applied sciences

MDPI

Article

Deep Knowledge Tracing Integrating Temporal Causal Inference and PINN

Faming Lu [1], Yingran Li [1] and Yunxia Bao [2,*]

[1] College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China; fm_lu@163.com (F.L.); lyran1129@163.com (Y.L.)
[2] College of Mathematics and System Science, Shandong University of Science and Technology, Qingdao 266590, China
* Correspondence: baoyunxia98@163.com; Tel.: +86-15092013016

Abstract: Knowledge tracing predicts students' future performance based on their historical performance data, which is significant for students' learning resource recommendation, learning path prediction, and other aspects. Students' knowledge mastery, learning ability, and question difficulty all influence the performance metrics of knowledge tracing. This paper proposes a deep knowledge tracing model that integrates temporal causal inference and the PINN (Physics-Informed Neural Network) model. The model first uses the temporal causality model to explores the causal relationships between students' knowledge points, which is then combined with the deep learning-based knowledge tracing model for prediction. Next, it treats the logical model as a 'physical model', adds a loss term, considers the confounding factors caused by students' answer preferences, and adjusts students' learning ability through backdoors to obtain more accurate predictions. In the public education datasets ASSISTment2012 and ASSISTchall, the predictive performance of the TLPKT-PINN model is superior to some classical models and LPKT. From the experimental results, we can conclude that considering the degree of mastery of causal knowledge points and adjusting the loss term can improve the accuracy of predicting student grades.

Keywords: smart education; knowledge tracking; temporal causality; causal inference; learning process

check for updates

Academic Editor: Douglas O'Shaughnessy

Received: 20 December 2024
Revised: 26 January 2025
Accepted: 30 January 2025
Published: 1 February 2025

Citation: Lu, F.; Li, Y.; Bao, Y. Deep Knowledge Tracing Integrating Temporal Causal Inference and PINN. *Appl. Sci.* **2025**, *15*, 1504. https://doi.org/10.3390/app15031504

1. Introduction

With the increasing emergence of online learning tutorials, students can study courses through the internet. AI-powered educational technologies (AI-EdTech) are increasingly being used to automate and scaffold learning activities [1]. Smart learning systems offer new ways of acquiring knowledge and have been expanding their popularity and influence over recent decades [2]. More and more traditional classrooms are integrating with online education, leading to a continuous accumulation of data from online learning activities. These data can be used in scenarios such as student knowledge state identification, learning performance prediction, exercise recommendation, etc. Students can autonomously solve problems based on their interests or teachers' requirements, and check their mastery levels to engage in targeted learning. Teachers can also monitor students' mastery levels to optimize their teaching methods. Knowledge tracing is a key research area in educational data mining, and has attracted a lot of attention from academics [3]. Knowledge tracking is the task of tracking students' knowledge status based on their learning activities, and it is mainly divided into Bayesian-based knowledge tracing and deep learning-based knowledge tracing.

The classic Bayesian Knowledge Tracing (BKT) model [4] estimates students' mastery of knowledge points through a probability model, using a Hidden Markov Model (HMM) [5] to treat students' knowledge states as hidden variables. BKT can based on the state transition matrix to predict the next knowledge state. Its advantages include that it can dynamically update a student's knowledge status to deal with uncertainty, and provide personalized learning recommendations. However, BKT also has some drawbacks, such as the sensitivity to model parameters, which can lead to the need for large amounts of historical data to model accurately. On large-scale datasets, its computational complexity is high. And the model defaults to the idea that students will not forget what they have done, which is not true. With the development of deep learning, knowledge tracing models have begun to integrate with deep learning techniques. The Deep Knowledge Tracing (DKT) model [6] utilizes recurrent neural networks (RNNs) [7] to capture students' learning processes, predicting student's mastery of knowledge points based on their sequence of responses. The advantages of DKT include its ability to handle complex time series data and capture long-range dependencies, thereby providing personalized learning recommendations. However, some deep learning-based knowledge tracing models may lack interpretability. And it models one skill at a time, ignoring the relationships between different skills.

In recent years, the Learning Process-consistent Knowledge Tracing (LPKT) model [8] has been used to track knowledge by simulating the evolving paths of knowledge mastery during the student learning process. This model can capture and utilize the dynamic changes in students' learning processes, providing more precise personalized learning support. However, this model only considers the relationship between knowledge states and exercises, without accounting for the causal relationships between knowledge points. According to real-life teaching, we know that there is a causal relationship between knowledge points, such as the causal relationship between addition and multiplication, where addition is the cause of multiplication and multiplication is the result of addition. When students are learning a certain knowledge point, their mastery of causal knowledge points will definitely affect their mastery of that knowledge point. So, when predicting students' grades, we need to consider the causal relationship between knowledge points. However, the LPKT model does not consider this relationship.

In our previous work, we utilized a causal inference algorithm to mine causal relationships between the levels of knowledge mastery, and established rules to identify the strength of causal effects, but we did not consider the time factor. The temporal causal model considers time factors, captures the dynamic characteristics of causal relationships over time, describes long-term effects and feedback effects, adapts to complex temporal systems, and can thus obtain more accurate causal relationships between knowledge points. Additionally, regarding the prediction results, they may be contrary to the principle that "if a student performs well, then his corresponding learning ability is high; if a student has a high mastery level of knowledge points, he may like to do relatively difficult exercises". Based on the LPKT model, we propose a deep learning knowledge tracing framework called the TLPKT_PINN model based on the temporal causality model and optimize the loss function by incorporating physical loss. We conducted experiments on two real datasets, and the results indicate that our proposed framework is better at accurately tracking the evolution of knowledge states. The innovations in our work are as follows:

(1) We use a temporal causal model to explore the relationships between the knowledge points, which can be combined with the general mastery levels of students to derive the final knowledge state influenced by causally related knowledge points.

(2) By implementing backdoor adjustment, we can obtain students' learning abilities and problem difficulty levels, effectively removing the confounding factors related to

students' preferences. It can improve the prediction of students' learning abilities and problem difficulties.

(3) We add the physical loss term. We also increase the accuracy of predicting student performance by increasing the penalty for violating common-sense patterns in the student learning process.

2. Related Work

2.1. Knowledge Tracing

Knowledge tracing is a well-known problem in AI for education, consisting of monitoring how the knowledge state of students changes during the learning process and accurately predicting their performance in future exercises [9].

Knowledge tracing can be traced back to the late 20th century, with Corbett and Anderson (1995) [4] proposing BKT. BKT uses a HMM to treat students' knowledge states as hidden variables and predicts the next state based on a state transition matrix. The BKT model primarily relies on four parameters: the probability that a student has mastered the knowledge point before answering questions related to it; the probability that the student will transition from not mastering to mastering the knowledge point on their next attempt; the probability that the student has mastered the knowledge point but answers related questions correctly (the guessing probability); and the probability that the student has mastered the knowledge point but answers incorrectly (the slipping probability). Based on these four parameters, the model can further characterize the student's response to the next question using conditional probabilities. Subsequently, BKT has a lot of extensions. Käser et al. [10] proposed the Dynamic Bayesian Knowledge Tracing (DBKT) model, which utilizes Dynamic Bayesian Networks (DBNs) to capture the dynamic changes in students' knowledge states during the learning process, then leading to more accurate student modeling and personalized education. DE BAKER et al. [11] introduced the Three-Learning-State BKT (TLS-BKT) model, which divides the learning process into three states through an evaluation function. This model improves the binary node states in the BKT model, transforming the original "not mastered/mastered" states into "not mastered/learning/mastered" states, enhancing the model's flexibility and robustness.

In recent years, an increasing number of researchers have introduced the deep learning into knowledge tracing to enhance its expressiveness and performance. A typical model based on deep learning is DKT proposed by Piech et al. RNNs based on long short-term memory (LSTM), These networks can learn sequences of observations, thus causing the model to be well suited for time series applications [12]. DKT was the first to apply deep learning to knowledge tracing, allowing it to capture more complex student knowledge states without the need for explicit manual encoding of knowledge domains. It uses an RNN as a hidden unit to generate knowledge state vectors that represent students' knowledge states, and it outputs predicted student responses through a sigmoid linear layer. The LSTM mechanism, with its inherent "gate" design, can effectively extract the features and correlations of multiple time sequences [13]. Due to the interpretability issues of inputs and outputs, deep learning knowledge tracing has seen numerous improved models. For example, the Dynamic Key–Value Memory Network (DKVMN) [14] incorporates an attention mechanism, allowing it to utilize the relationships between foundational concepts and directly output students' mastery levels for each concept. Graph-based Knowledge Tracing (GKT) [15] is a knowledge tracing method based on graph neural networks, aimed at capturing the complex relationships between knowledge by constructing knowledge graphs and leveraging the powerful representation learning capabilities of graph neural networks to model students' learning processes. The Context-aware Attentive Knowledge Tracing (AKT) [16] model constructs context-aware representations of questions and answers, using a monotonic attention mechanism to summarize students' past performances over time scales and employing the Rasch model to capture individual differences among questions covering the same concept. At present, deep learning technology has achieved state-of-the-art results in the processing of Euclidean data [17]. Lyu et al. [18] proposed a DKT-STDRL model that utilizes Convolutional Neural Networks to extract spatial features from students' learning sequences and LSTM to process temporal features.

2.2. Temporal Causal Inference

In real life, the occurrence of most events is not at regular intervals but is discrete. A point process is a powerful modeling tool for event sequences, consisting of a time series [19] of binary events occurring in continuous time. Point processes have been successfully applied in various fields such as social networks, finance, equipment maintenance, and electronic health records. The characteristic feature of point process models is their intensity function. The occurrence times of various types of events $\varepsilon = 1, \ldots, \mathrm{E}$ are unevenly distributed. A multivariate point process containing E type of events can be represented using a counting process $\{N_e\}_{\{e=1\}}^{E}$, where $N_e = \{N_e(t)|t \in [0,T]\}$. The intensity function shown in Equation (1) can be defined for type u as the expected instantaneous rate of occurrence of type e events given the history. Each intensity function captures the instantaneous occurrence rate of a class of events conditioned on historical events.

$$\lambda_e(t) = \frac{\mathbb{E}(dN_e(t)|\mathcal{H}_t)}{dt} \tag{1}$$

where $\mathcal{H}_t = \{t_i, e_i | t_i < t, e_i \in \varepsilon\}$ representing all types of events that occurred before time t.

Granger causality [20] emphasizes the temporal sequence of events and was originally used to study the dependency structure of multivariate time series. It has also been extended to multi-type event sequences [21]. In short, to test the causality between A and B, let Ω_n be the set of all information up to time $(t = 1, 2, \ldots, n)$ (including information other than A and B), and let B_n be all the information about B up to time n. Here, both B_n and Ω_n are multivariate random variables, and $B_\mathrm{n} \in \Omega_n$. Thus, $\Omega_n - B_\mathrm{n}$ represents all information up to time n excluding B. Assume that the following: (1) the present and the past can influence the future, but the future cannot influence the past; (2) Ω_n does not contain any redundant information, meaning that if a variable Z_n is functionally related to one or more other variables, it should be removed from Ω_n. If $P(A_{n+1}|\Omega_n) \neq P(A_{n+1}|\Omega_n - B_n)$, then variable B is considered a cause of variable A, indicating that B_n contains unique information that influences the occurrence of A_{n+1}. Intuitively, for event sequence data, if including historical events of one type improves the prediction of future events of another type, we say that one type of event has Granger causality with respect to the other type.

The Hawkes process [22] assumes that past events can independently and additively influence the occurrence of future events, specifically through a collection of pairwise kernel functions. The Hawkes process is a mathematical model used to model self-exciting processes; it is a counting process that describes a series of events occurring over time. In this process, the occurrence of each event increases the likelihood of the next event happening, while the effect of this stimulation gradually diminishes over time.

The core concept of the Hawkes process is the conditional intensity function, it represents the expected frequency of an event occurring near a certain point in time, given all past events. Its mathematical expression is as follows:

$$\lambda^*(t) = \mu + \sum_{t_i<t} g(t - t_i) \tag{2}$$

where $\lambda^*(t)$ is the conditional intensity function, μ is the background intensity, $g(t - t_i)$ is the triggering kernel, and t_i is the time of the i-th event. The triggering kernel can take various forms, with one common type being the exponential decay function, which indicates that the stimulating effect of past events on the current time point decreases exponentially as the time interval increases.

An event sequence records the occurrence of a specific type of event. Point processes can be used to describe event sequences. A set of point processes can be input, where each point process represents an event sequence, and the output is a causal graph established by the different processes. In the causal graph, each node represents a point process, and each directed edge captures the directed interaction from one point process to another.

2.3. PINN Model

In the field of computational science, surrogate models currently play an increasingly important role, especially in situations that require handling complex physical processes and large data analysis. Neural networks are becoming increasingly popular tools, being widely applied to simulate the complex systems. However, traditional neural network models often have shortcomings in strictly adhering to physical laws. Recent research [23] has combined high-fidelity and low-fidelity data, using data fusion techniques to reduce noise impact and improve solution accuracy. This method effectively integrates data of varying quality when dealing with complex systems in the real world, thereby enhancing prediction performance.

M. Raissi [24] combined the Gaussian process model with the linear differential equation. This method can automatically infer the parameters and structure of differential equations from data, addressing uncertainties in the process, thereby improving predictive performance. H. Owhadi [25] employed the Gaussian process [26] regression to design function representations for given linear operators, accurately inferring solutions and providing uncertainty estimates for several prototype problems in mathematical physics. The Physics-Informed Neural Networks (PINN) model [27] is a type of machine learning model that combines deep learning with physical knowledge. Unlike traditional data-driven neural networks, PINN models use physical laws to guide the model during training, thereby enhancing its generalization capability. This is particularly evident in situations where data are scarce or noisy. The PINN model is typically composed of a deep neural network. Its characteristic is the inclusion of physics-informed terms in the loss function, which represent the physical laws being followed. In traditional machine learning methods, the learning process is primarily data-driven, and models heavily rely on large amounts of high-quality data. However, in practical applications, there often exist challenges such as data scarcity or noisy data. In such cases, relying solely on data-driven models makes it difficult to achieve accurate and reliable predictions. Taking a one-dimensional damped harmonic oscillator as an example, the working principle of the PINN model is as follows: First, define the physical problem and the corresponding physical laws, such as the motion equation of the harmonic oscillator. The inputs are time t and position x, while the outputs are velocity v and acceleration a of the neural network. The loss function of the model consists of data error terms and physics-informed error terms. The data error terms measure the difference between predicted values and actual observations, while the physics-informed error terms ensure that the predictions comply with the relevant physical laws. By optimizing the loss function, the neural network is trained to ultimately produce predictions that adhere to the physical laws.

3. TLPKT_PINN Model

3.1. Overall Framework of the Model

In the learning process of students, their learning activities consist of a series of questions they pose and the corresponding answers. For a group of students $S = \{s_1, \ldots, s_i\}$, a set of exercises $E = \{e_1, \ldots, e_i\}$, and a set of knowledge concepts $K = \{k_1, \ldots, k_i\}$, student i spends time at at time point t answering the exercise e derived from the knowledge concept k, and receives a response $a \in \{0, 1\}$, where $a = 1$ indicates that the student answered the question correctly, and $a = 0$ indicates that the student answered the question incorrectly. Therefore, for a student, we can define their learning process as follows:

$$x = \{(e_1, at_1, a_1, t_1), it_1, (e_2, at_2, a_2, t_2), it_2, \ldots, (e_n, at_n, a_n, t_n), it_n\}$$

where (e_n, at_n, a_n, t_n) represents a fundamental learning unit in the student's learning process. Here, e_t is the exercise, at_t is the time spent by the student in answering the exercise e_t, a_t indicates the binary correctness label (1 for correct, 0 for incorrect), and it_t represents the interval time between learning units.

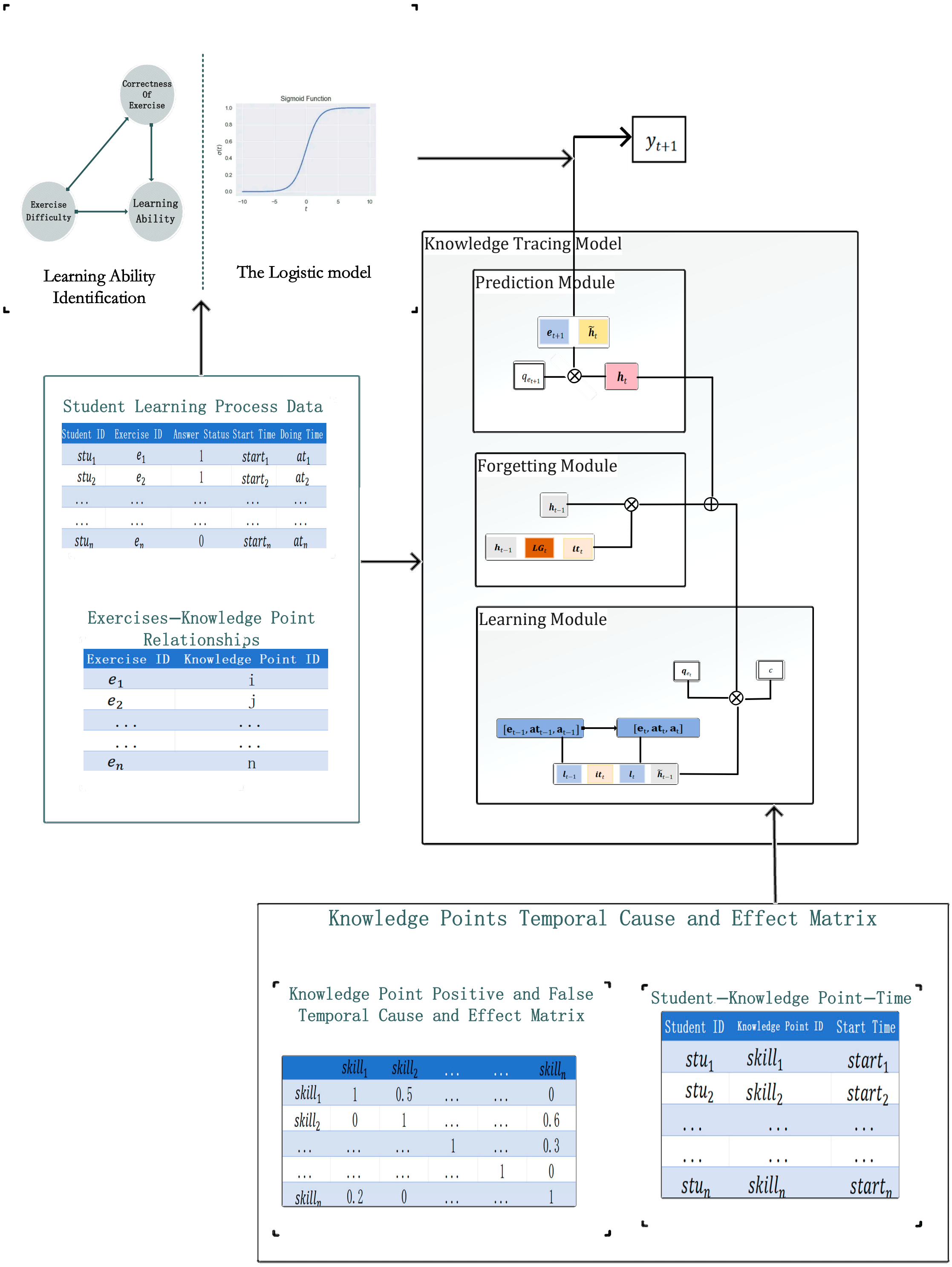
Since each exercise is associated with the specific knowledge points, we use the Q-matrix with 0 and 1 to represent the relationship between exercises and knowledge points. Specifically, if exercise e_j requires knowledge point k_m, then $Q_{\mathrm{j}m}= 1$; otherwise, $Q_{\mathrm{j}m}= 0$.

Our deep learning knowledge tracking model is shown in Figure 1, which includes three modules: learning module, forgetting module, and prediction module. The specific implementation will be explained in the following chapters.

The innovations of this model are as follows:

- Temporal causal knowledge point relationship mining: Inputting student ID, problem-solving time, and the correct or incorrect situation of knowledge points, the intensity function between knowledge points is obtained. Then, the intensity function is combined with Granger causality to finally obtain the temporal causal relationships between knowledge points. By using temporal causality to obtain the causal relationship between knowledge points, the causal relationship between knowledge points is combined with the student's mastery level of knowledge points output by the knowledge tracking model to obtain the final student's mastery level of knowledge points, which is finally used for predicting student grades;
- Calculating students' learning ability and the difficulty of exercises: Students with different learning abilities have varying preferences for answering questions. Students with stronger learning abilities tend to prefer more relatively difficult questions, which may result in a lower accuracy rate. However, we should not underestimate their learning potential because of this. Similarly, students with lower learning abilities may also have a low accuracy rate on simple exercises, and we cannot therefore overestimate the difficulty of the question. To more accurately assess students' learning abilities and the difficulty of exercises, we constructed a prior causal model for learning ability and exercise difficulty. Initially, we assumed that all students had the same level of learning ability. When identifying students' abilities, we considered exercise difficulty as the confounding factor, and when identifying exercise difficulty, we treated students' abilities as the confounding factor. We employed backdoor adjustment to eliminate the influence of these confounding factors.
- Using the physical loss function to adjust the loss term and optimize the prediction: the logistic growth model can describe the dynamic process of students' knowledge mastery. In order to enable the neural network to learn the rules that conform to the differential equations of the logistic model, the logistic model is used as a physical model to construct a physical loss function for constraints. The academic performance of

students is related to their abilities and the difficulty of their problems. If the predicted student score is high, it indicates that the student's learning ability is high and the selected question is relatively difficult. On the contrary, if the predicted student score is low, it indicates that the student's learning ability is low and the selected question is relatively easy. We use student abilities obtained through backdoor adjustments and interventions, and if we violate the above common sense, the punishment will be increased.



Student Learning Process Data

Student ID	Exercise ID	Answer Status	Start Time	Doing Time
stu_1	e_1	1	$start_1$	at_1
stu_2	e_2	1	$start_2$	at_2
…	…	…	…	…
…	…	…	…	…
stu_n	e_n	0	$start_n$	at_n

Exercises—Knowledge Point Relationships

Exercise ID	Knowledge Point ID
e_1	i
e_2	j
…	…
…	…
e_n	n

Knowledge Points Temporal Cause and Effect Matrix

Knowledge Point Positive and False Temporal Cause and Effect Matrix

	$skill_1$	$skill_2$	…	…	$skill_n$
$skill_1$	1	0.5	…	…	0
$skill_2$	0	1	…	…	0.6
…	…	…	1	…	0.3
…	…	…	…	1	0
$skill_n$	0.2	0	…	…	1

Student—Knowledge Point—Time

Student ID	Knowledge Point ID	Start Time
stu_1	$skill_1$	$start_1$
stu_2	$skill_2$	$start_2$
…	…	…
…	…	…
stu_n	$skill_n$	$start_n$

Figure 1. TLPKT_PINN modeling framework.

3.2. Causal Relationship Mining of Knowledge Points

For the relationship between knowledge points in temporal causal mining, its strength function [28] is modeled by two recurrent neural networks: one is a recurrent neural network (RNN) that captures the relationships between events over time, and the other is an RNN based on a time series for updating intensity functions. On this basis, an attention mechanism is introduced. For each student's interactive information $(z_i, t_i)_i^e$, that is to say, at time t_i, obtain the knowledge point z_i. Separate the situations of correctly and incorrectly obtained knowledge points, recode the knowledge point, and derive the causal relationship between correctness and errors of knowledge points.

The intensity function can be represented as $\lambda(t) = \phi_\theta(t; h_{t_i})$, where h_{t_i} is the feedback loop capturing the influence of previous events, defined as $h_{t_i} = h_v(t_i; h_{t_i})$. The weights ϕ and v represent the network parameters. The influence of historically learned knowledge naturally decays over time, so a temporal component is added to the original RNN to ensure that the final states gradually diminish.

Endogenous variables can be expressed as follows:

$$h_i^e = \phi_e^h(W_e^h(z_i, t_i) + B_e^h h_{i-1}^e \gamma(t_i - t_{i-1}) + b_e^h) \tag{3}$$

Here, ϕ represents the activation function, $\gamma(t)$ is the decay function, (z_i, t_i) denotes the embedding vector of the knowledge point z and the time features, $W_e^h(z_i, t_i)$ indicates the influence from the current event, and $B_e^h h_{i-1}^e \gamma(t_i - t_{i-1})$ represents the influence from historical events.

The temporal RNN event sequence encoder first transforms the event sequence into hidden states. Then, utilizing the attention mechanism of time process, the intensity function of the event type z appears on events that have a significant impact, guided by specific parameter vectors u_z. The attention distribution is represented as shown in Equations (4) and (5):

$$e_{z_i z} = \tanh(h_i^e u_z) \tag{4}$$

$$a_{z_i z} = \frac{\exp(e_{z_i z})}{\sum\limits_i \exp(e_{z_i z})} \tag{5}$$

Define the infectious matrix in a manner similar to a conventional point process to reflect the event dependency between events: $A_{z_i z} =< a_{z_i z} >$. Thus, we obtain the event dependency representation s_z for event type z:

$$s_z(t) = \sum_i a_{z_i z} h_i^e \gamma(t - t_i) \tag{6}$$

$$\lambda_z(t) = f(w_e s_z(t)) \tag{7}$$

To correctly integrate the dense feature vectors sampled at different timestamps, we employed a synchronous RNN, passing the hidden states to subsequent layers to compute the exogenous intensity. Using the synchronous RNN allows us to obtain the hidden state h_t^x of the time series x:

$$h_t^x = \phi_t^x(W_t^x x_t + B_t^x h_{t-1}^x + b_{\mathrm{x}}^{\mathrm{h}}) \tag{8}$$

Finally, the endogenous and exogenous intensities are jointly modeled, with the intensity function based on the collaborative layer under endogenous and exogenous representations. Therefore, the intensity function can be expressed as follows:

$$\lambda_z(t) = f(w_e s_z(t) + w_z^x h_t^x) \tag{9}$$

where the first term $w_e s_z(t)$ represents the endogenous component of event dynamics, and the second term $w_z^x h_t^x$ represents the exogenous intensity.

To integrate the intensity function with the Granger causality test, we first calculate the intensity function for each knowledge point occurrence sequence to quantify the changes in their occurrence rates. Next, by introducing the intensity function as an independent variable within the Granger causality test framework, we evaluate the predictive capability of these intensity functions for other knowledge points. This means that if the appearance of one knowledge point can increase the probability of another knowledge point appearing, it indicates that there is a causal relationship between the two knowledge points, and thus, we can deduce a causal relationship matrix between knowledge points.

3.3. LPKT Model Integrating Temporal Causality

3.3.1. Learning Embedding and Knowledge Point Embedding

Use an embedding matrix $E \in \mathbb{R}^{J \times d_e}$ to represent the exercise set, where J is the number of exercise and d_e is the dimension. Each exercise e_t in the learning unit x_t is represented as a vector $e_t \in \mathbb{R}^{d_e}$. To obtain the learning embedding $l_t \in \mathbb{R}^{d_k}$ of the basic learning unit, the exercise e_t, student response time at_t, and answer status a_t are concatenated together, and a multi-layer perceptron (MLP) is used to deeply fuse the exercise embedding, response time embedding, and answer embedding, as shown in Formula (10):

$$l_t = W_1^T [e_t \oplus at_t \oplus a_t] + b_1 \tag{10}$$

where $\oplus$ represents the concatenation operation, $W_1 \in \mathbb{R}^{(d_e + d_k + d_a) \times d_k}$ is the weight matrix, and $b_1 \in \mathbb{R}^{d_k}$ is the bias term.

The purpose of knowledge embedding is to store and update students' knowledge states during the learning process. In the LPKT model, knowledge embedding is initialized as an embedding matrix $h \in \mathbb{R}^{M \times d_k}$, where M is the number of knowledge concepts, and each row of matrix h represents the mastery level of the corresponding knowledge concept. In every learning interaction among students, the LPKT model updates the learning gains for each knowledge concept into the knowledge embedding, while also incorporating the forgetting effect.

3.3.2. The Learning Module

Learning gain represents the differences in student performance at two time points, that is to say, taking into account the differences in performance during two consecutive learning interactions. The previous learning embedding $\mathrm{l_{t-1}}$ and current learning embedding $\mathrm{l_t}$ of the student are connected as fundamental input elements to model the learning gain. There are two main factors influence a student's learning gain: the interval between learning sessions and the knowledge state of previously related causal knowledge points. On one hand, the interval between two learning units is a critical factor in the learning process; generally, if the interval is shorter, it indicates that the student's learning process is compact and continuous. On the other hand, when a student engages with a particular knowledge point, the mastery state of previously related causal knowledge points will also affect their mastery of that knowledge point. Therefore, the above two factors are modeled to capture the evolution of learning gain.

We concatenate the interval time to the basic input elements between two consecutive learning embedding. For the student's prior knowledge status, in order to focus on the knowledge state related to the knowledge concepts of the current exercise, we first combine the current knowledge concept vector q_{e_t} associated with $h_{\mathrm{t}-1}$ to obtain the knowledge state $\widetilde{h'}_{\mathrm{t}-1}$:

$$\widetilde{h'}_{\mathrm{t}-1} = q_{e_t} \cdot h_{t-1} \tag{11}$$

The level of mastery that a student has over a specific knowledge point is not only related to their current state but also to the states of knowledge points that have a causal relationship with that knowledge point. The mastery level of the knowledge points that have a causal relationship with knowledge point X in the knowledge state $\widetilde{h'}_{t-1}$ of student i is denoted as Mastery(Y). The mastery level for each knowledge point is defined as the sum of the product of the weights of the edges (the causal strength between knowledge points) in the causal matrix for knowledge point (X) and the mastery levels of the neighboring points (the knowledge points that are causally related). The weighted sum is then divided by the total weight to normalize the result. The specific formula is shown as follows:

$$Mastery(X) = \frac{\sum_{Y\in \mathrm{ch(X)}}^{|\mathrm{ch(X)}|} W_X(Y) \times Mastery(Y) + \sum_{Y\in \mathrm{ch(X)}}^{|\mathrm{ch(X)}|} W_Y(X) \times Mastery(Y)}{|A\mathrm{dj}(X)|} \tag{12}$$

Finally, the mastery of each student, denoted as Mastery(X), is integrated to obtain the final knowledge state vector $\widetilde{h}_{\mathrm{t}-1}$.

The modeling of students' learning gains lg_t is as follows:

$$\mathrm{lg}_t = \tanh(W_2^T[l_{t-1} \oplus it_t \oplus l_t \oplus \widetilde{h}_{t-1}] + b_2) \tag{13}$$

where $W_2 \in \mathbb{R}^{(4d_k)\times d_k}$ is the weight matrix and $b_2 \in \mathbb{R}^{d_k}$ is the bias term.

After obtaining the student's knowledge state and learning gains, the learning gate Γ_t^{l} is used to control the student's ability to absorb knowledge, defined as follows:

$$\Gamma_{\mathrm{t}}^{\mathrm{l}} = \sigma(W_3^T[l_{t-1} \oplus it_t \oplus l_t \oplus \widetilde{h}_{t-1}] + b_3) \tag{14}$$

where $W_3 \in \mathbb{R}^{(4d_k)\times d_k}$ is the weight matrix, $b_3 \in \mathbb{R}^{d_k}$ is the bias term, and σ is the nonlinear sigmoid activation function.

Then, multiply $\Gamma_{\mathrm{t}}^{\mathrm{l}}$ by lg_t to obtain the actual learning benefit of the student's t-th learning interaction. Similarly, in order to focus on the learning gain of relevant knowledge concepts in exercise e_t, we multiply LG_t by $\mathrm{q}_{\mathrm{e_t}}$ to obtain the relevant learning gain $\widetilde{LG}_{\mathrm{t}-1}$:

$$\mathrm{LG}_t = \Gamma_t^l(\mathrm{lg}_t + 1)/2 \tag{15}$$

$$\widetilde{LG}_{\mathrm{t}-1}= \mathrm{q}_{e_t} \cdot LG_t \tag{16}$$

where the learning gain of students LG_t will always be positive, with a range mapped from $(-1, 1)$ to $(0, 1)$.

3.3.3. Forgetting Module

Learning plays an enhancing role in students' knowledge status, but over time, according to the forgetting curve theory, the number of knowledge points remembered by students will exponentially decrease. In order to model the complex forgetting effect, we used the forgetting gate Γ_t^{f} to simulate the forgetting process of students. The gate is modeled based on three factors: (1) the student's previous knowledge state $h_{\mathrm{t}-1}$, (2) learning gain $LG_{\mathrm{t}-1}$, and (3) interval time it_{t}. Therefore, the definition of the forgetting gate is as follows:

$$\Gamma_{\mathrm{t}}^{\mathrm{f}} = \sigma(W_4^T[h_{t-1} \oplus LG_t \oplus it_t] + b_4) \tag{17}$$

where $W_4 \in \mathbb{R}^{(3d_k)\times d_k}$ is the weight matrix, $b_4 \in \mathbb{R}^{d_k}$ is the bias term, and σ is the nonlinear sigmoid activation function.

Then, the update of students' knowledge status after completing the t-th learning interaction is shown in Formula (18):

$$h_t = \widetilde{LG}_{\mathrm{t}} + \Gamma_t^f \cdot \mathrm{h}_{t-1} \tag{18}$$

3.3.4. Prediction Module and Objective Function

When given an exercise, students will attempt to apply the knowledge they have already mastered to the corresponding knowledge concepts to solve it. Therefore, the student's relevant knowledge state should be used to infer their performance in the next exercise. The prediction formula is as follows:

$$y_{t+1} = \sigma(W_5^T[e_{t-1} \oplus \widetilde{h}_t] + b_5) \tag{19}$$

where $W_5 \in \mathbb{R}^{(2d_k) \times d_k}$ is the weight matrix, and $b_5 \in \mathbb{R}^{d_k}$ is the bias term.

The cross-entropy loss between the predicted value y and the actual answer a is chosen as the objective function:

$$\mathbb{L}(\theta) = -\sum\nolimits_{t=1}^{T} (a_t \log y_t + (1 - a_t)\log(1 - y_t)) + \lambda_\theta ||\theta||^2 \tag{20}$$

where θ represents all parameters and λ_θ is the regularization hyperparameter. The Adam optimizer is used to minimize the objective function in small batches.

3.4. Loss Function Optimization

3.4.1. Student's Ability and Exercise Difficulty Based on Backdoor Adjustment

Students with different learning abilities have distinct preferences in answering questions. For example, students with stronger learning abilities tend to tackle more challenging exercises, but this may result in a relatively lower accuracy rate in their answers. Conversely, students with weaker learning abilities tend to choose easier problems, leading to a higher accuracy rate. Therefore, we should not overestimate or underestimate a student's learning ability. Based on relevant knowledge of causal inference, it can be argued that 'the accuracy of a student's answers' is a major 'cause' for identifying learning ability, while 'exercise difficulty' acts as a 'confounding factor' in the causal relationship between the two. We can derive the structural causal models shown in Figure 2.

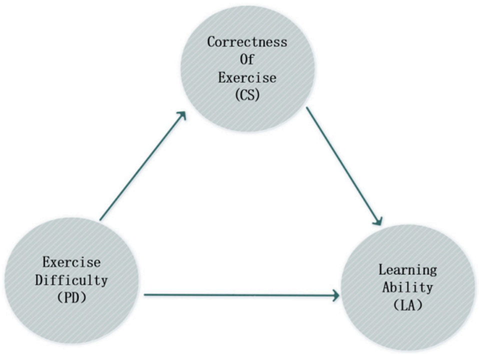


Figure 2. Structural causal model with the confounder 'difficulty of the exercise'.

In order to accurately identify students' learning abilities, we use 'backdoor adjustment' to conduct interventions. We defined the learning ability and exercise difficulty, assuming that each level has the same prior probability; that is,

$$P(PD = d) = \frac{1}{D}, P(LA = l) = \frac{1}{L} \tag{21}$$

where D and L are the number of divisions for learning ability and exercise difficulty levels, respectively. The post-intervention learning abilities is obtained as shown in Formula (22):

$$P(LA|do(CS)) = \sum_{d=1}^{D} P(LA|CS, PD = d)P(PD = d) \tag{22}$$

3.4.2. The Logistic Model

The logistic model is a type of model based on the logistic function. In the field of knowledge tracing, its basic concept is that the probability of correctly answering exercises can be represented by mathematical functions of students and KC parameters. Using student grades as the dependent variable, we can learn general parameters from historical data to model students and make predictions for answering questions. In the logistic model, students' binary answers (correct/incorrect) follow a Bernoulli distribution. At the beginning of the 21st century, researchers proposed using logistic models to handle knowledge tracking tasks, with student grades as the dependent variable, and learning general parameters from historical data to model students for answer prediction.

We use the Sigmoid function to describe the probability of a student mastering a certain knowledge point at a certain moment. This is a nonlinear function, typically manifested as:

$$P(y = 1|x) = \frac{1}{1 + \exp(-f_\theta(x))} \tag{23}$$

We consider its dynamic changes over time and convert it into a differential equation. Assuming that students' mastery of a certain skill changes over time, a dynamic model similar to "learning progress" can be used to represent the changes in students' mastery of the skill:

$$\frac{dP(t)}{dt} = rP(t)(1 - P(t)) \quad where \quad P(t) = \frac{1}{1 + \exp(-f_\theta(x(t)))} \tag{24}$$

where $P(t)$ is the probability of a student mastering a certain skill at time t. R is the learning rate, indicating the speed at which students acquire skills. $f_\theta(x(t))$ is the output of the model, representing the student's mastery of the knowledge points at time t. This differential equation is a standard logistic growth model that can describe the dynamic process of students' knowledge acquisition.

3.4.3. Optimize the Loss Function

In order for the neural network to learn the rules that conform to the differential equations of the logistic model, we can impose constraints by introducing loss terms in the differential equations. In this case, the loss function not only considers the prediction results of the network, but also needs to consider whether the network output $P(t)$ satisfies the differential equation of the logistic growth model.

We need to ensure that the derivative of the network output $P(t)$ with respect to time t conforms to the Logistic equation. We calculate the derivative of the neural network output $P(t)$ with respect to time t through automatic differentiation. Assuming $P(t)$ is the probability of a student mastering a certain skill at that time output by the neural network, we construct the loss function as follows:

$$L_{pinn} = \int_{t_0}^{T} \left(\frac{dP(t)}{dt} - rP(t)(1 - P(t)) \right)^2 dt \tag{25}$$

where $\frac{dP(t)}{dt}$ is the derivative of the neural network output with respect to time t. This loss term ensures that the probability $P(t)$ of the neural network output follows the dynamic changes in the logistic growth model.

According to the laws of reality, if the predicted student academic performance is high, the corresponding learning ability of the student should be relatively high, and if the predicted student academic performance is low, the corresponding learning ability of the student should be relatively low. Therefore, based on this phenomenon, we construct a loss term between academic performance and learning ability:

$$L_{\text{ability}} = \sum\nolimits_{\text{i}} \left(\widehat{y}_i - f(C_i)\right)^2 \tag{26}$$

where $f(C_i)$ is the student ability obtained through backdoor adjustment, and $\hat{y}_i$ is the score that maps the predicted student academic performance to the interval (0, 1).

The final loss function is expressed as shown in Formula (26):

$$L = L_{cls} + \lambda L_{pinn} + L_{ability} \tag{27}$$

where λ is a hyperparameter that balances physical constraints with traditional losses. By optimizing this comprehensive loss function, the neural network can not only make predictions based on students' historical data but also follow the differential equations of the logistic model, thus better simulating the dynamic process of students' knowledge mastery.

4. Experiment

4.1. Datasets

We use two diverse real-world datasets to evaluate the effectiveness of the model in different learning scenarios. Table 1 shows the statistical information of all datasets. We introduce and compare each datasets as follows:

- ASSIST2012: This dataset was collected from the educational platform ASSISTMENTS, which provides high school math problems. This dataset contains data from the 2012–2013 academic year, and students need to do similar exercises to master these problem sets. We filtered out records without knowledge concepts and students who completed fewer than 20 questions.
- ASSISTchall: This dataset was collected from ASSISTMENTS in 2017 and was used in a data mining competition. The data were gathered from a longitudinal study that tracked middle school students' use of the ASSISTMENTS blended learning platform from 2004 to 2007. In this dataset, the learning sequence of students is much longer than ASSIST2012.

Table 1. Information on the dataset.

	ASSIST2012	**ASSISTchall**
Number of students	28,914	1600
Number of knowledge concepts	265	102
Number of problems	532,090	3142

4.2. Results and Discussion

We compared TLPKT_PINN with several previous methods. For fair comparison, all these methods have been tuned to have the best performance.

- BKT: Using the hidden Markov model and Bayesian inference methods, it is used to evaluate and predict the dynamic changes in students' mastery of knowledge points during the learning process.
- DKT: The model is based on a Recurrent Neural Network (RNN) and is used to dynamically track students' mastery of knowledge points. It analyzes students' interactive data, learns their knowledge status, and predicts their performance in future tasks.

- DKVMN: This model defines a static matrix to store potential knowledge concepts and a dynamic matrix to update the corresponding knowledge states over time through read and write operations, using a memory network to obtain the interpretable student knowledge states.
- AKT: This uses two self-attention encoders to learn context-aware representations of exercises and answers; the knowledge evolution model is called the knowledge retriever, which utilizes attention mechanisms to retrieve knowledge obtained in the past that is relevant to the current exercise.
- LPKT: Modeling the learning gain during the learning process by capturing the difference between two consecutive learning units. The diversity of learning benefits is measured by students' relevant knowledge status and interval time. The learning gate is used to distinguish students' ability to absorb knowledge, and the forgetting gate is used to determine the decrease in students' knowledge over time.

This article conducted experimental comparisons with these classic knowledge tracking models and their latest variants, using Area Under Curve (AUC) and Root Mean Square Error (RMSE) under the ROC curve to analyze the performance of the comparison models. For all datasets, we performed standard 5-fold cross validation on all models. Therefore, for each group, 80% of the students were divided into a training set and the remaining 20% were used as a testing set. We randomly initialized all parameters in a uniform distribution. We learned all hyperparameters on the training set and evaluated the test set using the model that performed best on the validation set. We added a dropout layer with a dropout rate of 0.2 to prevent overfitting. In our implementation, dk and de parameters are set to 128, and da is set to 50. The small positive value γ to enhance the Q-matrix was set to 0.03. The experimental results are shown in Table 2.

Table 2. Experimental results.

	ASSIST2012		**ASSISTChall**	
	AUC	**RMSE**	**AUC**	**RMSE**
BKT	0.622	0.511	0.638	0.513
DKT	0.701	0.432	0.721	0.447
DKVWM	0.685	0.437	0.710	0.450
AKT	0.769	0.414	0.766	0.431
LPKT	0.778	0.407	0.772	0.415
TLPKT_PINN	0.828	0.375	0.798	0.382

It can be seen that as an improvement to the LPKT model, the TLPKT-PINN method in this paper performs better in both datasets compared to deep learning methods such as DKT, DKVMN, and DIKT-CI, which showed an increase of at least 5% in AUC values across three datasets. In addition, as an improvement of the LPKT model, TLPKT-PINN has increased the AUC values by at least 2.6% in both datasets. This improvement is due to its integration of temporal causal relationships between knowledge points and consideration of student performance prediction.

4.3. Ablation Experiment

In order to investigate the impact of each module in the TLPKT_PINN model on the final performance prediction, we designed several ablation experiments to validate our model. The specific setup information is as follows:

- LPKT-PINN: Remove the temporal causality module from the model;

- TLPKT_PINN_ability: Remove the module that adjusts the learning ability through backdoor adjustment in the model, and remove the learning ability term from the loss function term;
- TLPKT: Remove the physical loss module from the model.
- TLPKT_1: Remove additional loss terms from the model.

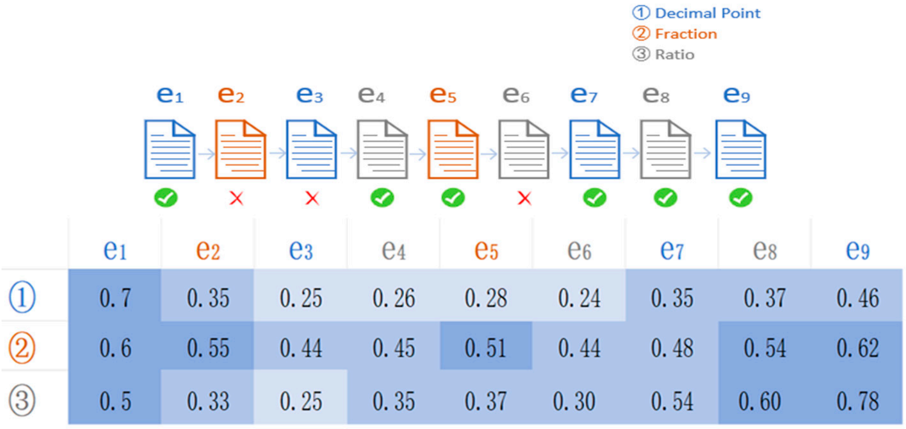
With the results given in Table 3, it is clear that the temporal causality module, the physical loss module, and the physical loss module adjusted by learning ability have an effect on the experimental results. By comparing the performance results of LPKT-PINN, TLPKT_PINN_ability, TLPKT, and TLPKT_1, it can be seen that both the temporal causality module and the loss module have improved the model performance, indicating the effectiveness of each module in the TLPKT-PINN model. In addition, it can be seen from the results of LPKT-PINN that a better prediction performance can be achieved by considering the temporal causality between knowledge points. It is worth noting that in the causal module of knowledge points, the optimization effect of ASSIST2012 dataset is more significant, which may be due to the relatively small number of knowledge points in this dataset, and the causal network of learned knowledge points is not too complex. Finally, it was verified through TLPKT and TLPKT_1 that optimizing the loss term can improve the prediction performance.

Table 3. Results of ablation experiments.

	ASSIST2012		**ASSISTChall**	
	AUC	**RMSE**	**AUC**	**RMSE**
LPKT_PINN	0.786	0.410	0.788	0.405
TLPKT_PINN_ability	0.803	0.384	0.789	0.392
TLPKT	0.801	0.392	0.784	0.398
TLPKT_1	0.792	0.405	0.781	0.399

4.4. Updating the Mastery Level of Students' Knowledge Points

Figure 3 shows a student's learning sequence. The student performed the following exercises in temporal order. The mastery level of students' knowledge points was obtained through the combination of neural networks and temporal causality:



	e1	e2	e3	e4	e5	e6	e7	e8	e9
①	0. 7	0. 35	0. 25	0. 26	0. 28	0. 24	0. 35	0. 37	0. 46
②	0. 6	0. 55	0. 44	0. 45	0. 51	0. 44	0. 48	0. 54	0. 62
③	0. 5	0. 33	0. 25	0. 35	0. 37	0. 30	0. 54	0. 60	0. 78

Figure 3. Student answer sequence and the mastery level of the student's knowledge points.Among them, e_1-e_9 are exercises, and the colors represent the knowledge points contained in the exercises. The specific knowledge points and corresponding colors are located in the upper right corner. The check or cross below the exercise represents whether the student did the problem correctly or incorrectly. For better identification, we will fill in different blue colors for students with mastery levels greater than or equal to 0.5, 0.3~0.5, and less than 0.3.

It is known that through the study of temporal causality, these three knowledge points have a causal relationship. We can see that when the degree of mastery of a knowledge point changes, the degree of mastery of its causal knowledge points also changes accordingly.

4.5. Performance Analysis of PINN Model

A key feature of the logistic model is its ability to simulate the "asymptotic" and saturation effects in the learning process. Students progress quickly in the early stages of learning, but as their mastery approaches saturation, their progress gradually slows down. As shown in Figure 4, during the learning process, students' knowledge reserves in the early stage are not high, which basically conforms to the trend of the logistic model. However, the deviation of the results predicted by the neural network is slightly larger. But in the latter part, after training the neural network, the neural network prediction performance is better. Therefore, we decided to use the logistic function as a physical model for the first 15% step time, and controlling the learning progress output of the network through the logistic function to simulate the process of students from not understanding a certain knowledge point to fully mastering it, in order to impose constraints. As shown in Figure 5, after adding the PINN model, its mean square error is slightly smaller, indicating that adding the PINN model can improve the accuracy of predicting students' grades.

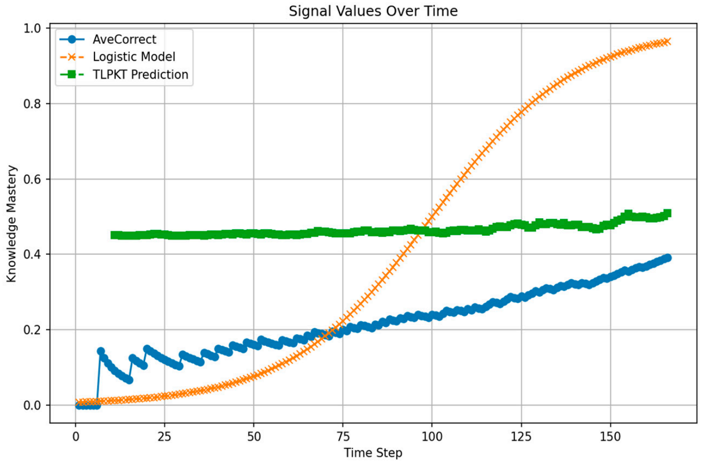


Figure 4. The time variation values of the three models (student's actual answering situation, neural network prediction, and the logistic model).

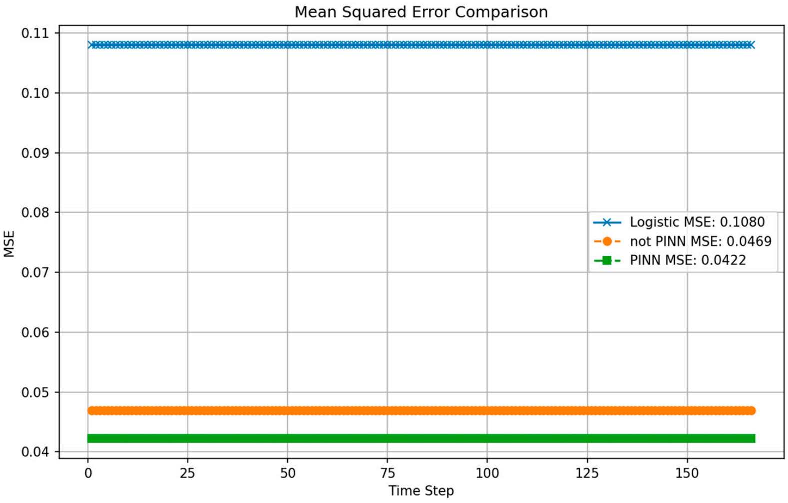


Figure 5. The mean square error of the three models (neural network prediction, the logistic model, and the neural network incorporating the PINN model).

5. Conclusions

In this paper, we propose a deep learning knowledge tracking model that integrates temporal causality and the Physics-Informed Neural Network (PINN) model. Since the probability of a student's mastery of a knowledge point is affected by the mastery probability of its causal knowledge points, the model mines the causal relationship between knowledge points through the temporal causality model and combines the temporal causality matrix with the student's learning, which can lead to the student's final mastery of the knowledge points affected by the related knowledge points. We found through experiments that in the first half of students' learning process, the logistic model is more in line with their actual answer situation, and after training, the neural network is more in line. So, we treated the logistic model as a physical model within the first 15% of the time step and constrained the prediction of the neural network, finding that it can improve the accuracy of student performance prediction. Adding another loss term would violate the principle that 'if a student's performance is high, they have high learning ability; if their learning ability is low, they do not have good learning ability', increase the punishment, and thus improve the ability to predict student performance. It can be concluded from the experiments that the TLPKT_PINN model has a better prediction effect in the dataset and shows the changes in the students' learning process in a more rationalized way.

In future work, we can further explore the possibility of integrating the temporal causal matrix more rationally with the student learning process, thus reducing the time complexity, and in the knowledge tracking model, we can investigate how to automatically learn the specific weights in the q-matrix in order to more accurately represent the relationship between the exercises and the knowledge concepts.

Author Contributions: Methodology, F.L., Y.L. and Y.B.; Writing—original draft, Y.L.; Writing—review & editing, F.L. and Y.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Science and Technology Major Project (2022ZD0119501), the NSFC (52374221), Sci. & Tech. Development Fund of Shandong Province of China (ZR2022MF288, ZR2023MF097), the Taishan Scholar Program of Shandong Province (ts20190936), and the Science and Technology Program Special Project of Qingdao West Coast New Area (202209).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The ASSISTment2012 and ASSISTchall datasets are from the ASSISTments platform.These datasets have been organized and can be found by following the following link: https://base.ustc.edu.cn/data/ASSISTment/ accessed on 20 November 2024.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI-EdTech	AI-powered educational technologies
AKT	Context-aware Attentive Knowledge Tracing
BKT	Bayesian Knowledge Tracing
DBKT	Dynamic Bayesian Knowledge Tracing
DBNs	Dynamic Bayesian Networks
DKT	Deep Knowledge Tracing
DKVMN	Dynamic Key–Value Memory Network
GKT	Graph-based Knowledge Tracing

HMM	Hidden Markov Model
LPKT	Learning Process-consistent Knowledge Tracing
LSTM	Long short-term memory
MDPI	Deep Knowledge Tracing Integrating Temporal Causal Inference and PINN
RNNs	Recurrent Neural Networks
TLS-BKT	Three-Learning-State BKT

References

1. Darvishi, A.; Khosravi, H.; Sadiq, S.; Gašević, D.; Siemens, G. Impact of AI assistance on student agency. *Comput. Educ.* **2024**, *210*, 104967. [CrossRef]
2. Chen, H.; Yin, C.; Li, R.; Rong, W.; Xiong, Z.; David, B. Enhanced learning resource recommendation based on online learning style model. *Tsinghua Sci. Technol.* **2019**, *25*, 348–356. [CrossRef]
3. Shen, S.; Liu, Q.; Huang, Z.; Zheng, Y.; Yin, M.; Wang, M.; Chen, E. A survey of knowledge tracing. *arXiv* **2021**, arXiv:2105.15106.
4. Corbett, A.T.; Anderson, J.R. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model. User-Adapt. Interact.* **1995**, *4*, 253–278. [CrossRef]
5. Eddy, S.R. Hidden markov models. *Curr. Opin. Struct. Biol.* **1996**, *6*, 361–365. [CrossRef]
6. Piech, C.; Bassen, J.; Huang, J.; Ganguli, S.; Sahami, M.; Guibas, L.J.; Sohl-Dickstein, J. Deep Knowledge Tracing. *Adv. Neural Inform. Process. Syst.* **2015**, *28*.
7. Jordan, M.I. Serial order: A parallel distributed processing approach. In *Advances in Psychology*; North-Holland: Amsterdam, The Netherlands, 1997; Volume 121, pp. 471–495.
8. Shen, S.; Liu, Q.; Chen, E.; Huang, Z.; Huang, W.; Yin, Y.; Su, Y.; Wang, S. Learning process-consistent knowledge tracing. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Virtual, 14–18 August 2021; pp. 1452–1460.
9. Zanellati, A.; Di Mitri, D.; Gabbrielli, M.; Levrini, O. Hybrid models for knowledge tracing: A systematic literature review. *IEEE Trans. Learn. Technol.* **2024**, *17*, 1021–1036. [CrossRef]
10. Kaser, T.; Klingler, S.; Schwing, A.G.; Gross, M. Dynamic Bayesian networks for student modeling. *IEEE Trans. Learn. Technol.* **2017**, *10*, 450–462. [CrossRef]
11. De Baker, R.S.J.; Corbett, A.T.; Aleven, V. More accurate student modeling through contextual estimation of slip and guess probabilities in Bayesian knowledge tracing. In Proceedings of the 9th International Conference on Intelligent Tutoring Systems (LNCS 5091), Montreal, QC, Canada, 23–27 June 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 406–415.
12. Zeng, G.; Zhuang, J.; Huang, H.; Tian, M.; Gao, Y.; Liu, Y.; Yu, X. Use of Deep Learning for Continuous Prediction of Mortality for All Admissions in Intensive Care Units. *Tsinghua Sci. Technol.* **2023**, *28*, 639–648. [CrossRef]
13. Yang, X.; Esquivel, J.A. Time-aware LSTM neural networks for dynamic personalized recommendation on business intelligence. *Tsinghua Sci. Technol.* **2023**, *29*, 185–196. [CrossRef]
14. Zhang, J.; Shi, X.; King, I.; Yeung, D.Y. Dynamic key-value memory networks for knowledge tracing. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 765–774.
15. Nakagawa, H.; Iwasawa, Y.; Matsuo, Y. Graph-based knowledge tracing: Modeling student proficiency using graph neural network. In Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, Thessaloniki, Greece, 14–17 October 2019; pp. 156–163.
16. Ghosh, A.; Heffernan, N.; Lan, A.S. Context-aware attentive knowledge tracing. In Proceedings of the 26th ACM SIGKDD International Conference On Knowledge Discovery & DATA Mining, Virtual, 6–10 July 2020; pp. 2330–2339.
17. Eljialy, A.E.M.; Uddin, M.Y.; Ahmad, S. Novel framework for an intrusion detection system using multiple feature selection methods based on deep learning. *Tsinghua Sci. Technol.* **2024**, *29*, 948–958. [CrossRef]
18. Jiang, Z.; Ning, Z.; Miao, H.; Wang, L. STDNet: A Spatio-Temporal Decomposition Neural Network for Multivariate Time Series Forecasting. *Tsinghua Sci. Technol.* **2024**, *29*, 1232–1247. [CrossRef]
19. Daley, D.J.; Vere-Jones, D. *An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods*; Springer: New York, NY, USA, 2003.
20. Granger, C.W.J. Investigating causal relations by econometric models and cross-spectral methods. *Econom. J. Econom. Soc.* **1969**, *37*, 424–438. [CrossRef]
21. Didelez, V. Graphical models for marked point processes based on local independence. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2008**, *70*, 245–264. [CrossRef]
22. Eichler, M.; Dahlhaus, R.; Dueck, J. Graphical modeling for multivariate Hawkes processes with nonparametric link functions. *J. Time Ser. Anal.* **2017**, *38*, 225–242. [CrossRef]

23. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Inferring solutions of differential equations using noisy multi-fidelity data. *J. Comput. Phys.* **2017**, *335*, 736–746. [CrossRef]
24. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Machine learning of linear differential equations using Gaussian processes. *J. Comput. Phys.* **2017**, *348*, 683–693. [CrossRef]
25. Owhadi, H. Bayesian numerical homogenization. *Multiscale Model. Simul.* **2015**, *13*, 812–828. [CrossRef]
26. Williams, C.K.I.; Rasmussen, C.E. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, USA, 2006.
27. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [CrossRef]
28. Xiao, S.; Yan, J.; Farajtabar, M.; Song, L.; Yang, X.; Zha, H. Learning time series associated event sequences with recurrent point process networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3124–3136. [CrossRef] [PubMed]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.