests.

5.3.3. RQ2: The change of problem-solving strategies

Figure 13 shows two mosaic plots comparing five problem-solving strategies and seven item types across two test blocks. In test block 1 (before learning interventions), participants primarily use the quick navigation strategy (Cluster 1) for easier one-direction rotation items ($x90$, $y90$, $x180$, $y180$). This strategy decreases as the items become more complex with two-direction rotations ($x180y90$, $x90y90$, $x90y180$), indicating greater difficulty. The targeted strategy (Cluster 3) is less common but increases slightly for more difficult two-direction rotations. The remaining strategies (Clusters 2, 4, and 5) show varying patterns. The G^2 test reveals a significant dependence between the testing cluster and item type in test block 1 ($G^2(24) = 159.15$, $p < 0.001$), with Cramer's V of 0.216, indicating a moderate association.

We observed a marked decrease for quick navigation (Cluster 1) in test block 2 compared to test block 1, especially on more difficult items ($x180y90$, $x90y180$), suggesting participants have moved away from a stepwise exploratory approach after learning interventions. The other four clusters all increase across different types of items. The G^2 statistics show a significant dependence between testing cluster in test block 2 and item type at significance level of 0.05, $G^2(24) = 184.35$, $p < 0.001$. The Cramer's V is 0.249, indicating a moderate association.

5.3.4. RQ3: The relationship between problem-solving strategies and score

The GLMM results are summarized in Table 7. Quick navigation strategy (Cluster 1) serves as the reference group, with test takers using this strategy showing the highest probability of correct responses after controlling for covariates. Both Example Driven (Cluster 2) and Less Decisive (Cluster 4) strategies have significantly lower probabilities of correct responses compared to Quick Navigation. No significant difference in response probability is found between Quick Navigation (Cluster 1) and Targeted strategy (Cluster 3), suggesting similar performance levels. Although not significant, the Balanced strategy (Cluster 5) shows a slightly lower probability of correct responses than Quick Navigation, implying that it may not consistently lead to correct outcomes. More complex items (item_type = 1) significantly reduce correct response probabilities, highlighting the difficulty of items requiring two skills. Questions in test block 2 (after intervention) show significantly higher probabilities of correct responses than test block 1.

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

Table 7. Generalized linear mixed model results

	Estimate	Std. error	*z* value	Pr(> |z|)
(Intercept)	1.6821	0.1840	9.140	< 2e-16
cls2	−0.6456	0.2502	−2.581	0.009856
cls3	−0.1737	0.2965	−0.586	0.558150
cls4	−0.5588	0.2540	−2.200	0.027823
cls5	−0.2745	0.2363	−1.161	0.245440
item_type2	−1.3279	0.1743	−7.618	<0.001
block_id1	0.5451	0.1644	3.317	0.000911

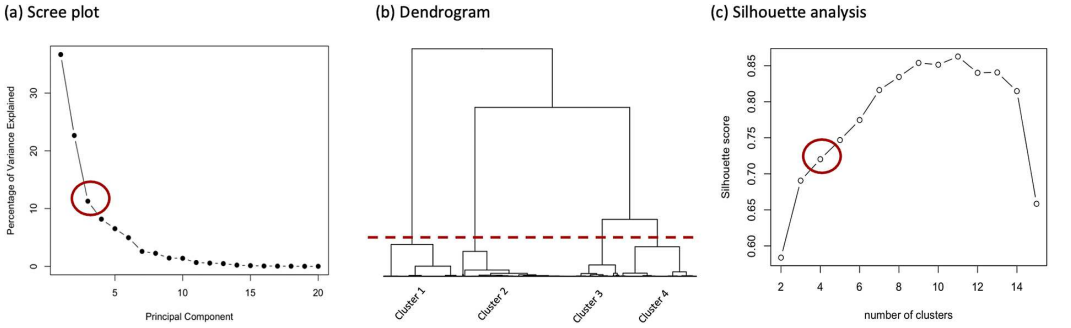

Figure 14. (a) Scree plot showing the percentage of variance explained by each principal component. (b) Hierarchical clustering dendrogram. (c) Silhouette analysis.

5.4. Results for learning behavior research questions (RQ4–RQ6)

5.4.1. Hierarchical clustering results

The extracted features from fixation sequences in learning blocks 1 and 2 have a high dimensionality of 20, which can make distance calculations for hierarchical clustering less meaningful due to the curse of dimensionality. To address this, we apply principal component analysis (PCA) to reduce dimensionality, removing noise and redundancy for more efficient clustering and improved interpretability. Based on the Scree plot in Figure 14(a), we identify three principal components as optimal, as the "elbow" in the plot shows that these components capture most of the variance, with additional components explaining minimal variance. Hierarchical clustering is then performed using these three components.

The results of the Silhouette analysis and dendrogram are shown in Figure 14(b) and (c). The Silhouette analysis shows a significant increase in the score when the number of clusters rises from two to three. Adding more clusters continues to improve the score, but at a slower rate, up to nine clusters. However, nine clusters reduce interpretability. Based on the dendrogram, four clusters are identified as the most suitable, as they are well-separated and avoid the weaker cohesion seen with further merging.

5.4.2. RQ4: Interpretation of learning clusters

The descriptive statistics for each learning cluster, including the number of observations, average accuracy, and the proportion of data from learning block 1, are summarized in Table 8. The fixation sequence in Clusters 1 and 2 has higher response accuracy than Clusters 3 and 4, and learning block 1 is dominated by Clusters 1 and 2.

Figure 15 documents the distributions of eye fixation locations for each cluster. Cluster 1 shows a balanced approach, distributing attention across the "Question" (26.82%), the example object's initial position ("Look 1," 25.90%), and other elements like "Look 2" (14.96%) and "Option Wrong" (16.09%).

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

Table 8. Descriptive statistics for four learning clusters

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Number	338	489	212	350
Avg. acc	0.84	0.87	0.67	0.67
Prop of learning block 1	59%	78%	15%	24%

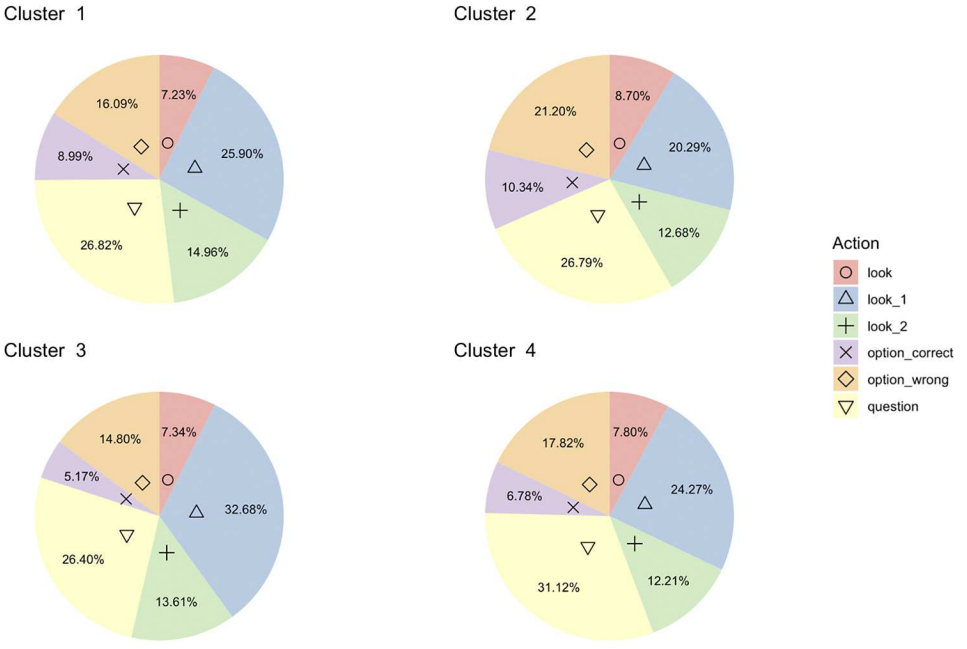

Figure 15. The distribution of six fixation locations for each cluster.

They balance revisiting the problem with analyzing both correct and incorrect options. Cluster 2 exhibits a fairly even distribution, with notable focus on "Look 1" (20.29%), "Option Wrong" (21.20%), and "Question" (26.79%). Participants likely use feedback to assess why their previous solution was incorrect, with high attention to "Option Wrong" indicating analysis of earlier mistakes. Cluster 3 focuses heavily on the example object's initial position ("Look 1," 32.68%), with less attention to answer options ("Option Wrong" 14.80%, "Option Correct" 5.17%). They spend significant time on the "Question" (26.40%) and some on "Look 2" (13.61%), relying primarily on the example setup to reanalyze the task, with limited focus on validating or correcting their solution. Cluster 4 focuses most on the "Question" (31.12%), indicating strong engagement with the task. Their attention to the example object ("Look 1," 24.27%; "Look 2," 12.21%) and options ("Option Wrong" 17.82%; "Option Correct" 6.78%) is lower, suggesting they prioritize applying feedback to the task rather than analyzing object transitions or comparing previous solutions.

Table 9 summarizes the distribution of sequence lengths for each cluster. Cluster 1 has short sequences, with a maximum of 331 fixations. Most sequences range from the 1st quartile (26.0) to the 3rd quartile (78.0), indicating brief task engagement. Cluster 2 shows the shortest sequences among four clusters, with a maximum of 124 fixations. Most sequence lengths range from 22.0 (1st quartile) to 48.0 (3rd quartile), reflecting minimal engagement with task components. Cluster 3 has the longest sequences, with a maximum of 555 fixations. Most sequences range from 71.75 (1st quartile) to 160.25 (3rd quartile), suggesting extensive engagement with the task. Cluster 4 has moderate sequence lengths, with a maximum of 451 fixations. Most sequences range from 54.0 (1st quartile) to 104.75 (3rd quartile),

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

Table 9. Summary statistics of sequence length distribution for each cluster

Cluster	Min	1st Q	Median	Mean	3rd Q	Max
1	2.00	26.00	44.50	62.33	78.00	331.00
2	2.00	22.00	33.00	36.72	48.00	124.00
3	21.00	71.75	106.00	129.09	160.25	555.00
4	16.00	54.00	71.50	88.73	104.75	451.00

Table 10. Learning behaviors interpretation

Cluster	Characteristics
LC1 (Reflective exploring)	Actively explore and evaluate their errors.
	Use feedback to analyze both the problem and example components
	Balance between reflection and application
LC2 (Feedback-focused)	Prioritize understanding why their solution was incorrect
	By revisiting and analyzing example transitions
	Minimal engagement with new tasks
LC3 (Deep process-oriented)	Engage deeply with the example rotation process and feedback
	Focusing on understanding the rotation principles
LC4 (Task-oriented)	Task-oriented and action-driven
	Engage with feedback to inform immediate problem-solving efforts

indicating balanced engagement—participants process the task meaningfully without the exhaustive focus seen in Cluster 3.

The adjusted transition matrices in Figure 16 reflect the dynamic characteristics of transitions between different actions for each cluster. Compared to the overall transition pattern, Cluster 1 shows a higher likelihood of transitioning from Option Wrong to Correct, L2 to L1, and Correct to L1, while transitions from L to L1, L2 to L1, Correct to Wrong, and Q to Q decrease. Participants in this cluster focus on exploring multiple task components after receiving feedback, especially incorrect options, reflecting a reflective approach to error evaluation. Cluster 2 has significantly fewer self-transitions (L1 to L1, L2 to L2, Q to Q, and Wrong to Wrong) and a higher likelihood of transitioning from Option Correct to Option Wrong, Wrong to Correct, L1 to L2, Q to Option Correct, and L1 to Q. This strategy centers on feedback and its connection to the example's rotation process, with less focus on the new task, indicating time spent consolidating understanding of previous mistakes. Cluster 3 shows a higher likelihood of staying in L1, transitioning from L to L1, and Q to Q, with slightly fewer transitions from L1 to L2, L to Q, Q to Correct, Wrong to Correct, and Correct to Wrong. Cluster 4's transition pattern is close to the overall average, with slightly higher chances for Q to Q, Wrong to Wrong, and Wrong to Q, and a slightly lower chance for Wrong to Correct.

The above analysis suggests that the four clusters represent distinct learning strategies based on participants' eye fixation patterns after receiving feedback on their solution. These patterns reveal how participants process feedback, evaluate their original solution, and learn from the example in the spatial rotation task. Table 10 summarizes our interpretations of the four learning clusters.

5.4.3. RQ5: Association between learning behavior and item type

Figure 17 shows two mosaic plots comparing four types of learning behaviors and different item types across two learning blocks. Block 1 consists mainly of one-direction rotation items, which are easier to

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

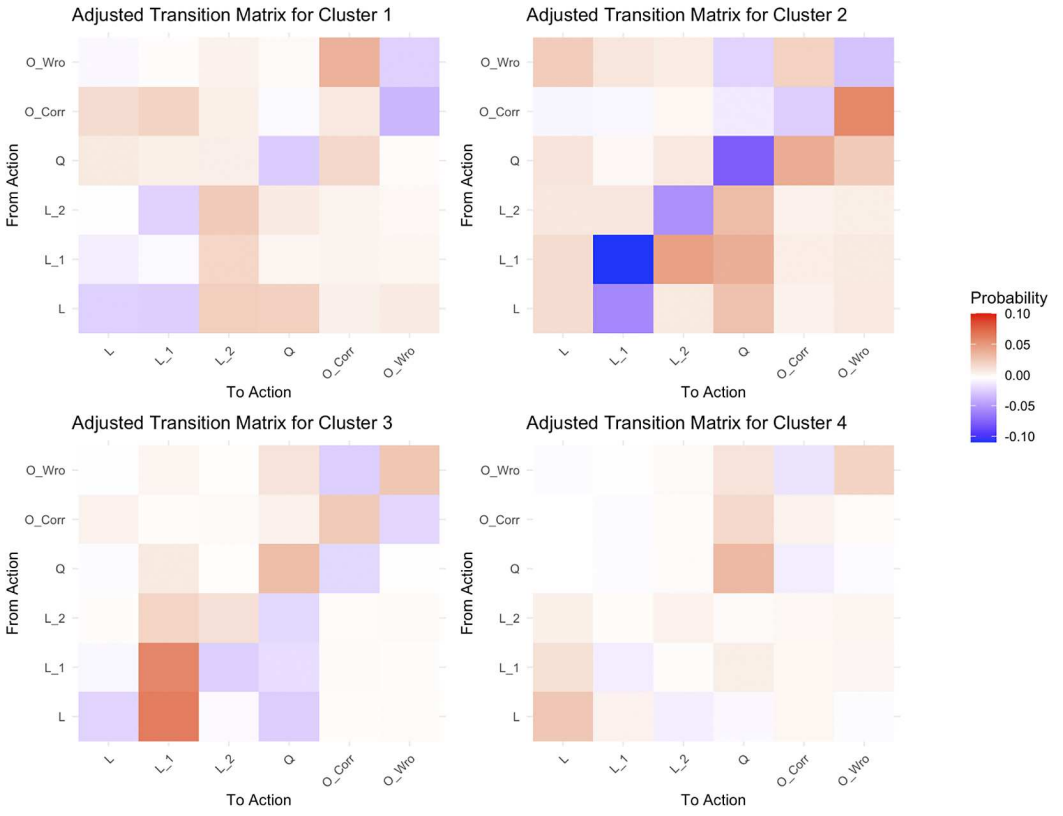

Figure 16. The adjusted transition matrix for four clusters.

visualize. Participants primarily use reflective (Cluster 1) or feedback-focused (Cluster 2) strategies after receiving feedback. Some also use the other two strategies for solving rotations like $x180$, $x90y90$, and $y180$. Block 2 introduces more complex two-direction rotations, where we observe a significant increase in deep process-oriented (Cluster 3) and task-oriented (Cluster 4) strategies. The χ^2 tests for both blocks show that learning strategy is significantly related to item type: $G^2(12) = 124.36$, $p < 0.001$, Cramer's $V = 0.219$ for Block 1 and $G^2(18) = 112.03$, $p < 0.001$, Cramer's $V = 0.214$ for Block 2.

5.4.4. RQ6: How learning behaviors impact the testing behaviors

Since test block 2 follows directly after learning block 2, we construct another 4 by 5 contingency table (Figure 18) to examine the relationship between participants' primary learning behaviors in learning block 2 and their primary problem-solving strategies in test block 2. It's important to note that each participant may exhibit different learning and testing behaviors across these blocks. The major strategy for each participant is defined as the one they use most frequently when solving or learning a question.

We observe that participants who primarily exhibit reflective exploring behavior (LC1) in learning block 2 are more likely to use Quick Navigation (TC1) and Balanced Problem Solving (TC5) strategies in test block 2. Participants with a feedback-focused learning behavior (LC2) tend to favor Quick Navigation (TC1). Those demonstrating a deep, process-oriented learning behavior (LC3) are more likely to use Example-Driven (TC2) strategy. Finally, participants with a task-oriented learning behavior (LC4) are more likely to use Quick Navigation (TC1), Example-Driven (TC2), and Balanced (TC5) problem-solving strategies. The association between these two types of clusters is statistically significant, with a moderate strength of association, $G^2(12) = 23.33$, $p = 0.03$, and Cramer's $V = 0.306$.

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

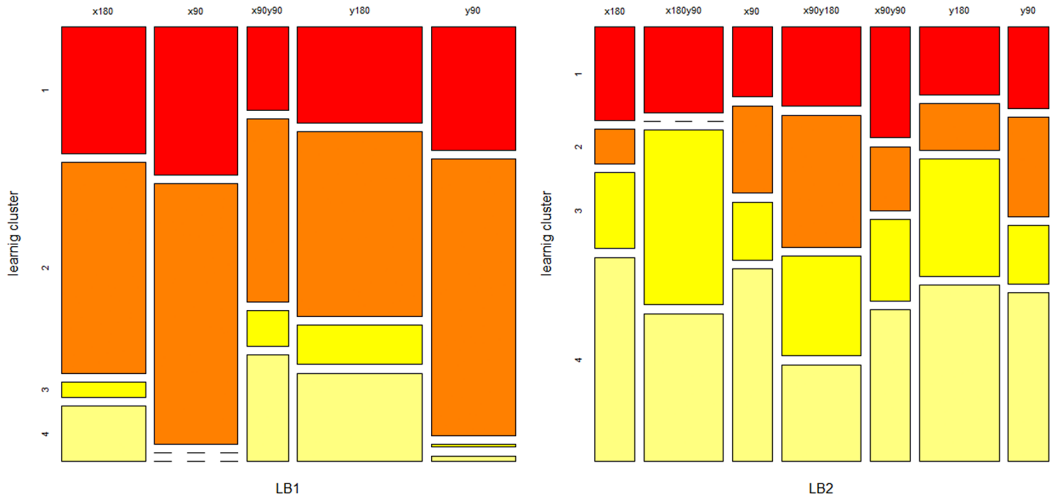

Figure 17. The mosaic plot of the two contingency tables.

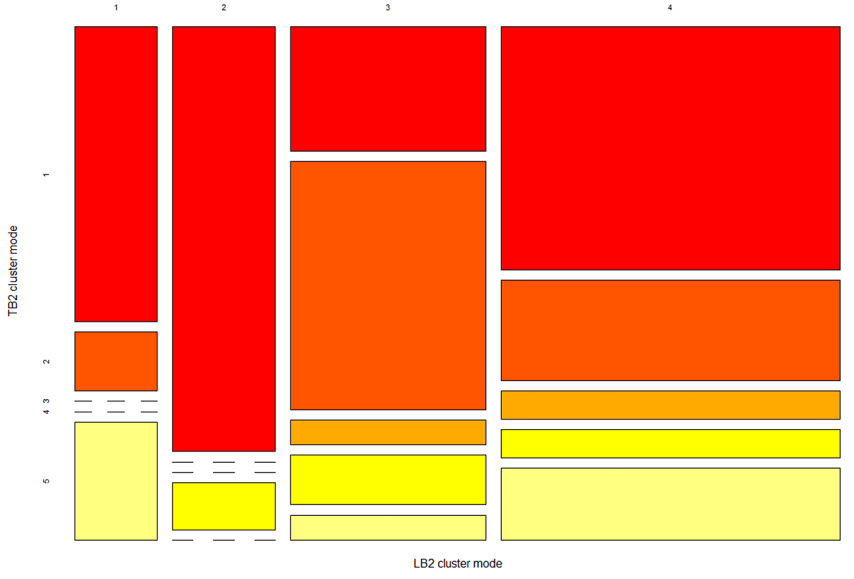

Figure 18. The plot of contingency table between LB2 cluster mode and TB2 cluster mode.

6. Conclusion and discussion

6.1. Contributions of this work

The major methodological contribution of this work is the T-LSTM Autoencoder framework, which extracts features from unstructured data with temporal and sequential dependencies. Our proposed T-LSTM Autoencoder differs from the original T-LSTM models in several key ways. First, our approach uses two time signals as inputs. Unlike prior models that only use elapsed time between events (Δt), we feed both elapsed time and event durations into the network by constructing the input $X_i = [E_i, \ell_i, d_i]$, where E_i are action embeddings, ℓ_i elapsed times, and d_i durations. Second, we employ a data-driven choice of the time-decay $g(\cdot)$ guided by empirical transition patterns, whereas Baytas et al. (2017) predefine this function, which may lead to the loss of important information present in real data. Third, we have a multi-target reconstruction objective. We jointly reconstruct actions and both time sequences

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

using $L = L_{CE} + \lambda L_{MSE}$, which regularizes latent features to capture what happened and when/how long. Finally, we discuss how to select hyperparameters and the weight function $g(.)$ through a three-way evaluation based on various evaluation metrics. Results in Sections 5.1 and 5.2 demonstrate the advantages of incorporating time duration and elapsed time to improve the reconstruction of categorical fixation sequences.

While primarily illustrated with an eye-tracking dataset, our feature extraction method can be applied to any sequential data with temporal and interdependent structures, such as process and log data (e.g., Bergner & von Davier, 2019; Chen, 2020; Fang & Ying, 2020; Greiff et al., 2013; He et al., 2016, 2023; LaMar, 2018; Liu et al., 2018; Qiao & Jiao, 2018; Tang, 2023; Ulitzsch et al., 2021; Ulitzsch, Ulitzsch, et al., 2023). The evaluation framework presented here can assist researchers in selecting the best Autoencoder model for their specific applications. For instance, in our case, where the goal is to reconstruct the eye-fixation sequence using time information (duration and elapsed time), the λ parameter in the loss function (15) is set within a small range. For applications where time information plays a more significant role, a larger λ can be used.

The additional methodological contribution is demonstrating how different statistical tools can be leveraged to interpret feature extraction results in the context of educational questions. Features extracted from process data are often difficult to interpret, so we use a set of statistical tools to summarize patterns and investigate the associations between variables derived from these features. The results from these analyses address key educational questions, such as evaluating the effectiveness of test questions in assessing participants' spatial rotation skills (examining changes in testing behaviors (E1.2) and their association with test scores (E1.3)) and the effectiveness of learning interventions in improving those skills (exploring the association between learning interventions and testing behaviors (E2.3)). These findings also have implications for the development of new psychometric models and the design of learning programs. For example, the current learning program provides the same test questions and interventions to all participants. However, results from this study suggest that participants employ different problem-solving strategies and learning behaviors, which are significantly related to specific types of questions. Based on these findings, an adaptive learning program could be developed, tailored to individual testing and learning behaviors, to improve overall learning efficiency.

6.2. Future directions

Our proposed two-component DAK can be further developed through the following perspectives. First, as one of the reviewers mentioned that, it is possible to extend our work to learn the decay coefficient α adaptively during training rather than pre-specifying it. This would allow the model to automatically discover the most appropriate temporal decay rate for different datasets.

Second, we can strengthen our current T-LSTM framework by using some complementary techniques in other T-LSTM models. For example, Nguyen et al. (2021) retain the T-LSTM cell but introduce cost-sensitive learning to address class imbalance. Our objective could be extended analogously with class-weighting to handle imbalanced action distributions. Zhang (2019) proposes an attention-based T-LSTM that aggregates multiple past states with temporal decay. A similar temporal-attention module could be integrated into our encoder–decoder framework to emphasize salient events.

As suggested by one reviewer, we have conducted a small preliminary experiment with an attention-based autoencoder (Appendix B of the Supplementary Material). Results show that with our current dataset size and model scale, Transformer architectures did not perform competitively compared to LSTM-based approaches. This result is consistent with the broader literature indicating that attention-based models typically require both larger model capacity and substantially larger training corpora to realize their full potential (Brown et al., 2020; Haque et al., 2024; Kaplan et al., 2020). Nevertheless, attention mechanisms remain a promising avenue for modeling eye-tracking sequences, particularly given their ability to capture both local dependencies and long-range interactions. Future work could explore scaling up Transformer architectures, adopting relative or learned positional encodings tailored for sequential fixation data, and leveraging larger or pretraining datasets (e.g., synthetic or cross-task

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

gaze corpora). Such extensions may enable attention-based models to complement or even surpass recurrent architectures in this domain.

Supplementary material. The supplementary material for this article can be found at https://doi.org/10.1017/psy.2026.10096.

AI statement. The author utilized ChatGPT5 to help with manuscript writing process. Generative artificial intelligence was used to rephrase the authors' original written context, which is mainly for language polishing purposes. The authors reviewed the created content produced by this generative AI Tool and assert that the content within this document is factually accurate and free of plagiarism. The authors take full responsibility for the submitted document.

Funding statement. This research was partially supported by NSF SES-2243044 and SES-2051198.

Competing interests. The authors declare that they have no competing interests.

Ethical standards. The authors declared no potential competing interests with respect to the research, authorship, and/or publication of this article. This research does not involve human participants and/or animals.

References

Almeida, L. B. (2020). Multilayer perceptrons. In E. Fiesler & R. Beale (Eds.), *Handbook of neural computation* (pp. C1–C2). CRC Press.

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, *67*(1), 1–48.

Baytas, I. M., Xiao, C., Zhang, X., Wang, F., Jain, A. K., & Zhou, J. (2017). Patient subtyping via time-aware LSTM networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 65–74), Association for Computing Machinery.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, *3*, 1137–1155.

Bergner, Y., & von Davier, A. A. (2019). Process data in NAEP: Past, present, and future. *Journal of Educational and Behavioral Statistics*, *44*(6), 706–732.

Bogarín, A., Cerezo, R., & Romero, C. (2018). A survey on educational process mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(1), e1230.

Breslow, N. E., & Clayton, D. G. (1993). Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association*, *88*(421), 9–25.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C.,... Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901.

Chen, Y. (2020). A continuous-time dynamic choice measurement model for problem-solving process data. *Psychometrika*, *85*(4), 1052–1075.

Fang, G., & Ying, Z. (2020). Latent theme dictionary model for finding co-occurrent patterns in process data. *Psychometrika*, *85*(3), 775–811.

Greiff, S., Wüstenberg, S., Holt, D. V., Goldhammer, F., & Funke, J. (2013). Computer-based assessment of complex problem solving: Concept, implementation, and application. *Educational Technology Research and Development*, *61*(3), 407–421.

Guo, C., & Berkhahn, F. (2016). *Entity embeddings of categorical variables*. Preprint, arXiv:1604.06737.

Hahn, L., & Klein, P. (2025). Clustering eye-movement data uncovers students' strategies for coordinating equations and diagrams of vector fields. *Educational Studies in Mathematics*, *118*(3), 359–385.

Hancock, J. T., & Khoshgoftaar, T. M. (2020). Survey on categorical data for neural networks. *Journal of Big Data*, *7*(1), 28.

Haque, S. T., Debnath, M., Yasmin, A., Mahmud, T., & Ngu, A. H. H. (2024). Experimental study of long short-term memory and transformer models for fall detection on smartwatches. *Sensors*, *24*(19), 6235.

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.

He, Q., Shi, Q., & Tighe, E. L. (2023). Predicting problem-solving proficiency with multiclass hierarchical classification on process data: A machine learning approach. *Psychological Test and Assessment Modeling*, *65*(1), 145–177.

He, Q., von Davier, M., & Han, Z. (2016). Analyzing process data from game/scenario-based tasks: An information-theoretic approach. *Psychological Test and Assessment Modeling*, *58*(3), 377–398.

He, Q., von Davier, M., & Han, Z. (2019). *Exploring process data in problem-solving items in computer-based large-scale assessments: Case studies in PISA and PIAAC*. Information Age Publishing, Inc.

He, S., & Cui, Y. (2025). A systematic review of the use of log-based process data in computer-based assessments. *Computers in Education*, *228*, 105245.

Hihi, S., & Bengio, Y. (1995). Hierarchical recurrent neural networks for long-term dependencies. *Advances in Neural Information Processing Systems*, *8*, 493–499.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504–507.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*. *9*(8), 1735–1780. MIT-Press.

Irie, K., Tüske, Z., Alkhouli, T., Schlüter, R., & Ney, H. (2016). LSTM, GRU, highway and a bit of attention: An empirical overview for language modeling in speech recognition. In *Proceedings of Interspeech 2016* (pp. 3519–3523). International Speech Communication Association (ISCA).

Ivanová, L., Laco, M., & Benesova, W. (2022). Unsupervised clustering-based analysis of the measured eye-tracking data. In W. Osten, D. Nikolaev, & J. Zhou (Eds.), *Fourteenth international conference on machine vision (ICMV 2021)* (Vol. *12084*, pp. 53–60). SPIE.

Kaczorowska, M., Plechawska-Wójcik, M., & Tokovarov, M. (2021). Interpretable machine learning models for three-way classification of cognitive workload levels for eye-tracking features. *Brain Sciences*, *11*(2), 210.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020). *Scaling laws for neural language models*. Preprint, arXiv:2001.08361.

Križanić, S. (2020). Educational data mining using cluster analysis and decision tree technique: A case study. *International Journal of Engineering Business Management*, *12*, 1847979020908675.

LaMar, M. M. (2018). Markov decision process measurement model. *Psychometrika*, *83*(1), 67–88.

Liashchynskyi, P., & Liashchynskyi, P. (2019). *Grid search, random search, genetic algorithm: a big comparison for NAS*. Preprint, arXiv:1912.06059.

Lindemann, B., Maschler, B., Sahlab, N., & Weyrich, M. (2021). A survey on anomaly detection for technical systems using LSTM networks. *Computers in Industry*, *131*, 103498.

Liu, H., Liu, Y., & Li, M. (2018). Analysis of process data of PISA 2012 computer-based problem solving: Application of the modified multilevel mixture IRT model. *Frontiers in Psychology*, *9*, 1372.

Liu, X., & Cui, Y. (2025). Eye tracking technology for examining cognitive processes in education: A systematic review. *Computers & Education*, *229*, 105263.

Man, K., & Harring, J. R. (2023). Detecting preknowledge cheating via innovative measures: A mixture hierarchical model for jointly modeling item responses, response times, and visual fixation counts. *Educational and Psychological Measurement*, *83*(5), 1059–1080.

Mikolov, T. (2013). *Efficient estimation of word representations in vector space*. Preprint, arXiv:1301.3781.

Min, W., Frankosky, M. H., Mott, B. W., Rowe, J. P., Smith, A., Wiebe, E., Boyer, K. E., & Lester, J. C. (2019). DeepStealth: Game-based learning stealth assessment with deep neural networks. *IEEE Transactions on Learning Technologies*, *13*(2), 312–325.

Nazareth, A., Killick, R., Dick, A. S., & Pruden, S. M. (2019). Strategy selection versus flexibility: Using eye-trackers to investigate strategy use during mental rotation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *45*(2), 232.

Nguyen, A., Chatterjee, S., Weinzierl, S., Schwinn, L., Matzner, M., & Eskofier, B. (2021). Time matters: Time-aware LSTMs for predictive business process monitoring. In S. Leemans & H. Leopold (Eds.), *International conference on process mining, ICPM 2020* (pp. 112–123). Springer.

Pardos, Z. A., Baker, R. S., San Pedro, M. O., Gowda, S. M., & Gowda, S. M. (2014). Affective states and state tests: Investigating how affect and engagement during the school year predict end-of-year learning outcomes. *Journal of Learning Analytics*, *1*(1), 107–128.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In S. Dasgupta & D. McAllester (Eds.), *Proceedings of Machine Learning Research*, (Vol. *28*, pp. 1310–1318). JMLR.org.

Peña-Ayala, A. (2014). Educational data mining: A survey and a data mining-based analysis of recent works. *Expert Systems with Applications*, *41*(4), 1432–1462.

Qiao, X., & Jiao, H. (2018). Data mining techniques in analyzing process data: A didactic. *Frontiers in Psychology*, *9*, 374277.

Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, *124*(3), 372.

Rayner, K. (2009). Eye movements and attention in reading, scene perception, and visual search. *The Quarterly Journal of Experimental Psychology*, *62*(8), 1457–1506.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386.

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.

Ruder, S. (2016). *An overview of gradient descent optimization algorithms*. Preprint, arXiv:1609.04747.

Schnipke, D. L., & Scrams, D. J. (1997). Modeling item response times with a two-state mixture model: A new method of measuring speededness. *Journal of Educational Measurement*, *34*(3), 213–232.

https://doi.org/10.1017/psy.2026.10096 Published online by Cambridge University Press

Shutaywi, M., & Kachouie, N. N. (2021). Silhouette analysis for performance evaluation in machine learning with applications to clustering. *Entropy*, *23*(6), 759.

Song, M., Günther, C. W., & Van der Aalst, W. M. (2008). Trace clustering in process mining. In D. Ardagna, M. Mecella, & J. Yang (Eds.), *International conference on business process management, BPM 2008* (pp. 109–120). Springer.

Tang, X. (2023). A latent hidden Markov model for process data. *Psychometrika*, *89*(1), 205–240.

Tang, X., Wang, Z., He, Q., Liu, J., & Ying, Z. (2020). Latent feature extraction for process data via multidimensional scaling. *Psychometrika*, *85*(2), 378–397.

Tang, X., Zhang, S., Wang, Z., Liu, J., & Ying, Z. (2021). ProcData: An R package for process data analysis. *Psychometrika*, *86*(4), 1058–1083.

Ulitzsch, E., He, Q., Ulitzsch, V., Molter, H., Nichterlein, A., Niedermeier, R., & Pohl, S. (2021). Combining clickstream analyses and graph-modeled data clustering for identifying common response processes. *Psychometrika*, *86*(1), 190–214.

Ulitzsch, E., Molter, H., & Pohl, S. (2022). Using partial credit model-based regex features to track partial correctness in complex problem-solving processes. *Psychological Test and Assessment Modeling*, *64*, 319–349.

Ulitzsch, E., Ulitzsch, V., He, Q., & Lüdtke, O. (2023). A machine learning-based procedure for leveraging clickstream data to investigate early predictability of failure on interactive tasks. *Behavior Research Methods*, *55*(3), 1392–1412.

van Dongen, B. F., & Adriansyah, A. (2009). Process mining: Fuzzy clustering and performance visualization. In S. Rinderle-Ma, S. Sadiq, & F. Leymann (Eds.), *International conference on business process management, BPM 2009* (pp. 158–169). Springer.

Wang, Q., Mousavi, A., Lu, C., and Gao, Y. (2023). Examining adults' behavioral patterns in a sequence of problem solving tasks in technology-rich environments. *Computers in Human Behavior*, *147*, 107852.

Wang, S., Hu, Y., Wang, Q., Wu, B., Shen, Y., & Carr, M. (2020). The development of a multidimensional diagnostic assessment with learning tools to improve 3-D mental rotation skills. *Frontiers in Psychology*, *11*, 305.

Wang, S., Wu, S., Chen, Y., Fang, L., Xiao, L., & Li, F. (2026). Exploring latent constructs through multimodal data analysis. *Journal of Educational Measurement.*, *63*(1), e12412.

Wang, S., Zhang, S., & Shen, Y. (2020). A joint modeling framework of responses and response times to assess learning outcomes. *Multivariate Behavioral Research*, *55*(1), 49–68.

Ward, J. H. , Jr. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, *58*(301), 236–244.

Xue, J., Li, C., Quan, C., Lu, Y., Yue, J., & Zhang, C. (2017). Uncovering the cognitive processes underlying mental rotation: An eye-movement study. *Scientific Reports*, *7*(1), 10076.

Yaneva, V., Clauser, B. E., Morales, A., & Paniagua, M. (2021). Using eye-tracking data as part of the validity argument for multiple-choice questions: A demonstration. *Journal of Educational Measurement*, *58*(4), 515–537.

Yaneva, V., Clauser, B. E., Morales, A., & Paniagua, M. (2022). Assessing the validity of test scores using response process data from an eye-tracking study: A new approach. *Advances in Health Sciences Education*, *27*(5), 1401–1422.

Yang, M., Cai, C., & Hu, B. (2022). Clustering based on eye tracking data for depression recognition. *IEEE Transactions on Cognitive and Developmental Systems*, *15*(4), 1754–1764.

Yin, Y., Alqahtani, Y., Feng, J. H., Chakraborty, J., & McGuire, M. P. (2021). Deep learning methods for the prediction of information display type using eye tracking sequences. In M. A. Wani, I. K. Sethi, W. Shi, G. Qu, D. S. Raicu, & R. Jin (Eds.), *2021 20th IEEE international conference on machine learning and applications (ICMLA)* (pp. 601–605). IEEE.

Yin, Z., & Shen, Y. (2018). On the dimensionality of word embedding. *Advances in Neural Information Processing Systems*, *31*, 895–906.

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. *Neural Computation*, *31*(7), 1235–1270.

Zhang, S., Li, A., & Wang, S. (2023). Exploration of latent structure in test revision and review log data. *Educational Measurement: Issues and Practice*, *42*(4), 53–65.

Zhang, S., Wang, Z., Qi, J., Liu, J., & Ying, Z. (2022). Accurate assessment via process data. *Psychometrika*, *88*(1), 76–97.

Zhang, T., Taub, M., & Chen, Z. (2022). A multi-level trace clustering analysis scheme for measuring students' self-regulated learning behavior in a mastery-based online learning environment. In *LAK22: 12th international learning analytics and knowledge conference* (pp. 197–207), Association for Computing Machinery.

Zhang, Y. (2019). *Attain: Attention-based time-aware LSTM networks for disease progression modeling*. In S. Kraus (Ed.), *proceedings of the 28th international joint conference on artificial intelligence (IJCAI-2019)* (pp. 4369–4375). Macao.

Zhu, M., & Feng, G. (2015). An exploratory study using social network analysis to model eye movements in mathematics problem solving. In *Proceedings of the fifth international conference on learning analytics and knowledge* (pp. 383–387), Association for Computing Machinery.

Cite this article: Fang, L., Wang, S., Chen, Y., Zhang, S., Liu, Z. and Zhong, W. (2026). Analyzing Complex Educational Data: A Data Analytic Framework for Integrating Structured and Unstructured Eye-Tracking Data. *Psychometrika*, 839–867. https://doi.org/10.1017/psy.2026.10096