A Genetic Causal Explainer for Deep Knowledge Tracing

Qing Li, Xin Yuan, Sannyuya Liu, *Member, IEEE*, Lu Gao, Tianyu Wei, Xiaoxuan Shen, and Jianwen Sun

***Abstract*—Knowledge tracing (KT) has become an increasingly relevant problem in intelligent education services. Deep learning-based KT (DLKT) achieves superb performance in terms of prediction accuracy, but it lacks of explainability, which makes us hard to trust or understand models. The previous work on explaining DLKT was mainly based on gradients or attention scores, which is susceptible to spurious correlations, reducing the credibility of the explanation. To address this limitation, in this article, we propose a causal explanation method based on the genetic algorithm (GA), named genetic causal explainer (GCE), which constructs a causal framework to estimate the attribution of subsequence to the predictions of DLKT models, and a genetic coding system is designed. Further, A multistrategy initialization method inspired by domain prior knowledge is proposed, and a global empirical matrix is introduced to capture the causal correlation knowledge during the search process across instances, and guiding the mutation operators. The GCE as a post hoc explanation method can generate explanation results without affecting model training, and can be applied to analyze different DLKT models. Experimental results demonstrate the GCE perform better than other explanation methods in terms of accuracy and readability in quantitative assessments. Meanwhile, the GCE also shows good application prospects in mining educational laws and comparing KT models.**

***Index Terms*—Cause-effect, explanation method, genetic algorithm (GA), knowledge tracing (KT).**

I. Introduction

KNOWLEDGE tracing (KT), [1], [2], [3] is the core module of the intelligent tutoring system (ITS) [4] and is mainly used to build a model of learners' cognitive development based on their historical sequences. This model achieves a precise prediction of learners' learning outcomes, which forms the basis for decision-making in adaptive learning technologies, such as learning strategy adaptation and learning resource recommendation [5], [6]. With the introduction of deep learning technology into KT in recent years, deep KT models have gradually replaced traditional models due to their superior prediction performance. However, due to that the black-box nature of deep neural networks makes them less trustworthy and limits their wider acceptance [7], many scholars have started to study the explainability of neural networks and try to open their black-box in various ways. Similarly, the deep learning-based KT (DLKT) model is also affected by the fact that the decision process of the neural network is always a black box and users cannot know the decision process [8]. To achieve high-quality student learning process modeling, as well as improving the performance of model prediction, we should increase the explainability of DLKT models. Clarifying why the DLKT model makes certain predictions will contribute to the analysis the student's cognitive modeling process and provide better-learning strategy support and learning resource recommendations for the intelligent learning tutoring system.

The research on the explainability of deep neural networks is divided into post hoc explanation and ante hoc explanation [9], [10]. The former is a model-agnostic approach [11], [12], while the latter is the study of neural networks with explainability as the learning goal. Most of the previous work on the explainability of DLKT models has been based on self-explanatory models, i.e., the models are designed to be explainable, but the performance of the models is sacrificed to a certain extent. One of the main advantages of post hoc explanation methods is that there is no need to compromise the explainability of the prediction performance [11] since prediction and explanation are two independent processes that do not interfere with each other. In contrast, there has not been much research related to post hoc explainability for DLKT models. The available studies mainly calculated the gradients in backpropagation in the model to determine the importance of features and thus provide the basis for decision-making in DLKT models [13], but all the methods are based on correlations, and the final explainable results obtained by these methods may be affected by spurious correlations between data.

A. Research Motivations

Based on the shortcomings of existing deep KT explanation methods that sacrifice model performance, spurious correlations, and low generality, we propose a post hoc explanation method for explaining DKLT models, which mainly explores

Manuscript received 24 February 2023; revised 5 May 2023; accepted 11 June 2023. Date of publication 15 June 2023; date of current version 1 August 2024. This work was supported in part by the National Key Research and Development Project of China under Grant 2021ZD0110700; in part by the National Natural Science Foundation of China under Grant 62293554, Grant 62107017, and Grant 62077021; in part by the Hubei Provincial Natural Science Foundation of China under Grant 2022CFB414; in part by the China Postdoctoral Science Foundation under Grant 2020M682454; and in part by the Fundamental Research Funds for the Central Universities under Grant CCNU22LJ005. *(Corresponding authors: Xiaoxuan Shen; Jianwen Sun.)*

The authors are with the National Engineering Research Center of Educational Big Data and the Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan 430079, China (e-mail: viven_a@ccnu.edu.cn; yuanxin@mails.ccnu.edu.cn; liusy5918@outlook.com; gaolu@mails.ccnu.edu.cn; weitianyu@mails.ccnu.edu.cn; shenxiaoxuan@ccnu.edu.cn; sunjw@ccnu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TEVC.2023.3286666.

Digital Object Identifier 10.1109/TEVC.2023.3286666

the causal relationships between model inputs and outputs. A causal attribution measurement is proposed for the importance of input units which can avoid the influence of spurious correlations and achieve the explanation of various DLKT models. This explanation method is a general, the model-agnostic explanation method that can be applied to the explanation of arbitrary DLKT models. At the same time, the essence of our explanation method is finding the optimal input subsequence, which is an NP-hard discrete search problem; however, most of the search algorithms used to solve such problems are heuristics based on random or greedy strategies, which have difficulty achieving global optimality. Other methods, such as machine learning methods, require long training cycles and are costly due to poor transferability and flexibility, as they typically require pretraining of search models, which is not very advantageous for fields that require frequent data updates, as once data changes or model updates are needed, retraining is necessary. Therefore, we introduce the genetic algorithm (GA) as an out of the box method to solve the optimal subsequence search problem. To the best of our knowledge, this article is the first to propose an explanation method for deep KT models based on GA to achieve flexible, efficient, accurate, and readable post hoc explainability of deep learning.

B. Research Contributions

1) Construct a causal attribution measurement framework and design a genetic coding scheme for the KT domain that can efficiently, flexibly, faithfully and concisely explain the decision basis for model prediction.
2) Develop a genetic causal explainer (GCE) and propose a prior-based multistrategy population initialization method, which can greatly improve the generalizability of initialized populations.
3) Introduce a global empirical matrix (EM) that can capture the causal correlation values between DLKT model inputs within and across instances and use them to guide the mutation of individual gene.
4) Conduct extensive experiments on two representative DLKT models to demonstrate the effectiveness of our explainer and the results generated by our explainer can uncover potential educational laws.

II. Background and Related Work

A. Knowledge Tracing

The task of KT can be described as the input sequence of student's observations $S_t = \{x_1, x_2, \ldots, x_t\}$ that is trained by model f to predict the student's performance at the next moment x_{t+1}, usually, $x_t = \{q_t, a_t\}$, an interaction pair of answers consisting of questions and response answers, where q_t represents the question answered by students, a_t represents whether the student answered the question correctly, $a_t = 1$ represents correct answers, and $a_t = 0$ represents incorrect answers. Since the student's answer sequence is a time sequence, the student's knowledge mastery is a dynamic change process, where the input sequence $\{x_1, x_2, \ldots, x_t\}$ corresponds to the encoding of the student's answer information at the moment $\{t_1, t_2, \ldots, t_t\}$, and the output sequence $\{y_1, y_2, \ldots, y_t\}$ of the model obtained after the training of the model corresponds to the probability of the student answering all the questions in the question bank correctly at each moment, i.e., $y_t = f_\theta(S_{t-1})$, f_θ is the DLKT model, and θ is the model parameter. f_θ can be trained with numerous observed learning sequences from different students. A brief introduction to KT is shown in Fig. 1.

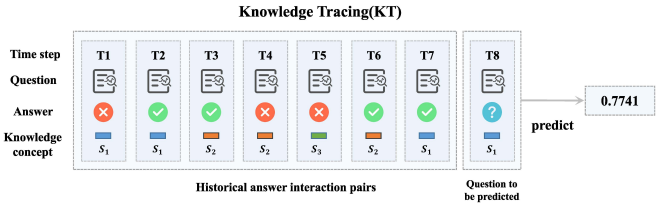

Fig. 1. Brief introduction of KT.

Traditional KT models, represented by BKT [14] and factor analysis models [15], [16], [17], [18], exhibit explainability due to the presence explainable parameters in the model. However, these models have problems, such as difficulty in capturing complex nonlinear relationships among knowledge concepts and do not consider the forgetting factor in the learning process, resulting in low-prediction performance of the model. With the development of deep learning, neural networks have been applied to the field of KT with their powerful ability to fit complex nonlinear relationships, and the proposed models are called DLKT models. The earliest DLKT model, DKT [19], was proposed by scholars in 2015 to model the learning process of students with recurrent neural networks [20], and this model achieved a qualitative leap in prediction accuracy compared with traditional models.

B. Explainable Deep Neural Network

With the booming development of artificial intelligence, its core technology, deep learning, has replaced many traditional methods due to its excellent performance. Deep learning has been used in all major fields for analysis and prediction to solve practical problems. However, while deep neural networks have powerful fitting ability, they also have the problem of poor explainability [8], i.e., there no way to know the decision mechanism of the model and why the model makes a decision, making it difficult to establish trust between humans and machines. Therefore, to help people trust the decision results of neural networks, we need to improve the explainability of deep neural network models with explanation methods.

In recent years, some scholars have started to study the explainability of deep neural networks to try to open the black box of them. Current mainstream explainability methods are divided into two types: ante hoc explainability and post hoc explainability. Ante hoc explainability refers to the self-explanatory model in which a simple model with good explainability is trained or explainability is incorporated in a specific model structure so that the model itself has certain explainability. However, often in the process of designing the model, the performance of the model will be sacrificed to improve the explainability. In this article, we focus on the post hoc explainability of neural networks.

With post hoc explainability of deep neural networks, there is a large class of explanation methods used to calculate the importance of each unit in the input, i.e., input unit importance attribution [11], among which there are some gradient-based methods that analyze the importance of input units according to the magnitude of the gradient. For example, Springenberg et al. [21] proposed the GBP algorithm, the main idea of which is to calculate the gradient by specific backpropagation. The gradient can then reflect the feature importance. Additionally, Bach et al. [22] proposed the LRP algorithm, which focuses on the backpropagation of gradients in recurrent neural networks and then performs importance analysis on each pixel in the input image. In addition to gradient-based methods, other methods calculate the importance of each input unit by special processing of the input features or samples. For example, Li et al. [23] used the Shapley value to fully consider the interaction between different pixel blocks in the same image to provide an accurate explanation of each image instance and class and Lundberg and Lee [24] proposed the SHAP value as a uniform measure of feature importance. SHAP can assign a specific predicted importance value to each feature to obtain the predictive contribution of the input features. Ribeiro et al. [25] proposed the LIME method, which uses a linear model to fit the local behavior of a neural network so that the parameters of the linear model reflect the importance of the input image superpixels. Although these methods have achieved good results in different fields, the attribution importance measurements of these methods are all based on correlations and, thus, may be more affected by spurious correlations, which makes it difficult to guarantee the fidelity of the explanation results.

As a result, some scholars have proposed causal-based explainable methods. For example, Bau et al. [26] proposed a causal framework to understand how deep convolutional GANs (DCGANs) generate images and why they generate them, however, such methods are limited to explaining generative adversarial networks. Chattopadhyay et al. [27] proposed structural causal models as an explainable method, but this method is designed for specific types of models and lacks transferability. Schwab and Karlen [28] proposed CXplain, a causal-based explainable model, but it requires pretraining a corresponding explanatory model for any prediction model. This is not friendly to some fields that require frequent incremental updates of prediction models and may incur high-computational costs. Recently, Wang et al. [29] proposed a reinforcement learning-based causal explainer for graph neural networks (GNNs). This method treats causal effects as the reward of the reinforcement learning algorithm and trains an agent that can generate an explanatory subgraph for a graph to be explained. However, this method requires a significant amount of time to train an excellent decision-making agent, and it also suffers from the aforementioned frequent incremental update problem.

C. Explainability for Knowledge Tracing

Precisely due to the black-box property of deep learning, there is an urgent need for an explainable analysis of DLKT models. DLKT models with high-prediction accuracy have poor explainability, so many scholars have carried out a series of studies to address the explainability problem. Most of these explanation methods study neural networks with explainability as the learning goal, such as Deep-IRT [30] models, which use the output of the deep model as the input of the IRT model [15] and then use the IRT model to make predictions. In addition, some researchers replace question embedding with skill embedding and set some parameters corresponding to question embedding to zero in the answer prediction module of the trained deep model, thus considering the new output probability as the mastery state value of the skill, such as the DKVMN [31] model. In addition, explanations based on attention mechanisms include the EKT [32], SAKT [33], RKT [34], and AKT [35] models. These methods include the calculation of attention weights of historical answer records concerning the current question, and the prediction of the current question is explained by the answers to the historical questions with higher weights based on their semantic correlation with the current question. However, some of these methods improve their explainability at the expense of model accuracy, and all of them are self-explanatory models with strong coupling, which cannot achieve the explanation of other models.

Therefore, scholars have begun focusing on post hoc explanations of KT models. Hu and Rangwala [36] devised a method to give explanations of model predictions of student grades based on a course-specific model assuming that the prerequisite courses the student completed had a significant impact on the current course when predicting student grades. Lu et al. [13] applied LRP for post hoc explainable analysis of KT models to calculate the correlation between model outputs and inputs to explain their prediction results; however, this approach can only achieve the explanation of such models based on recurrent neural networks and still cannot be applied to the explanation of different KT models. Therefore, Deliang Wang et al. [37] proposed using DeepSHAP to explain DLKT models by reacting to the importance of input features through the gradient in backpropagation, and although this method can be applied to arbitrary DLKT models, this gradient-based method has the problem of spurious correlation.

In summary, in order to solve the problem of spurious associations in explainable KT and improve the flexibility, decoupling, and transferability of explainable methods, we have introduced GA as the core technology and combined it with a causal framework to design a deep KT explainer called GCE. The subsequent section will provide a detailed explanation of the explainer.

III. Proposed Algorithm

Based on the post hoc explanation method paradigm, the proposed GCE for deep KT can be applied as an additional pendant in a variety of deep KT models without interfering with the prediction process of the model. In this section, various parts of the GCE are described in detail, including the task description of the entire explanation method, the attribution measurement, and the overall framework of the GA-based optimal explainable subsequence search.

A. Task Description

For the KT explainability problem, our study focuses on explaining the decision process of the model, i.e., finding the correlation ($x \xrightarrow{f} y$) of model input x to output y given a model f. In our scenario, determining which question and answer interaction pairs in the input subsequence have the most influence on the final prediction outcome y is of interest. Feature importance attribution [27], [38], [39], [40], a popular technique that provides post hoc and model-agnostic explanations, decomposes the prediction into input units. Each input unit in the model is associated with an attribution score to indicate the degree of its contribution to the prediction. It is an explanation method belonging to the importance attribution of input units. Formally, by deriving K significant answer interaction pairs in the historical answer input sequence for each student and constructing a faithful explanatory subsequence $S_k^* = \{x_1, x_2, \ldots, x_k\} \subseteq S_t$, the model provides evidence to support the prediction. However, in reality, for a large number of predictions about learning outcomes from different learners, we must complete multiple explainable instance tasks., i.e., given the input sequence set $\Phi = \{S_{t_1}, S_{t_2}, \ldots, S_{t_n}\}$, the final explainable subsequence set $\Psi = \{S_{k_1}^*, S_{k_2}^*, \ldots, S_{k_n}^*\}$ is obtained.

B. Attribution Measurement

For the problem of input unit importance attribution measurement, our measurements are different from the general correlations. When the machine learning model is based on the causal relationship between the causal features and the prediction target, the model is truly explainable and can be improved in terms of stability [29], [41]. Therefore, our study proposes attribution measurement from the perspective of causality, starting from the discovery of causal correlations between model inputs and outputs. First, for every instance task, the attribution of the input sequence is constructed from the causality perspective, and formally, the explanatory subsequence S_k^* can be constructed by maximizing the attribution measurement $A(\cdot)$

$$S_k^* = \arg\max_{S_k \subseteq S_t} A(S_k | y_t, f_\theta)\,. \tag{1}$$

Indicator A measures the contribution of each candidate subsequence S_k to the target prediction y_t. For causal explainability, indicator A quantifies the causal effect of S_k. It is possible to study how the model predictions change by directly manipulating the values of the input sequences. Such manipulation is an intervention in causal inference [42], [43], which builds on the $do(\cdot)$ calculus. It cuts off all incoming links to the variable and forces the assignment of a certain value to the variable that is no longer influenced by its causal parent term. The individual causal effect (ICE) [44] is formulated in the attribution function by intervention. It is defined as

$$A(S_k | y_t, f_\theta) = \mathrm{ICE}(S_k) = |Y(do(S_k)) - Y(do(\emptyset))|. \tag{2}$$

Specifically, the answer interaction pair of whether the input sequence is deleted is considered as a control variable and intervened twice: $do(S_k)$ and $do(\emptyset)$, $do(S_k)$ indicates that the original input sequence is processed (i.e., the sequence after being deleted is input into the KT model) and control $do(\emptyset)$ (the original sequence that is not deleted is input into the KT model), where Y is the predictor variable, $Y(do(S_k))$ refers to the predicted value obtained after the deletion in the original sequence is input into the model, and $do(\emptyset)$ is the predicted value of the model obtained without any intervention operation on the original sequence, ICE is the absolute value of the difference value between the results under intervention and control, and the larger value of this difference represents a larger causal effect of S_k on y_t.

C. Genetic Algorithm Hyper-Heuristics

According to the previous section, our explanation method is essentially an optimal subsequence search problem based on combinatorial optimization, i.e., searching for the subsequence with the maximum ICE value from the historical answer interaction pair sequence, which is an NP-hard discrete search problem and difficult to solve by traversal when the historical answer interaction pair sequence is long.

GA [45] is a stochastic global search optimization method. The algorithm process simulates the crossover and mutation phenomena in natural selection and genetics, generates individuals who are more suitable for the environment and stay according to their fitness from the parents, and finally converges to find the optimal solution of the problem, GA has been proved to be able to solve such problems well [46], [47], [48]. Therefore, we introduce GA as the core algorithm to explore the decision basis of the KT model. To the best of our knowledge, our proposed method is the first to apply GA to the deep KT explainable problem.

D. GCE Overall Framework

The overall framework of our proposed GCE is shown in Fig. 2, where Φ represents the set of input sequences and S_M represents the set of generated individuals and their corresponding ICE value obtained for each generation of the population. In particular, due to the specificity of the KT task, we introduce a certain domain knowledge prior to guide the population initialization, while in the actual case, our explainable task is a multi-instance task, i.e., the GA needs to complete multiple instances of the task of searching for the optimal subsequences. In the process of GA search, the ICE value of each generation can be saved as experience, whether within a single instance or between different instances. Therefore, we introduce a global EM, which is constructed in our task as a knowledge causal correlation directed graph used to record and update the ICE values between the sequence of explainable answer pairs and the knowledge concepts behind the questions to be predicted in each generation and apply it to the adaptive mutation strategy in the GA. The details will be presented later.

Algorithm 1 gives the pseudocode of the GCE. The input of the algorithm is the sequence of students' historical answer interaction pairs and the information from the questions to be predicted. The output of the algorithm is the explainable historical answer interaction pair subsequence with the highest ICE

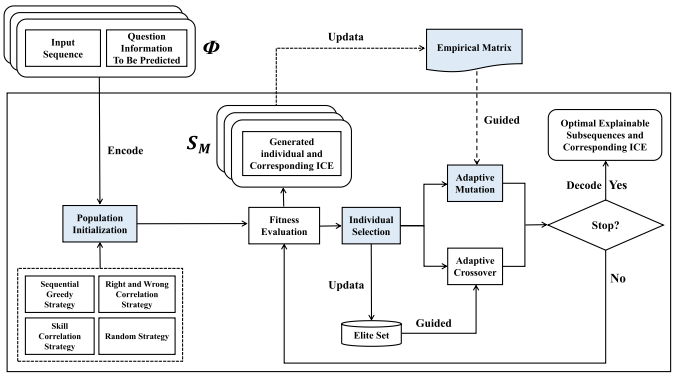

Fig. 2. GCE framework.

Algorithm 1: Pseudocode for GCE

Input: S, interaction pair sequence of students' historical answers; q, question information to be predicted; M, empirical matrix;

Output: S^*, optimal explainable subsequence; ICE_{S^*}, the ICE of S^*;

$i \leftarrow 0$ // i is the current generation
// Population initialization and gene encoding
Pop, $E \leftarrow geneEncode(popInit(S, q))$
// Evaluation of population fitness
$f \leftarrow fitnessEvaluate(\text{Pop})$
// *iterNum* is the total iterations
while $i \leq iterNum$ **do**
  Pop $\leftarrow indSelect(\text{Pop}, f)$
  Pop $\leftarrow crossover(\text{Pop}, f, E, i)$
  Pop $\leftarrow mutation(\text{Pop}, M, q, i)$
  $f, S_M \leftarrow fitnessEvaluate(\text{Pop})$
  $updateEliteSet(\text{Pop}, f)$
  $updateMatrix(M, S_M, q)$
  $i = i + 1$
// Take the highest fitness individual from the elite set
$bestInd \leftarrow getBestInd(E)$
$S^*, ICE_{S^*} \leftarrow geneDecode(bestInd)$
Return S^*, ICE_{S^*}

and its corresponding ICE value searched by the algorithm. First, the initialized population and the elite set are obtained by initialization and genetic coding of the population guided by domain prior knowledge. Then, the population is evaluated for fitness and natural selection to obtain the evolved population. Next, new offspring are obtained by the adaptive crossover strategy guided by the elite set and the adaptive mutation strategy guided by the EM to participate in the subsequent evolution. Iterations of the process are implemented until the termination condition is reached. Finally, the optimal individuals are selected from the elite set and decoded to obtain the final sequence of explainable historical answer interaction pairs and their ICE values. The key steps in Algorithm 1 are described in detail in the next sections.

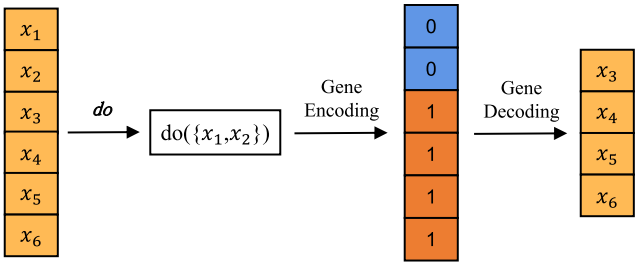

Fig. 3. Gene encoding and decoding strategy.

E. Gene Encoding and Decoding Strategy

As stated in the previous section, our goal is to explore the causal effect of the input subsequence on the predicted outcome of the model and apply the *do* operator to impose an intervention on the input sequence. We use only one form of intervention, i.e., whether to delete a certain pair of interaction pairs in the input sequence, so the gene encoding and decoding can just correspond to the intervention operation with binary numbers; therefore, we only use the *01* code to represent the individual information. After performing the *do* calculus, for example, the gene encoding of $do(\{x_1, x_2\})$ is $[0, 0, 1, 1, 1, 1]$, where 0 represents the deletion of the interaction pair on the corresponding bit and 1 represents the retention of the interaction pair on the corresponding bit, so the gene encoding indicates the deletion of the first and second answer interaction pairs. The decoding operation, similarly, can restore subsequences from individual gene sequences after evolution, as shown schematically in Fig. 3.

F. Fitness Evaluation

In our proposed KT explainable task, we use the ICE value of the input subsequence to measure the importance of the subsequence to the prediction result. The ultimate goal of the task is to find the maximum ICE value, so we directly use the ICE value as the universal fitness of its corresponding individual. The specific calculation formula is provided in (2).

G. Population Initialization

Since the KT task includes more prior knowledge, we introduce four initialization strategies that, together, form the initialized population in the process of population initialization to speed up the population evolution and convergence: the sequential greedy strategy, right and error correlation strategy, skill correlation strategy, and random strategy. The random strategy is introduced to prevent the local optimum problem caused by the complete use of the first three strategies.

1) *Sequential Greedy Strategy:* First, the gene of an individual encoded by all 1 is changed to 0 by sequential search, and the individual with the changed gene is retained if the universal fitness value obtained after the change is higher than that before the change and vice versa. In this way, n individuals are generated.
2) *Right and Wrong Correlation Strategy:* The gene of the individual in the corresponding bit of the interaction pair that is correct or incorrect in the input sequence is set

Algorithm 2: popInit()

Input: S, interactive pair sequence of students' historical answers; q, question information to be predicted;
Output: Pop, initialized population

```
Pop ← ∅
// 1. Sequential greedy strategy
ind ← geneEncode(S)
for gene in ind do
    f1 = fitnessEvaluate(ind)
    gene ← 0
    f2 = fitnessEvaluate(ind)
    if f2 < f1 then
        gene ← 1
    Pop ← Pop ∪ ind
// 2. Knowledge correlation strategy
ind ← geneEncode(S)
for gene in ind do
    if the knowledge concepts of q is equal to the ind then
        gene ← 0
Pop ← Pop ∪ ind
ind ← (ind + 1)%2
Pop ← Pop ∪ ind
// 3. Right and wrong correlation strategy
ind ← geneEncode(S)
for gene in ind do
    if the corresponding answer result is correct then
        gene ← 0
Pop ← Pop ∪ ind
ind ← (ind + 1)%2
Pop ← Pop ∪ ind
// 4. Random strategy
Pop′ ← randomGenerateInd()
Pop ← Pop ∪ ind
Return Pop
```

to 0, and the remainder is set to 1; thus, two individuals are obtained.

3) *Skill Correlation Strategy:* The genes of individuals in the input sequence are set at the corresponding positions of the last predicted knowledge concept related or unrelated interaction pairs to 0 and the rest to 1; thus, two individuals are obtained.
4) *Random Strategy:* A set of $popNum//6$ individuals is randomly generated, where $popNum$ is the number of populations.

The individuals generated by strategies 1, 2, and 3 are copied, and then the individuals formed by strategy 4 are combined to generate a population with the number of individuals $popNum$. In addition, the elite set is initialized by selecting the individuals with the first $popNum//6$ in the population. The preliminary optimized population and elite set are thus obtained. The pseudocode is shown in Algorithm 2.

H. Total Mutation Operator

We introduce a mutation operator called the total mutation operator (TMO), which operates on an individual basis rather than a gene basis. The ICE values we seek are absolute values; generally, the genetic codes of the individuals with the largest negative ICE values and the largest positive ICE values are opposite in the corresponding positions, so we designed this operator to enable the algorithm to evolve the individuals with the largest negative effect into the individuals with the largest positive effect and thus jump out of the local optimum. The mutation strategy of the TMO is different from that of the common flip mutation operator. For details, see Algorithm 3 below.

I. Adaptive Crossover Strategy

In our proposed GA, crossover operations between individuals are used and correspond to bit crossover [46], the elite set crossover [49] is used to speed up the population evolution, and the elite set is stored in the form of a queue, sorted by the higher or lower fitness. After each generation of evolution, if there is an individual who is better than the best individual in the elite set, the elite set performs a queue-in and queue-out operation after which the best individual is in the queue and the worst elite individual is out. In addition, we dynamically adjust the individual crossover probability according to the individual fitness and the current number of evolutions of the population.

Equations (3) and (4) show the calculation of crossover probability

$$p_c = \begin{cases} p_{\max} - (p_{\max} - p_{\min}) * \left(\frac{g}{2G} + \frac{f_i - f_A}{2(f_{bset} - f_A)}\right), & f_i \geq f_A \\ p_{\max}, & f_i < f_A \end{cases} \tag{3}$$

$$p_{\max} = \begin{cases} 0.07, & g < G/4 \\ 0.06, & G/4 \leq g \leq 3G/4 \\ 0.05, & 3G/4 < g \leq G \end{cases} \tag{4}$$

where g is the current number of iterations, G is the total number of iterations, $p_{\min}$ is the minimum crossover probability set manually, f_i is the individual fitness, f_A is the average population fitness, and f_{best} is the highest-population fitness. In the early iterations of individuals and individuals with low fitness, the ability to evolve to the elite set with a high-crossover probability is desirable. In the late iterations of individuals and individuals with low fitness, the crossover probability gradually decreases, which helps to accelerate the speed of population convergence.

J. Adaptive Mutation Strategy

In GCE, the common flip mutation operation is to set the gene from 0 to 1 or 1 to 0. We have different adaptive mutation strategies for the TMO and the common flip mutation operator, and we found that the TMO does not play a large role in the first iteration but affects the convergence speed of the population after the preliminary experiments. Therefore, we set the TMO to play its mutation effect gradually when the population generalization tends to be saturated. Meanwhile,

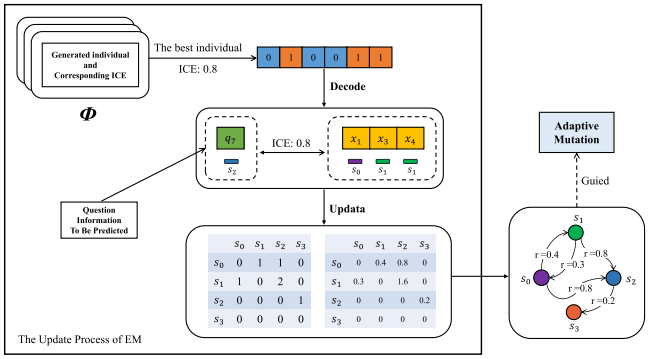

Fig. 4. Update process of EM.

the probability of the ordinary mutation operator consists of three components.

1) The number of current iterations; the later the iteration is, the greater the probability of mutation (the probability of mutation is increased in the later iterations to avoid the algorithm falling into a local optimum while encouraging the generation of new individuals and maintaining individual diversity).
2) The location of the gene; the more backward the location of the gene is, the greater the probability of mutation (motivated by the *learning curve theory* [39], for the KT task, the more recent answer interaction pair sequences should have a greater impact on the model prediction).
3) The degree of causal correlation provided by the knowledge causal correlation directed graph (the higher the causal correlation is, the greater the probability of mutation and vice versa).

The pseudocode of the individual mutation *mutation*() method and the flowchart of the correlation graph update are shown in Algorithm 3 and Fig. 4, where EM is mainly used to update the causal correlation between the knowledge concepts behind the individual gene locus corresponding to the answer interaction pair and the knowledge concepts behind the question to be predicted based on the optimal individual and its corresponding ICE value obtained in each round of iteration. In addition, the individuals obtained during the iterative process can be updated to EM as reliable information regardless of their ICE values. The question is denoted by q, s is the knowledge concept, and r is the causal correlation value in Fig. 4. We maintain two matrices for updating EM. The pseudocode for the method function of calculating the causal correlation is shown in Algorithm 4.

In Algorithm 3, p_{all} is the mutation probability of TMO, p_m is the manually set ordinary mutation probability, *updateNum* is the number of updates to EM, *skillNum* is the number of knowledge concepts in datasets, $w_1, w_2,$ and w_3 are the weights occupied by the mutation probabilities of the three parts of the ordinary mutation, where w_1 is the weight occupied by the EM mutation probability. Since the causal correlation value of EM is not reliable in the early GCE iterations, we introduced confidence factor C and temperature T_2, which increases as the number of iterations increases, and the growth rate can also be controlled by adjusting the temperature T_2. In addition, to ensure that the introduction of EM

Algorithm 3: mutation()

Input: Pop, population; S, interaction pair sequence of students' historical answers; q, question information to be predicted;
Output: Pop, population;

$p_{\text{all}} \leftarrow 0$
if $g/G \geq 0.3$ **then**
  if *No new optimal individual in the population for PopNum//3 consecutive generations* **then**
    $p_{\text{all}} \leftarrow 0.03$ *break*
  if *No new optimal individual in the population for PopNum//4 consecutive generations* **then**
    $p_{\text{all}} \leftarrow 0.02$ *break*
  if *No new optimal individual in the population for PopNum//5 consecutive generations* **then**
    $p_{\text{all}} \leftarrow 0.01$
for *i in* len(Pop) **do**
  if *rand()* $\leq p_{\text{all}}$ **then**
    Pop[i] $\leftarrow$ (Pop[i] + 1)%2
  else
    for *j in* len(Pop[*i*]) **do**
      $r \leftarrow graphPro(S[i][j], q)$
      $w_1 \leftarrow 0.5, w_2 \leftarrow (1 - w_1)/2, w_3 \leftarrow (1 - w_1)/2$
      if $power(updateNum/(2 * skillNum), T_2) \geq 1$ **then**
        $C \leftarrow 1$
      else
        $C \leftarrow power(updateNum/(2 * skillNum), T_2)$
      $B \leftarrow (1 - c * C)/(2 * a)$
      if $rand() \leq p_m * (B * w_2 * g/G + B * w_3 * j/\text{len(Pop}[i]) + C * w_1 * r)$ **then**
        Pop[i][j] $\leftarrow$ (Pop[i][j] + 1)%2
Return Pop

does not affect the overall mutation probability, we introduce a balancing factor B, i.e., when the confidence factor is low, it increases the mutation probability of the other two parts by balancing factor B. The composition formula of the adaptive mutation probability is shown in

$$p = p_m\left(Bw_2\frac{g}{G} + Bw_3\frac{j}{\text{len(Pop}[i])} + Cw_1 r\right). \tag{5}$$

In Algorithm 4, the M_1 matrix is used to record the number of times the knowledge concepts are updated, the M_2 matrix is used to accumulate the ICE values between the two knowledge concepts for each update, and T_1 is the growth ratio. The causal correlation between the two knowledge concepts is obtained by dividing the accumulated ICE values by the cumulative number of updates and normalizing them, and then calculating their T_1 exponential power to obtain the final

Algorithm 4: graphPro()

Input: M_1, the matrix used to calculate the number of updates; M_2, the matrix used to calculate the ICE; i, the ID of knowledge concept to be calculated; j, the ID of knowledge concept to be predicted;

Output: r, the causal relevance;

$ACE \leftarrow M_2/M_1$
$meanACE \leftarrow nonZeroMean(ACE)$
$normalACE \leftarrow (ACE - min(ACE))/(max(ACE) - min(ACE))$
if $normalICE[i][j] == 0$ **then**
  $r \leftarrow meanACE$
else
  $r \leftarrow normalACE[i][j]$
Return $power(r, T_1)$

mutation probability. Additionally, to address the null case, we used the nonzero mean filling *noneZeroMean*() method.

K. Individual Selection and Offspring Generation

In individual selection, we use roulette wheel selection [50] to generate offspring. To increase the explanation length of the final explainable results as much as possible to improve the readability of the explainable subsequences, we added a penalty term in the individual selection process to reduce the selection of those individuals containing more genes with no or fewer causal effects to generate better-explainable offspring, as shown in

$$L = \frac{n_1}{10n}\left(abs\left(\frac{1}{2} - p_b\right) + \frac{1}{2}\right). \tag{6}$$

In (6), n is the length of the original input sequence, n_1 is the number of individuals containing 1 gene, and p_b is the predicted output value of the original input sequence of the model.

IV. Experimental Studies

In this section, some datasets, parameter settings, evaluation indicators, and the summary of compared methods required for the experiments are detailed. GCE achieved good results in the comparison experiments and we also conducted ablation experiments on various subcomponents in GCE to verify the effectiveness of our proposed innovation points. Finally, we summarize and discuss the results of the experiments.

A. Experiments Setup

The proposed explainer, GCE, is a typical post hoc explanation method that is designed to analyze DLKT models. In the experiment section, two representative DLKT models are introduced as the interpreting objects, namely, DKT [19] and SAKT [33]. DKT is generally regarded as the first proposed DLKT model, whereas SAKT is its variant with attention blocks and outperforms DKT. Both models are trained separately on two KT datasets, ASSIST09 [51] and EdNet [52].

TABLE I
Summary Statistics and the AUC of DKT and SAKT in the ASSIST09 and EdNet Data Set

	ASSIST09	EdNet
The number of students	4160	5002
The number of questions	15680	11775
The number of knowledge concepts	167	1837
The total number of answering questions	259105	1658820
The AUC of DKT	0.737	0.722
The AUC of SAKT	0.756	0.752

The ASSIST09 dataset is the most commonly used benchmark dataset in the field, and it is collected and shared by Professor Heffernan.[1] EdNet[2] is the dataset collected from the online learning platform Santa[3] from 2017 to 2019. Referring to previous works, we deleted the records with empty knowledge concept labels, and the records associated with scaffolding problems in the ASSIST09 dataset were also removed. The statistics of two datasets are summarized in Table I, and the performances and hyperparameters of the DKT and SAKT models are collected in Table I and Appendix A, respectively.

The whole experiments were performed on a PC server equipped with an Intel Core i9-12900 K CPU@3.20 GHz, NVIDIA GeForce GTX 3090 Ti GPU, and 64-GB RAM. All methods were implemented using the Pytorch software library [53]. And the detailed parameters of the GCE model can be found in Appendix B.

B. Evaluation Indicators

There are many aspects to consider when evaluating an explanation method. In this article, we focus on the accuracy and readability evaluation. Accuracy reflects the fidelity of the explanation algorithm and the reliability of the interpretation results. Readability indicates whether the explanation result is easily understandable by humans.

To evaluate the accuracy of explanation methods, two sets of experiments are conducted. First, for short sequences, we traverse the solution space to find the optimal explanation subsequence, and the *success rate* is introduced to evaluate the accuracy by contrasting the optimal explanation subsequence with the explanation subsequence generated by explainer.

Success Rate, which measures the rate of successfully finding the optimal subsequence.

Second, we measure the *ICE value* of the generated explanation subsequence, the higher the ICE value, the better the explainer.

ICE Value, the ICE value of the generated optimal subsequence S_k^*

$$\mathrm{ICE}(t) = \mathrm{ICE}\left(S_k^*\right) = |Y\left(do\left(S_k^*\right)\right) - Y(do(\emptyset))|. \tag{7}$$

Third, the length of explanation subsequence is an important indicator for readability of explainer [54], the shorter the explanation length is, the better the readability. To eliminate the effect of sample bias on explanation length, we

[1] https://sites.google.com/site/assistmentsdata/home/2009-2010-assistment-data
[2] https://github.com/riiid/ednet
[3] https://www.aitutorsanta.com/

TABLE II
SUMMARY OF COMPARED METHODS

Method	Summary
Gradient [56]	Importance evaluation of input features according to the gradient of the model
SV [57]	Importance assessment of input features based on game interaction theory
RS [58]	Random search algorithm
GB [45]	Traditional GA with fixed crossover and mutation probabilities
ACMS [59]	Adaptive cosine mutation strategy is added to the traditional GA
AAMS [60]	Adaptive adjusting mutation strategy is added to the traditional GA

introduced *unit explanation length* to measure the readability of the explainer.

Unit Explanation Length, which measures the length of the generated optimal subsequence S_k^*

$$\mathrm{EL}(t) = \mathrm{len}\left(S_k^*\right)/\mathrm{ICE}\left(S_k^*\right). \tag{8}$$

Finally, referring to previous work [28], [55], we introduced a universal evaluation indicator *log odds* to evaluate the performance of all explainable methods. The main idea is to measure the change in confidence of the classification model by calculating the difference in log odds between the original sequence and the generated optimal subsequence. The higher the log odds, the better the explainer.

Log Odds, which measures the change in model confidence between the original sequence S and the generated optimal subsequence S_k^*

$$\mathrm{LO}(t) = \mathrm{logodds}(p_S) - \mathrm{logodds}\left(p_{S_k^*}\right) \tag{9}$$

where $\mathrm{logodds}(p) = log(p/[1-p])$, and p_S and $p_{S_k^*}$ are the classification models' outputs $p \in [0, 1]$ for the original sequence and the generated optimal subsequence, respectively.

In order to ensure the fairness of the comparison experiment, we selected sequences of length 14, 15, 25, and 30 in the datasets as experimental objects and set a unified search space. See Appendix C for details.

C. Summary of Compared Methods

Six methods are selected as comparison methods. Table II shows the detailed descriptions of these six methods, where *Gradient* and *SV* are two classical deep learning explanation methods. *RS* and *GB* are set as baselines for heuristic search algorithms. *ACMS* and *AAMS* are introduced as the state-of-the-art GAs.

D. Accuracy Experimental Result

In the comparison experiment, we compared and analyzed the experimental results of the six methods under four evaluation indicators. For the two explanation methods of *Gradient* and *SV*, we filter the input subsequences that are most associated with the prediction results by their obtained feature importance correlation values and then apply our attribution measurement function to obtain the ICE values and log odds of their most associated subsequences and compare them.

Table III and Fig. 5 show the experimental results under the ICE value and log odds indicator, Table IV shows the experimental results under the success rate indicator, and Fig. 6 shows the average convergence curves of the four search algorithms and the GCE in 300 cases, where $LEN30-ASSIST09-DKT$ represents the experimental results of the DKT model in the ASSIST09 dataset with a prediction length of 30. Based on these results, we can draw the following conclusions.

1) Compared with all the methods, the GCE can get subsequences with higher-ICE values and log odds. At the same time, compared with the other four search algorithms, the GCE can get a population with higher-fitness value through population initialization, which reflects the advantages of our method, and the initialization strategy we proposed has played a role.
2) Compared with the four search algorithms, the GCE has a great improvement in the success rate indicator, which indicates that the GCE can find the optimal subsequence faster and better under the short sequence.
3) Compared with the four search algorithms, the GCE performs better under the longer sequence. This may be because the other four search algorithms have achieved a good performance under the given time complexity under the short sequence, but with the increase of the spatial complexity under the long sequence, the other four algorithms will perform poorly.
4) *SV* and *Gradient*, two explanation methods performed poorly in both the ICE value and log odds indicators, which may be due to the use of non causal methods that easily capture spurious input-output correlations and miss causal relationships.
5) Finally, we also evaluated the average runtime of all methods. The GCE has 9.2% more runtime than the search algorithm's benchmark *RS* method and 7.6% more runtime than the GA's benchmark *GB* method. The *Gradient* method is much faster than other methods, this is because for each sequence, it only needs to calculate the gradient and does not require additional sample generation or search.

E. Readability Experimental Result

For readability comparison, Fig. 7 shows the experimental results of the GCE and other four search algorithms under the unit explanation length indicator. Through the GCE, shorter explainable subsequences with the same ICE value can be obtained. That is, compared with other algorithms, the GCE can obtain explainable subsequences with better readability. This should be related to the introduction of the sparsity penalty-based individual selection method.

F. Ablation Experimental Result

Table V shows the experimental results of the ablation experiments, where GCE-B is the most primitive GA, GCE-I

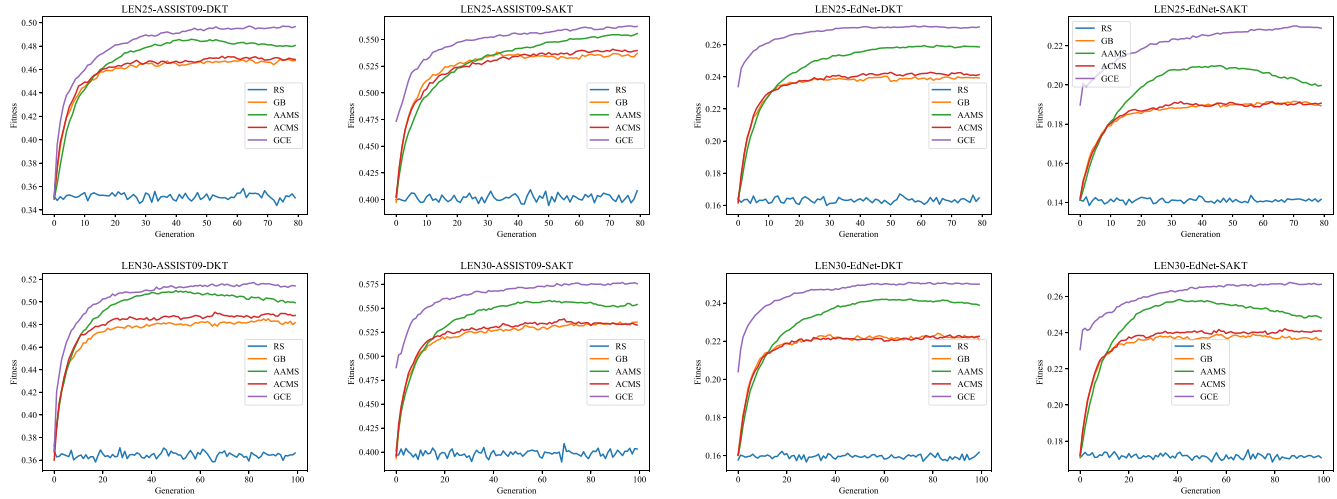

Fig. 5. Average convergence curves of four search algorithms in 300 cases.

TABLE III
COMPARISON OF ICE VALUES OF FINAL EXPERIMENTAL RESULTS AND AVERAGE RUN TIME(S) OBTAINED BY SIX EXPLANATION METHODS. (CARRY OUT FIVE EXPERIMENTS ON THE FOUR SEARCH ALGORITHMS, AND TAKE THE MEAN AND STANDARD DEVIATION AS THE EXPERIMENTAL RESULTS.)

Dataset	Model	Length	Gradient	SV	RS	GB	ACMS	AAMS	GCE
ASSIST09	DKT	14	71.88	96.15	117.81±0.92	**123.05±0.79**	122.56±0.76	122.66±0.77	122.75±0.76
		15	74.98	97.69	121.60±0.89	127.55±0.79	127.05±0.78	126.61±0.75	**128.28±0.76**
		25	89.45	98.66	141.18±0.96	152.89±0.81	153.10±0.82	154.91±0.83	**157.01±0.79**
		30	94.33	100.13	139.73±0.99	154.33±0.82	155.52±0.80	155.03±0.79	**160.07±0.78**
EdNet	DKT	14	61.72	57.83	69.56±0.72	75.65±0.65	75.49±0.59	74.87±0.55	**75.78±0.55**
		15	65.65	56.94	72.31±0.72	77.79±0.64	77.84±0.60	77.51±0.62	**77.86±0.61**
		25	62.90	54.89	68.40±0.69	79.55±0.65	80.30±0.64	80.50±0.60	**81.80±0.61**
		30	66.42	53.27	64.02±0.76	73.47±0.61	74.36±0.63	75.01±0.59	**76.47±0.54**
ASSIST09	SAKT	14	70.86	73.71	143.39±0.90	150.11±0.86	145.95±0.83	146.51±0.84	**154.04±0.84**
		15	71.78	78.64	148.27±0.91	156.25±0.82	154.97±0.82	155.53±0.79	**159.87±0.80**
		25	72.70	89.45	155.10±1.3	172.18±0.91	173.50±0.90	171.51±0.86	**179.43±0.84**
		30	64.30	88.34	150.55±1.3	168.17±0.92	168.41±0.93	169.38±0.90	**179.21±0.86**
EdNet	SAKT	14	58.42	71.15	89.69±0.73	94.55±0.59	94.52±0.61	94.90±0.60	**96.38±0.58**
		15	77.79	69.69	89.04±0.71	94.32±0.62	94.37±0.58	94.58±0.57	**95.59±0.57**
		25	26.14	44.36	55.45±0.65	67.50±0.56	68.76±0.57	68.82±0.56	**73.46±0.54**
		30	69.09	55.80	68.91±0.79	79.63±0.65	80.68±0.64	81.92±0.60	**86.04±0.59**
Average run time(per sequence)			0.11	6.12	14.63	14.85	14.75	14.72	15.98

TABLE IV
COMPARISON OF SUCCESS RATE(%) OF FINAL EXPERIMENTAL RESULTS OBTAINED BY SIX EXPLANATION METHODS. (CARRY OUT FIVE EXPERIMENTS, AND TAKE THE MEAN AND STANDARD DEVIATION AS THE EXPERIMENTAL RESULTS.)

Dataset	Model	Length	RS	GB	ACMS	AAMS	GCE
ASSIST09	DKT	14	10.7±1.8	57.7±3.6	55.3±4.0	60.1±3.2	**71.0±4.8**
		15	11.3±2.0	62.7±3.8	63.0±4.4	64.0±3.7	**74.3±5.0**
EdNet	DKT	14	8.0±1.6	62.3±4.0	63.7±3.7	59.0±3.9	**75.7±3.0**
		15	6.7±1.4	54.7±3.2	54.7±4.9	54.2±3.9	**78.7±5.8**
ASSIST09	SAKT	14	6.7±1.6	30.3±2.7	26.7±2.3	28.2±2.1	**51.6±3.1**
		15	6.0±1.6	47.7±2.6	52.3±3.6	53.1±2.7	**64.7±3.5**
EdNet	SAKT	14	7.7±1.5	38.7±2.4	40.3±2.8	43.2±2.1	**64.0±3.8**
		15	5.0±1.1	55.3±3.9	54.3±3.0	53.0±2.9	**66.3±4.1**

is the algorithm with the addition of our initialization method, and GCE-NG is the algorithm after removing the EM on the complete GCE. This shows that, on the whole, the introduction of the population initialization strategy, adaptive crossover strategy, adaptive mutation strategy, and EM we designed in the GCE is effective.

V. APPLICATION

A. Explainablity Case Analysis

The explanation results are assessed in a typical and real case, as shown in Fig. 8. Given a trained DLKT model and a learner's practice sequence as input, the input consists of 14 consecutive question interaction pairs, and the prediction

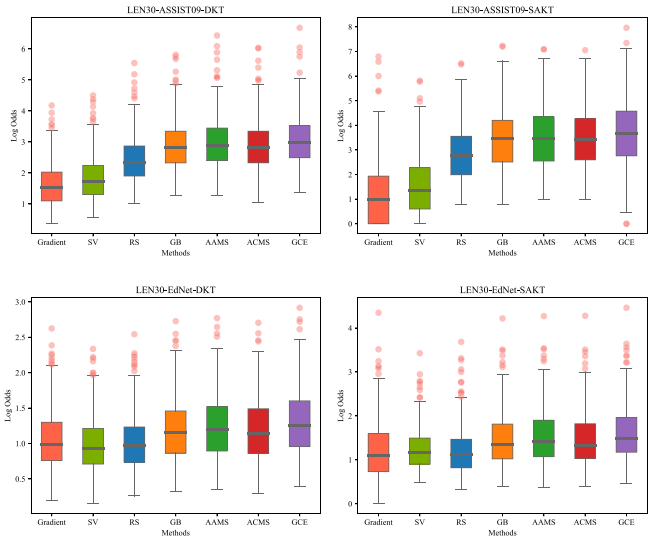

Fig. 6. Average log odds of six methods in 300 cases.

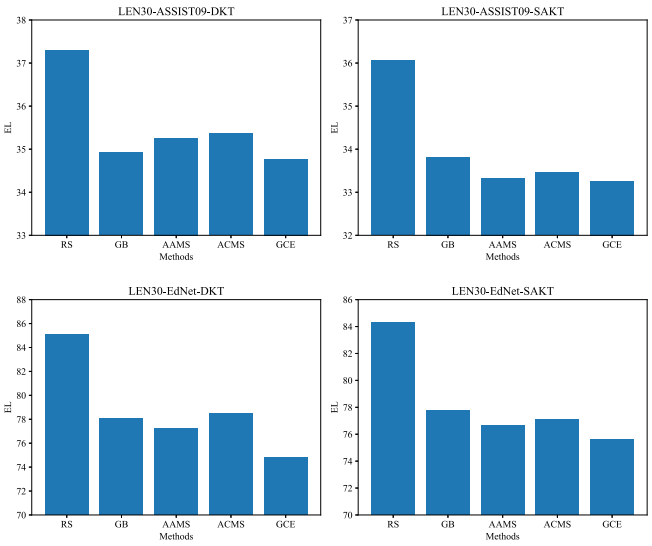

Fig. 7. Unit ICE average explanation length of four search algorithms in 300 cases.

result of 0.87, the probability of correctly answering the next question, is obtained from the output. By using the proposed explainer GCE, the final prediction output can be obtained with the largest causal effect of questions 1 to 3, 5 to7 and 10 to 11 in the historical interaction pairs. The subsequence formed by the answer pairs they consist of is the explanation subsequence, which contributes the most to the prediction task. In this case, by analyzing the knowledge concept of each question, we can draw some preliminary information:

1) The *SV* method and the *Gradient* method focus more on interaction pairs, but the ICE value is not as high as GCE.
2) The GCE pays more attention to the interaction of knowledge concepts related to prediction questions. It believes that in addition to the knowledge concepts of the prediction question itself, *Probability of a Single Event* and *Percent Of* are also important prediction criteria for the model.

TABLE V
COMPARISON OF ICE VALUES IN ABLATION EXPERIMENTS

Dataset	Model	Length	GCE-B	GCE-I	GCE-NG	GCE
ASSIST09	DKT	14	123.05	122.73	**123.52**	122.75
		15	127.55	128.03	128.11	**128.28**
		25	152.89	154.42	155.91	**157.01**
		30	154.33	156.04	157.57	**158.74**
EdNet	DKT	14	75.65	**75.86**	75.63	75.78
		15	77.79	77.85	77.70	**77.86**
		25	79.55	80.30	81.47	**81.80**
		30	73.47	72.46	75.60	**76.25**
ASSIST09	SAKT	14	150.11	154.09	**154.54**	154.04
		15	156.25	159.44	159.74	**159.87**
		25	172.18	177.29	179.00	**179.43**
		30	168.17	175.20	177.31	**179.21**
EdNet	SAKT	14	94.55	95.99	95.94	**96.38**
		15	94.32	94.93	95.18	**95.59**
		25	67.50	69.95	72.26	**73.46**
		30	79.63	82.26	84.21	**86.04**

3) The *SV* method believes that other knowledge concepts, such as *Venn Diagram* and *Volume Rectangular Prism*, are also one of the basis for model decision-making, but intuitively, these two knowledge concepts are not significantly related to the knowledge concepts of the question to be predicted.
4) The *Gradient* method, despite its high-ICE value, only focuses on historical interaction pairs that are consistent with the model's predicted results. This is obviously unreasonable, as incorrect answers may also enhance the level of mastery of the knowledge state.

This case reflects that GCE, from a causal perspective, strongly punishes interactions outside the region of interest, and these results can help us in educational analysis to some extent.

B. Knowledge Structure Discovery

In addition, we used these explainable results to help us in knowledge structure discovery. Specifically, we obtained the final explainable subsequence in 2400 instances by GCE, considering that if the predicted questions and knowledge concepts co-occur with the question and knowledge concepts in the explainable subsequence, it indicates that there is some relationship between them. Based on this sampling of these data, a network graph between questions, and knowledge concepts were constructed. In Figs. 9 and 10, the node labels represent the IDs and names of the corresponding questions or knowledge concepts in the original data. These interesting results demonstrate that our explanation approach can at least partially recover the internal knowledge-level relationships captured by the constructed DLKT model, while further research may be needed to explore their potential meaning from an educational perspective.

C. DLKT Model Comparison

Similarly, we analyzed the differences between the two deep KT models, SAKT and DKT, based on these explainable results in the same sequence length and explainable results in

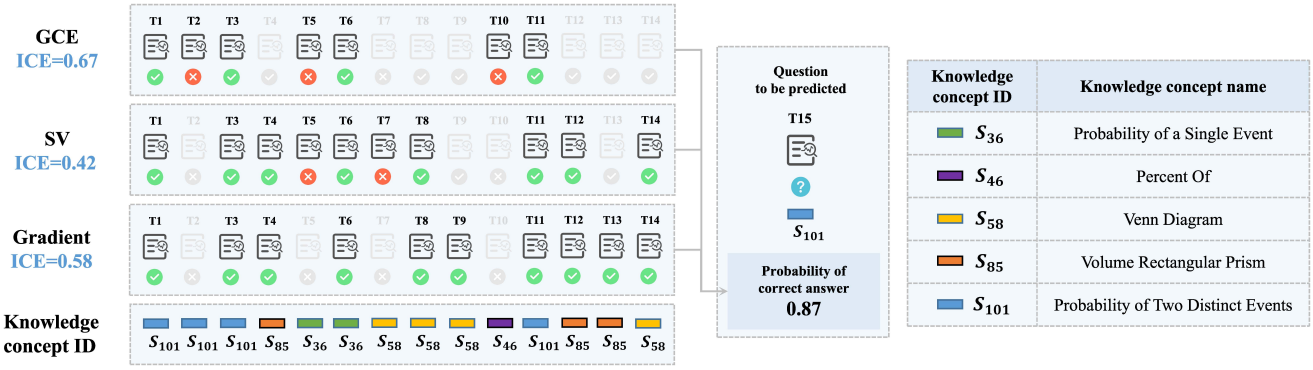

Fig. 8. DLKT model for decision attribution explanation case.

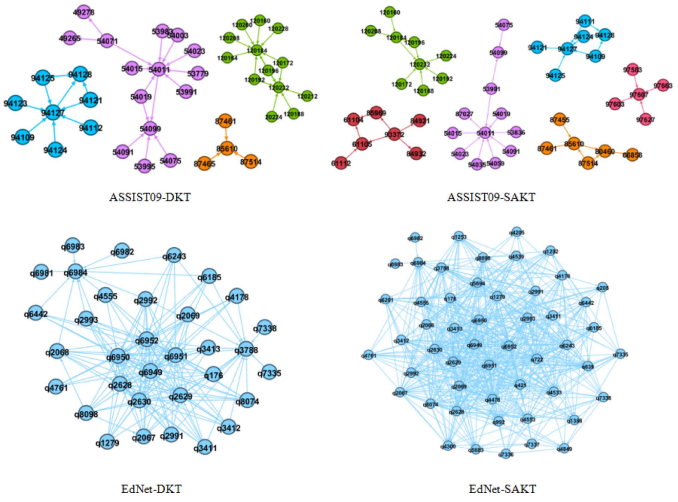

Fig. 9. Relationship graph between questions.

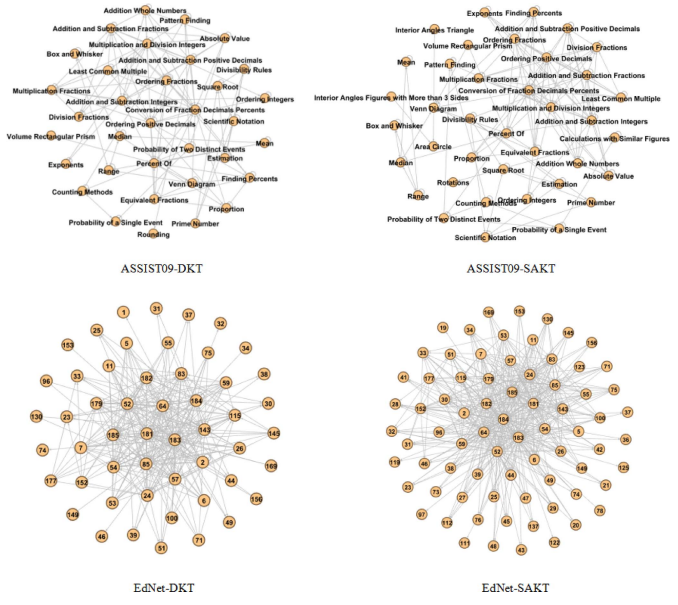

Fig. 10. Relationship graph between knowledge concepts.

the same datasets in terms of the distribution of the explainable length, the distribution of the position of the explainable subsequences, and the distribution of ICE. In Fig. 11, the left side shows that the SAKT model had a longer explainable

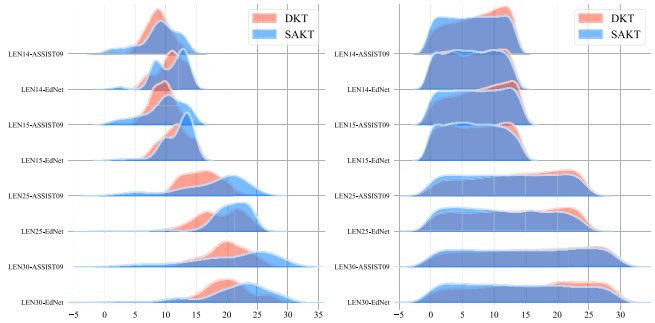

Fig. 11. Distribution of explanation length and subsequence position (the left panel shows the explanation length distribution, and the right panel shows the explanation subsequence distribution).

length than the DKT model, indicating that the model focuses on more and broader sequences. The right side shows the distribution of explainable subsequences of the two models. The DKT model pays more attention to the historical answer interaction pairs close to the predicted answers, indicating that the DKT model pays more attention to the recent answer performance of students. Additionally, through comparison experiments and ablation experiments as in Tables III and IV, we found that the ICE values captured by our method were significantly different for these two models; the ICE values obtained from the SAKT model were higher than those from the DKT model, which to some extent indicates that the SAKT model is more sensitive to our intervention of using partial input answer pairs removal, thus focusing more on the student's historical answer performance, while the DKT model showed less sensitivity, suggesting that the DKT model may focus more on predicting some information about the questions themselves, such as question difficulty.

VI. Conclusion and Future Work

In this article, we proposed a GCE for deep KT. We designed a set of explanation methods based on the importance attribution of input units to design a set of encoding methods under the domain that can transform our explainable problem into a GA to solve the NP-hard discrete search problem of combinatorial optimization and design an attribution measurement framework for evaluating explainable subsequences while making some improvements based on

TABLE VI
PARAMETER SETTINGS OF DLKT MODELS (DKT AND SAKT)

KT Model parameter	value	Description
batchSize	32	batch size
lr	0.001	learning rate
dropout	0.2	probability of dropout
embedDim	128	the dim of question embedding
l2Weight	1e-5	l_2 weight decay rate
epoch	100	training epoch
minSeqLen	3	the minimum length of input sequence
maxSeqLen	200	the maximum length of input sequence
hiddenSize	256	the hidden layer size of LSTM for DKT
numAttnLayer	4	the number of attention layer for SAKT
numHeads	4	the number of head of attention layer for SAKT

TABLE VII
PARAMETER SETTINGS OF THE GCE

GA parameter	value	Description
n	10-30	the length of input sequence
popNum	p	the number of population
iterNum	i	the number of iteration
p_{min}	0.05	the minimum of crossover rate
p_m	0.05	the mutation rate
w_1	0.5	the mutation weight of EM
T_1	1.2	the growth ratio of correlation intensity of EM
T_2	0.8	the growth ratio of confidence factor

TABLE VIII
CORRESPONDENCE BETWEEN LENGTH, TIME COMPLEXITY, AND SOLUTIONSPACE SIZE (FOR GA, THE COMPLEXITY IS EQUAL TO THE NUMBER OF POPULATION MULTIPLY BY THE NUMBER OF ITERATIONS)

length	time complexity	solution space size
14	40×40	2^{14}
15	50×50	2^{15}
25	80×80	2^{25}
30	100×100	2^{30}

traditional GA, three of which are important parts of our improvements. First, we proposed a multistrategy initialization method based on domain prior knowledge. Second, we introduced a global EM capable of capturing causal correlation values within and across instances between DLKT model inputs and used them to guide individual genetic mutation, which is a general method that can be applied not only to our task but also to other tasks. Third, a sparsity penalty-based individual selection method was designed for generating sparse offspring, resulting in explainable results with short explanation lengths. Comparative experiments showed that our method outperformed other methods in terms of average ICE value, log odds, explanation length, and success rate, and it can also obtain better-explainable subsequences faster and better. The ablation experiments showed that all of our proposed innovations were effective in the GA. Finally, we presented a case study using the explainable results and uncover the underlying knowledge structure, i.e., the directed correlation graphs between questions and questions, knowledge concepts and knowledge concepts, and compare the differences between the two models through these data.

In summary, the GCE provides a certain approach for using GA to solve post hoc explainable problems. Due to its gradient free and insensitive to local minima, combined with appropriate attribution measurement frameworks, GA is more cost-effective, flexible, and decoupled compared to traditional machine learning methods. In the future, we will apply GA to other fields based on the design idea of GCE, such as natural language processing, computer vision, GNN, and other more common fields. As long as explainable problems can be transformed into search tasks, such as subsequence/subgraph search problems by special means, we believe that GA will achieve significant achievements.

APPENDIX A

See Table VI.

APPENDIX B

See Table VII.

APPENDIX C

See Table VIII.

REFERENCES

[1] J. R. Anderson, C. F. Boyle, A. T. Corbett, and M. W. Lewis, "Cognitive modeling and intelligent tutoring," *Artif. Intell.*, vol. 42, no. 1, pp. 7–49, 1990.

[2] G. Abdelrahman, Q. Wang, and B. P. Nunes, "Knowledge tracing: A survey," *ACM Comput. Surv.*, vol. 55, no. 11, pp. 1–37, 2023.

[3] Z. Liu, J. Chen, and W. Luo, "Recent advances on deep learning based knowledge tracing," in *Proc. Int. Conf. Web Search Data Min.*, 2023, pp. 1295–1296.

[4] J. Psotka, L. D. Massey, and S. A. Mutter, *Intelligent Tutoring Systems: Lessons Learned*. London, U.K.: Psychol. Press, 1988.

[5] Z. Pardos, Y. Bergner, D. Seaton, and D. Pritchard, "Adapting Bayesian knowledge tracing to a massive open online course in edX," in *Proc. Int. Conf. Educ. Data Min.*, 2013, pp. 137–144.

[6] Z. Wang, J. Zhu, X. Li, Z. Hu, and M. Zhang, "Structured knowledge tracing models for student assessment on Coursera," in *Proc. ACM Conf. Learn. Scale*, 2016, pp. 209–212.

[7] F. Xu, H. Uszkoreit, Y. Du, W. Fan, D. Zhao, and J. Zhu, "Explainable AI: A brief survey on history, research areas, approaches and challenges," in *Natural Language Processing and Chinese Computing*. Cham, Switzerland: Springer, 2019, pp. 563–574.

[8] G. Montavon, W. Samek, and K. Müller, "Methods for interpreting and understanding deep neural networks," *Digit. Signal Process.*, vol. 73, pp. 1–15, Feb. 2018.

[9] Y. Zhang, P. Tiňo, A. Leonardis, and K. Tang, "A survey on neural network interpretability," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 5, no. 5, pp. 726–742, Oct. 2021.

[10] F.-L. Fan, J. Xiong, M. Li, and G. Wang, "On interpretability of artificial neural networks: A survey," *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 5, no. 6, pp. 741–760, Nov. 2021.

[11] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *Commun. ACM*, vol. 63, no. 1, pp. 68–77, 2019.

[12] D. Li, Y. Liu, J. Huang, and Z. Wang, "A trustworthy view on explainable artificial intelligence method evaluation," *Computer*, vol. 56, no. 4, pp. 50–60, Apr. 2023.

[13] Y. Lu, D. Wang, P. Chen, Q. Meng, and S. Yu, "Interpreting deep learning models for knowledge tracing," *Int. J. Artif. Intell. Educ.*, to be published.

[14] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User Model. User Adapt. Interact.*, vol. 4, no. 4, pp. 253–278, 1994.

[15] S. E. Embretson and S. P. Reise, *Item Response Theory*. New York, NY, USA: Psychol. Press, 2013.

[16] H. Cen, K. Koedinger, and B. Junker, "Comparing two IRT models for conjunctive skills," in *Proc. 9th Int. Conf. Intell. Tutoring Syst.*, 2008, pp. 796–798.

[17] P. I. Pavlik Jr., H. Cen, and K. R. Koedinger, "Performance factors analysis–a new alternative to knowledge tracing," in *Proc. Int. Conf. Artif. Intell. Educ.*, 2009, pp. 531–538.
[18] J.-J. Vie and H. Kashima, "Knowledge tracing machines: Factorization machines for knowledge tracing," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 750–757.
[19] C. Piech et al., "Deep knowledge tracing," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2015, pp. 505–513.
[20] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
[21] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller, "Striving for simplicity: The all convolutional net," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015, pp. 1–14.
[22] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLoS One*, vol. 10, no. 7, 2015, Art. no. e0130140.
[23] J. Li et al., "Instance-wise or class-wise? A tale of neighbor Shapley for concept-based explanation," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 3664–3672.
[24] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 4768–4777.
[25] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should i trust you?': Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2016, pp. 1135–1144.
[26] D. Bau et al., "GAN dissection: Visualizing and understanding generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 1–19.
[27] A. Chattopadhyay, P. Manupriya, A. Sarkar, and V. N. Balasubramanian, "Neural network attributions: A causal perspective," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 981–990.
[28] P. Schwab and W. Karlen, "CXPlain: Causal explanations for model interpretation under uncertainty," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2019, pp. 10220–10230.
[29] X. Wang, Y. Wu, A. Zhang, F. Feng, X. He, and T.-S. Chua, "Reinforced causal explainer for graph neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 2297–2309, Feb. 2023.
[30] C.-K. Yeung, "Deep-IRT: Make deep learning based knowledge tracing explainable using item response theory," 2019, *arXiv:1904.11738*.
[31] J. Zhang, X. Shi, I. King, and D.-Y. Yeung, "Dynamic key-value memory networks for knowledge tracing," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 765–774.
[32] Q. Liu et al., "EKT: Exercise-aware knowledge tracing for student performance prediction," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 1, pp. 100–115, Jan. 2021.
[33] S. Pandey and G. Karypis, "A self-attentive model for knowledge tracing," 2019, *arXiv:1907.06837*.
[34] S. Pandey and J. Srivastava, "RKT: Relation-aware self-attention for knowledge tracing," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 1205–1214.
[35] A. Ghosh, N. Heffernan, and A. S. Lan, "Context-aware attentive knowledge tracing," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2020, pp. 2330–2339.
[36] Q. Hu and H. Rangwala, "Reliable deep grade prediction with uncertainty estimation," in *Proc. 9th Int. Conf. Learn. Anal. Knowl.*, 2019, pp. 76–85.
[37] D. Wang, Y. Lu, Z. Zhang, and P. Chen, "A generic interpreting method for knowledge tracing models," in *Proc. 23rd Int. Conf. Artif. Intell. Educ.*, 2022, pp. 573–580.
[38] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 618–626.
[39] R. C. Murray et al., "Revealing the learning in learning curves," in *Proc. 16th Int. Conf. Artif. Intell. Educ.*, 2013, pp. 473–482.
[40] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3319–3328.
[41] R. Moraffah, M. Karami, R. Guo, A. Raglin, and H. Liu, "Causal interpretability for machine learning—Problems, methods and evaluation," *ACM SIGKDD Explorations Newslett.*, vol. 22, no. 1, pp. 18–33, 2020.
[42] J. Pearl, *Causality*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[43] M. Glymour, J. Pearl, and N. P. Jewell, *Causal Inference in Statistics: A Primer*. Chichester, U.K.: Wiley, 2016.
[44] J. Pearl, *Models, Reasoning and Inference*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[45] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed. Berlin, Germany: Springer-Verlag, 1996.
[46] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: Past, present, and future," *Multimedia Tools Appl.*, vol. 80, no. 5, pp. 8091–8126, 2021.
[47] N. W. Hasan, A. S. Saudi, M. I. Khalil, and H. M. Abbas, "A genetic algorithm approach to automate architecture design for acoustic scene classification," *IEEE Trans. Evol. Comput.*, vol. 27, no. 2, pp. 222–236, Apr. 2023.
[48] M. A. Ardeh, Y. Mei, M. Zhang, and X. Yao, "Knowledge transfer genetic programming with auxiliary population for solving uncertain capacitated arc routing problem," *IEEE Trans. Evol. Comput.*, vol. 27, no. 2, pp. 311–325, Apr. 2023.
[49] N. Saini, "Review of selection methods in genetic algorithms," *Int. J. Eng. Comput. Sci.*, vol. 6, no. 12, pp. 22261–22263, 2017.
[50] K. Jebari and M. Madiafi, "Selection methods for genetic algorithms," *Int. J. Emerg. Sci.*, vol. 3, no. 4, pp. 333–344, 2013.
[51] M. Feng, N. Heffernan, and K. Koedinger, "Addressing the assessment challenge with an online system that tutors as it assesses," *User Model. User Adapt. Interact.*, vol. 19, no. 3, pp. 243–266, 2009.
[52] Y. Choi et al., "EdNet: A large-scale hierarchical dataset in education," in *Proc. 21st Int. Conf. Artif. Intell. Educ.*, 2020, pp. 69–73.
[53] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.
[54] R. R. Hoffman, S. T. Mueller, G. Klein, and J. Litman, "Metrics for explainable AI: Challenges and prospects," 2018, *arXiv:1812.04608*.
[55] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3145–3153.
[56] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *Proc. 2nd Int. Conf. Learn. Represent.*, 2014, pp. 1–8.
[57] J. Castro, D. Gómez, and J. Tejada, "Polynomial calculation of the Shapley value based on sampling," *Comput. Oper. Res.*, vol. 36, no. 5, pp. 1726–1730, 2009.
[58] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. 2, pp. 281–305, 2012.
[59] G. Hu, B. Du, X. Wang, and G. Wei, "An enhanced black widow optimization algorithm for feature selection," *Knowl. Based Syst.*, vol. 235, Jan. 2022, Art. no. 107638.
[60] W.-X. Wang, K.-S. Li, X.-Z. Tao, and F.-H. Gu, "An improved MOEA/D algorithm with an adaptive evolutionary strategy," *Inf. Sci.*, vol. 539, pp. 1–15, Oct. 2020.

Qing Li received the Ph.D. degree in radio physics from Central China Normal University, Wuhan, China, in 2011.

She is currently an Associate Professor with the National Engineering Research Center of Educational Big Data, Central China Normal University. Her research interests include learning analytics, artificial intelligence, and educational information technology.

Xin Yuan received the B.Sc. degree in digital media technology from the University of South China, Hengyang, China, in 2021. He is currently pursuing the M.Sc. degree with the Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan, Hubei, China.

His current research interests are explainable knowledge tracing and causal inference.

Sannyuya Liu (Member, IEEE) received the Ph.D. degree in systems engineering from Huazhong University of Science and Technology, Wuhan, China, in 2003.

He is currently a Professor and the Ph.D. Supervisor with the National Engineering Research Center of Educational Big Data and National Engineering Research Center for E-Learning, Central China Normal University, Wuhan. His research interests include computer application, artificial intelligence, and educational information technology.

Lu Gao received the B.Sc. degree in software engineering from Central China Normal University, Wuhan, China, in 2015, where she is currently pursuing the Ph.D. degree with the Faculty of Artificial Intelligence in Education.

Her current research interests are explainable knowledge tracing and causal inference.

Tianyu Wei received the B.Sc. degree in applied mathematics from Tianjin University, Tianjin, China, in 2017, and the master's degree in computer science and technology from Central China Normal University, Wuhan, China, in 2021, where he is currently pursuing the Ph.D. degree with the Faculty of Artificial Intelligence in Education.

His research interests include wireless sensor networks, recommendation algorithms, artificial intelligence, and machine learning.

Xiaoxuan Shen received the Ph.D. degree in educational information technology from Central China Normal University, Wuhan, China, in 2020.

He is currently working as a Postdoctoral Fellow with the National Engineering Research Center of Educational Big Data, Central China Normal University. His research interests include deep learning, representation learning, and their applications in recommendation system and intelligent e-learning environment.

Jianwen Sun received the Ph.D. degree in educational technology from Central China Normal University, Wuhan, China, in 2011.

He is currently an Associate Professor and the Ph.D. Supervisor with the National Engineering Research Center of Educational Big Data, Central China Normal University. His research interests include educational data mining, explainable artificial intelligence, and intelligent tutoring system.

Dr. Sun is a member of the China Computer Federation (CCF).