STEM Education

STEM Education, 2 (2): 157–172
DOI: 10.3934/steme.2022011
Received: December 2021
Accepted: May 2022

https://www.aimsciences.org/journal/A0000-0006

*Case study*

# Personalized exercise recommendation method based on causal deep learning: Experiments and implications

**Suhua Wang[1], Zhiqiang Ma[1], Hongjie Ji[2], Tong Liu[2], Anqi Chen[2] and Dawei Zhao[1,*]**

[1] Department of Computer Science, Changchun Humanities and Sciences College, Changchun, 130117, China; wangsuhua@ccrw.edu.cn; mazq@nenu.edu.cn; zhaodawei@ccrw.edu.cn
[2] School of Information Science and Technology, Northeast Normal University, Changchun 130117, China; jihj328@nenu.edu.cn; liut790@nenu.edu.cn; chenaq669@nenu.edu.cn

*** Correspondence:** Email: zhaodawei@ccrw.edu.cn; Tel: +86-431-84536338

Academic Editor: Jun Shen

**Abstract:** The COVID-19 pandemic has accelerated innovations for supporting learning and teaching online. However, online learning also means a reduction of opportunities in direct communication between teachers and students. Given the inevitable diversity in learning progress and achievements for individual online learners, it is difficult for teachers to give personalized guidance to a large number of students. The personalized guidance may cover many aspects, including recommending tailored exercises to a specific student according to the student's knowledge gaps on a subject. In this paper, we propose a personalized exercise recommendation method named causal deep learning (CDL) based on the combination of causal inference and deep learning. Deep learning is used to train and generate initial feature representations for the students and the exercises, and intervention algorithms based on causal inference are then applied to further tune these feature representations. Afterwards, deep learning is again used to predict individual students' score ratings on exercises, from which the Top-N ranked exercises are recommended to similar students who likely need enhancing of skills and understanding of the subject areas indicated by the chosen exercises. Experiments of CDL and four baseline methods on two real-world datasets demonstrate that CDL is superior to the existing methods in terms of capturing students' knowledge gaps in learning and more accurately recommending appropriate exercises to individual students to help bridge their knowledge gaps.

**Keywords:** online learning, personalized exercise recommendation, causal inference, deep learning

## 1. Introduction

With more learners engaging with online learning, with massive amounts of learning resources readily available either freely or paid, choosing appropriate learning resources from the massive stack of online materials becomes an important problem for the learners in terms of both efficiently managing the time for study and achieving the best possible learning outcomes. One of these problems is how the learning system can provide a particular learner with well-fitted online exercises on a topic under study from a large number of accumulated exercises on the topic hosted in the learning system [1]. This is commonly known as personalized recommendation in e-learning.

Some popular recommendation methods have been proposed by many researchers in the past two decades. For example, Walker et al. [2] proposed a method of personalized exercise recommendation by collaborative filtering in 2004. Hsu et al. [3] used a collaborative filtering algorithm to analyze the books and documents read by students and give personalized elective course recommendations. Milicevic et al. [4] provided different questions to different students guided by the individual students' learning habits and hobbies. Segal et al. [5] used a personalized recommendation method called EduRank to feed exercises with different difficulty levels to different cohorts of students according to similarity scores shared by the students. Toledo et al. [6] used a collaborative filtering method to achieve similar recommendations for students studying programming online. Wu et al. [7] recommended learning materials to students through a fuzzy matching method. Dwivedi et al. [8] made a further improvement on recommendations by considering multiple factors, such as the student's knowledge level, learning approaches, learning objectives and so on. Although these methods had certain positive effects on student learning, none of them could accurately catch a particular student's grasp of a specific topic in relation to the levels or "knowledge points" associated with the topic. Recently, Jiang [9] et al. used the knowledge points associated with the exercises to provide recommendations to students, which has been well received by many educators [10].

Machine learning methods have been applied to exercise recommendations in recent years. In [11], the student's mastery of knowledge points was fed to the recursive neural network to obtain better recommendation results. In [12], the authors added a new knowledge representation to deep learning to improve the recommendation results, which were further improved by subsequent studies [13–15]. However, machine learning algorithms by nature are highly influenced by the training data, which sometimes can create the bias problem. Also, machine learning mainly identifies the correlations between samples and labels but not the causality between them, if such is not properly indexed. However, the causal relationship between learning events is critical most of the time, as knowledge acquisition is often from knowns to unknowns.

Causal inference could be used to find the cause and effect between two events (or two datasets), in addition to the surface correlation [16–18]. In terms of exercise recommendations, if a student did the earlier exercises well, the student would be likely to do the next batch of exercises well. If the student consistently did all successive sets of exercises well, we would have high confidence to expect the student to do the examination well. These causal relationships should be captured in the student's interactive records in the system. Of course, a similar pattern may be found in the datasets of other students. If so, not only can the same causal relationship be consolidated, but also the students can be properly grouped so that they receive the same recommendations from the system. Hence, learning based on recommendations through causal inference may provide the best possible support to students engaging with online studies. A framework of applying causality in machine learning was proposed in a few studies recently [19–21]. However, how to effectively capture the causal relationships for automatic exercise recommendations in machine learning is still case-specific and has different challenges.

In this study, we propose a causal deep learning (CDL) model and an implementation of a new personalized exercise recommendation algorithm to evaluate the usefulness and effectiveness of this approach in assisting student learning online. Our experiments were conducted using the databases of a self-developed online learning system by the Northeast Normal University of China and the ACM KDD Cup competition. The performances of CDL on the two datasets are benchmarked with those of four existing methods.

In Section 2, the framework of CDL is introduced, along with the representations of the main parts of CDL. Section 3 outlines the experimental conditions, and Section 4 presents the experimental results, data analysis and discussions on the performances of CDL with respect to the other four existing methods. A brief conclusion is summarized in Section 5.

## 2. The framework for exercise recommendation based on causal deep learning (CDL)

### 2.1. The framework

This model mainly targets improving the accuracy of exercise recommendations through personalized recommendation driven by the potential causal relationships existing in both the exercises and the records of the student's historic performances in attempting the exercises. For specific students, deep learning neural networks are used to learn the different scores they obtained on different exercises in historical records. The inputs to the network are extracted from both the student's historic records and the exercise database. The output is the prediction of a set of ranked scores of the chosen exercises for the student. The framework of CDL is outlined in Figure 1.

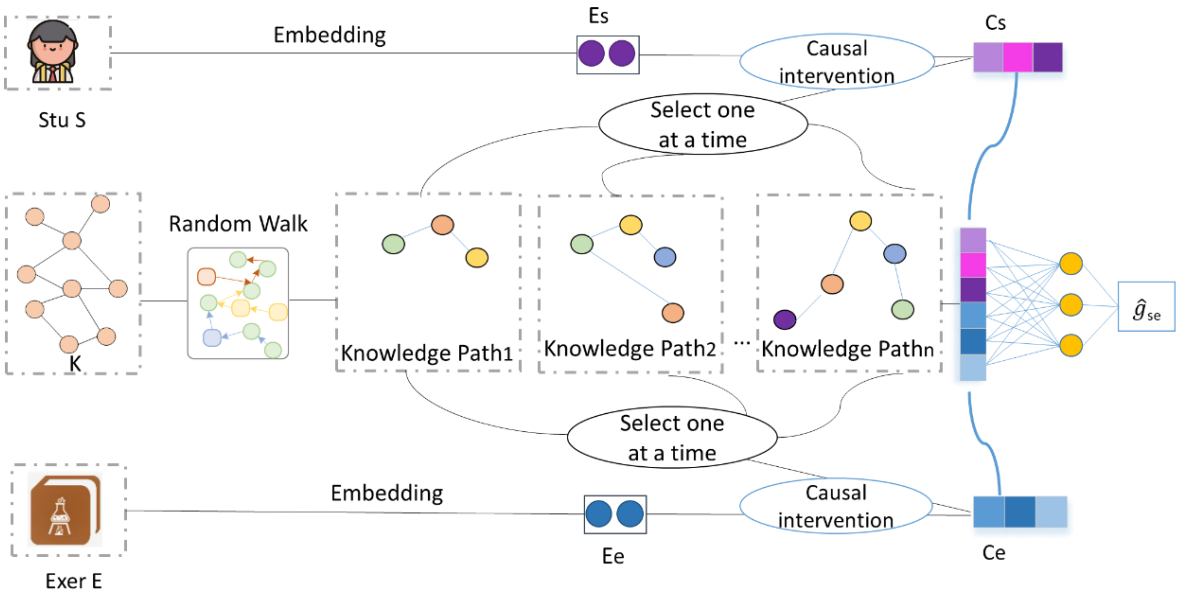



**Figure 1.** Framework for causal deep learning **(**CDL**)** (*Cs* is the student input embedded with causal interventions; *Cs* is the exercise input embedded with causal interventions)

The two inputs work in a sequential way, rather than concurrently in other circumstances. The student's historic performance after incorporating any potential causal relationships (*Cs*) is fed to the network first to "identify the student's most appropriate academic level or capacity." Then, a set of causally ranked candidate exercises (*Ce*) is fed to the "personalized" network to forecast "that specific student's performances" in the selected exercises. The predicted scores with different

candidate exercises are sorted in ascending order [22]. A smaller predicted score corresponds to a lower likelihood that the student would have been familiar with the exercises. Hence, the knowledge area indicated by related exercises with low scores should be further enhanced for the student. Instead of providing the student with more exercises with which he/she has been familiar, indicated by a higher score, the student should be recommended more exercises with lower scores to improve the student's performances in her/his weak areas. This in turn should bring improvement in the student's performances in the weak areas. The inclusion of the causal relationships also helps enhance the learning efficiency for the student by eliminating repeated feeding of those exercises which the student may have already mastered.

In most circumstances, recommendations are usually made based on "taking the top N highest-ranked items" from a sorted list, or the Top-N rule. Hence, the ranked scores from the network are converted to a descending order by the loss rate ($r_f$) defined as

$$r_f = 1 - \frac{g}{g_a}, \tag{1}$$

where $g$ denotes the actual score for an exercise obtained by the student, and $g_a$ denotes the perfect score allocated to the exercise. If the student's actual score is the same as the perfect score for an exercise, i.e., $g/g_a = 1$, its loss rate is zero, or the smallest. If the student's actual score is zero, indicating the attempt being completely incorrect, with $g/g_a = 0$, the loss rate is 1, or the highest. Thus, more exercises similar to that type of exercise should be recommended to the student.

The following subsections present the five parts of CDL, respectively.

### 2.2. Exercise encoding

To illustrate our model framework more clearly, we first introduce some definitions.

- **Knowledge point.** A knowledge point is the smallest unit of knowledge in a given discipline. In linear functions, for example, $k_1$ represents the slope, while $k_2$ represents the intercept, and so on. These smallest knowledge units are numbered sequentially to form a set containing all knowledge points in the subject. In symbolic terms, this means the following: $\mathrm{K} = \{k_1, k_2, \cdots, k_N\}$. Each of these knowledge points is indexed by numbering the points in the order in which they appear in the discipline.
- **Exercise.** As shown in Table 1, each column represents a knowledge point, and each row represents an exercise. The knowledge points contained in the exercise will be set to 1 in the corresponding position, and other knowledge points not contained in the exercise will be set to 0 in the corresponding position. In this way, each row vector in Table 1 is a representation of an exercise.
- **Knowledge graph and path.** For each knowledge point, we look for other knowledge points that co-occur with it in the same exercise in order to generate a knowledge graph. For example, for a particular knowledge point $k_2$, after traversing all the exercises, we find that it co-occurs 18 times with $k_4$, 12 times with $k_7$, 10 times with $k_9$, and 5 times with $k_{11}$. The knowledge graph for $k_2$ in terms of triples is then as follows: ($k_2$, 18/(18+12+10+5), $k_4$), ($k_2$, 12/(18+12+10+5), $k_7$), ($k_2$, 10/(18+12+10+5), $k_9$), ($k_2$, 5/(18+12+10+5), $k_{11}$). In this order, we can generate its knowledge graph for all knowledge points. We know that $k_2$ has four first-order (direct) neighbors, $k_4$, $k_7$, $k_9$ and $k_{11}$. Next, this process is propagated to the leveled neighbors of $k_2$, respectively, i.e., looking

for the neighbors of the first-order neighbors of $k_4$, $k_7$, $k_9$ and $k_{11}$, or the second-order neighbors of $k_2$ (neighbors of neighbors). This cycle is repeated to generate a knowledge path centered at $k_2$.

The above process Is performed once for each knowledge point to generate knowledge paths for all knowledge points. In each round of computation with the model, the knowledge paths of all knowledge points in a particular exercise (determined by the bottom branch) are fed into the intermediate branches to participate in training and learning, i.e., Knowledge Path 1, Knowledge Path 2, ..., Knowledge Path *n* in Figure 1.

**Table 1.** Exercise-knowledge matrix

| Exercise | Knowledge | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $k_1$ | $k_2$ | $k_3$ | $k_4$ | $k_5$ | $k_6$ | $k_7$ | $k_8$ | $k_9$ | $k_{10}$ | … | $k_N$ |
| $\boldsymbol{E_1}$ | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 |
| $\boldsymbol{E_2}$ | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 |
| ... | | | | | | | | | | | | |
| $\boldsymbol{E_i}$ | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | | 0 |

- **Path length.** To facilitate the application of knowledge paths in Figure 1, we standardize the lengths of the paths in our experiments. The path length is 1 if all knowledge points are kept to first-order neighbors only and 2 if all knowledge points are kept to second-order neighbors. The path length is a hyperparameter of our model that can be set prior to the experiment.

### 2.3. Student representation

As a student's ID does not reflect the student's learning ability, it is not sufficient to use student IDs alone to characterize individual students. Therefore, in the CDL model in Figure 1, we integrate a particular student's mastery of different knowledge paths into the representation of that student. This is done as follows.

First, we encode a student into a one-hot vector indexed by the student's ID. For example, student number 1 is represented as (1,0,0,0,...), with student number 2 as (0,1,0,0,...), student number 3 as (0,0,1,0,...) and so on. Such encoding, while able to identify different students, is too sparse (i.e., too many zeros involved in computations) and wastes storage space. Thus, immediately afterwards, the one-hot neurons, which are the input on the student side, are fed into a fully-connected network, i.e., the embedding operation in the top branch in Figure 1. This process maps the student from the one-hot vector to a low-dimensional, dense and real-value vector, shown as the *Es* layer in Figure 1.

Second, as already mentioned above, the *Es* is derived from the student ID alone and can only identify the student, and it does not contain semantic information describing the learning ability of that student. Therefore, based on the *Es*, the model learns the local knowledge paths for the different knowledge points mentioned in Subsection 2.2. Through the learning mechanism, the model can incorporate the knowledge paths with the highest failure rates for this student into *Es* with larger weighted probabilities, while other knowledge paths with lower failure rates (points where the student has a better grasp) are ignored. Through the operation of such a causal intervention, the resulting student representation *Cs* already contains the reasons for some of the student's failures in learning, i.e., a portrayal of the extent to which that student can learn different knowledge points.

### 2.4. Exercise representation

In a subject, different exercises contain different knowledge points. In order to represent each exercise in a structured way, we first number all the knowledge points in the subject. Assuming that there are *N* knowledge points in total, we number these knowledge points 1 to *N* in the order in which they are learned.

With the encoding of all knowledge points 1 to *N*, each exercise can be characterized as an *N*-dimensional vector. For example, if a specific exercise contains only knowledge points #2 and #5, its representation is (0, 1, 0, 0, 1, 0, …, 0). That is, the second and fifth elements of this *N*-dimensional vector are set to 1, and all other positions are set to 0. Another example is that if a specific exercise contains only knowledge points 1, 3 and 7, it is represented as (1, 0, 1, 0, 0, 0, 1, 0, …, 0). That is, the first, third and seventh elements of this *N*-dimensional vector are set to 1, and all other positions are set to 0. Following this way, all the exercises are represented as *N*-dimensional multi-hot vectors, which form part of the input to the exercises in Figure 1.

Similar to the student one-hot representation in Subsection 2.3, the multi-hot vector is still high-dimensional, sparse and wasteful of storage space. Thus, the multi-hot neurons that are the input on the exercise side are fed into a fully-connected network, shown as the embedding operation in the bottom branch in Figure 1. This process maps the particular exercise from a multi-hot vector to a low-dimensional, dense and real-value vector as the *Ee* layer in Figure 1.

Since *Ee* is derived only from the statistics of the knowledge points in a given exercise and does not contain semantic information describing the context of each knowledge point, the degree of difficulty of the exercise, being contained in the local knowledge paths where the individual knowledge points of the exercise are included, and these must be incorporated with *Ee*. Hence, through the learning mechanism, the model will incorporate the knowledge paths that are more likely to be answered incorrectly by students into *Ee* by assigning greater weighted probabilities to indicate the level of difficulty for the exercise. Other knowledge paths that are likely to be answered correctly by students (knowledge points that are easy to master) are ignored. Through the operation of such a causal intervention, the resulting representation of the exercise, *Ce*, should contain the specific reasons why the exercise was answered incorrectly, and the subsequent recommendations will be more biased towards the knowledge paths where these reasons lie.

### 2.5. Exercise recommendation based on causal inference

In order to explore causal relationships, for each student-exercise pair, in addition to feeding the student one-hot representation and the exercise multi-hot representation into the model in Figure 1, we also feed the knowledge path of each knowledge point for the exercise into the intermediate branches of the model to learn and find the relationship between the knowledge points and students' scores. The results of the knowledge path finding are incorporated into the student branch and the exercise branch, respectively, to generate the final student representation (i.e., *Cs* in Figure 1) and exercise representation (i.e., *Ce* in Figure 1) so that each student's weaknesses in knowledge acquisition can be identified accurately, and the personalized exercise suggestions can be given accordingly.

Next, the loss rate can be calculated using the student representation *Cs* and the exercise representation *Ce*. In the previous studies, the interaction of multiple tasks is achieved by the dot product. It usually indicates the degree of match or similarity between the two vectors. If two vectors are remarkably similar, their dot product would be large; on the contrary, the product would be small. However, this similarity-based approach fails to satisfy the triangular inequality in the dot product

model. To address this shortcoming and to satisfy the objectives of this task, we use the Euclidean distance instead of the dot product to measure students' interactions with exercises.

This idea of characterizing a student's loss rate score for an exercise by calculating the Euclidean distance of the two relevant vectors is in line with the intuition of our task. The goal of the task is to predict the student's failure rate on an exercise, which itself can be seen as a "distance" of the student to the exercise. If a student's mastery of an exercise is poor, there is a large distance between the student's feature vector and the exercise feature vector, or there is a significant difference or mismatch between the student vector and the exercise vector. Therefore, a large distance between *Cs* and *Ce* indicates that the student's ability is not yet up to the difficulty level of the exercise, and the model would predict a higher loss rate score for the student on the exercise. The student is then recommended similar exercises by means of the Top-N rule chosen from the highest to the lowest loss rate scores.

The specific implementation method is shown in the interaction layer in Figure 1. *Cs* and *Ce* have been mapped to a space of the same dimension in the embedding layer. They have the same length to facilitate the element-wise subtraction calculation later. We leverage the element-by-element subtraction of two vectors, *Cs* and *Ce*, to model the "distance" interaction between the student and the exercise.

Some nonlinear hidden layers are stacked on the input layer to form a deep structure to utilize its strong ability to explore the potential nonlinear relationships during learning. With increases in complexity for advanced subjects with more knowledge points and knowledge paths, the nonlinear hidden layer could help maintain the huge amount of semantic information and relationships between students and exercises during the entire process.

Our goal is to predict the probability value, which is essentially a classification task rather than a regression task. Therefore, cross-entropy is selected as the loss function.

### 2.6. Data normalization

First, the scores of exercises need to be normalized. In the original dataset, what was recorded is the student's real score on an exercise, which is not appropriate for direct use as the label of the model. For example, a student's score for exercise A is 6, and that for exercise B is 4. On the surface, the score for B is low. It seems that the model should recommend more exercises similar to B. However, if the total score for A is 12, and that for B is 5, the scoring rate for A is 6/12 (50%), and that for B is 4/5 (80%). This shows that the student's mastery of A is not as good as that of B, and more exercises similar to A should be recommended to the student. Therefore, normalization should be applied to all recorded scores against the exercises before feeding them to the system for recommendations.

## 3. Experiments

### 3.1. Datasets

#### *3.1.1. Self-built dataset for the Preliminary Advanced Mathematics (PAM) database*

Since most public databases do not contain the calibrated exercise knowledge points, to verify the effectiveness of our method, the Preliminary Advanced Mathematics (PAM) database, a database based on the self-developed online learning system containing the exercises for the advanced mathematics course for the preparatory students at the Northeast Normal University of China, was

selected for our experiments. The basic information of the PAM database is summarized in Table 2 and is described as follows.

**Table 2.** Summary of the PAM database

| Types of PAM exercises | | | |
|---|---|---|---|
| Multiple choice | Judgement | Filling the blank | Calculation |
| 917 | 326 | 384 | 591 |

- There are 2218 exercises in the PAM database, and all the exercises include 368 knowledge points that students need to master from four types: multiple choice, judgment, filling in the blank, and calculation. Each exercise contains 1 to 6 knowledge points.
- The records in the PAM database contain 1264 answers to the exercises in the courses students attempted in the recent three years.

#### *3.1.2. The ACM Knowledge Discovery and Data Mining (KDD) Cup*

Algebra 2005-2006 is one of the datasets used by the KDD Cup for educational data mining and from the Bridge to Algebra online learning platform. It is often used to test the knowledge tracking algorithm. Data of the KDD Cup can be downloaded through http://pslcdatashop.web.cmu.edu/KDDCup. It contains data for 575 students and 437 exercises, with 809694 interactive records. Each record includes 19 fields, such as student number, exercise type, knowledge points and answer results. Due to the large number of datasets, some being incomplete, we only selected 3000 interactive records from 300 students, covering 437 knowledge points and 1085 exercises (Table 3).

**Table 3.** Summary of the PAM and Algebra 2005-2006 datasets for experiments

| Dataset | Number of students | Number of exercises | Knowledge concepts | Records |
|---|---|---|---|---|
| PAM | 450 | 2218 | 368 | 1264 |
| Algebra 2005-2006 | 300 | 1085 | 437 | 3000 |

### 3.2. Experimental setup

The data preparation goes through the following processes: data preprocessing, score normalization for all exercises and calculation of the loss rate of each exercise for individual students. The standardized records are split randomly into two lots, 80% for training and 20% for testing. Cross validation is conducted 10 times, and the average value is taken as the final result. The program runs on a GeForce GTX1080 GPU.

### 3.3. Baselines

For exercise recommendation, the collaborative filtering algorithm, deep leaning based algorithms, and knowledge graph based algorithms are currently in use. Therefore, the following algorithms based on these three types are chosen as the baselines to compare with our proposed CDL method.

- User-CF [23]: This model is a collaborative filtering (CF) algorithm [24]. It is used to estimate the similarity of students based on their scores using the similarity formula, which leads to recommending similar exercises to students at similar scoring levels.
- KS-CF [25]: KS-CF is also a collaborative filtering algorithm, but it calculates the student's knowledge level matrix using the similarity formula, from which exercises are recommended to students at a similar knowledge level.
- DKT$^+$ [15]: DKT refers to deep knowledge tracing. It was used to estimate a student's mastery of knowledge. Exercises were recommended to the students according to the level of their mastery. DKT$^+$ adds a regulation term to the loss function of DKT to improve the accuracy and stability of the DKT algorithm.
- KGEB-CF [26]: KGEB-CF treats students, exercises and results as student entities, exercise entities and the cross-entity relationship. These three data items form the necessary vectors in the knowledge graph. The exercise recommendation is made by combining the knowledge graph correlation algorithm and the collaborative filtering.

## 4. Results and discussion

The experimental results are compared in two aspects: the root mean square error (RMSE) of all algorithms and the accuracy and recall of the model in the Top-N recommendation.

### 4.1. Comparison of RMSE

Table 4 shows the performances of CDL and four baselines for predicting the loss rate. A smaller RMSE corresponds to a more accurate prediction of the loss rate. The following facts can be drawn from the RMSE data in Table 4.

**Table 4.** RMSE and comparison

| Method | Algebra 2005-2006 | | PAM | |
|---|---|---|---|---|
| | RMSE | CDL improvement | RMSE | CDL improvement |
| User-CF | 0.8441 | 10.95% | 0.8718 | 14.44% |
| KS-CF | 0.8033 | 6.42% | 0.7989 | 6.63% |
| DKT$^+$ | 0.7892 | 4.75% | <u>0.7602</u> | 1.88% |
| KGEB-CF | <u>0.7768</u> | 3.23% | 0.7633 | 2.28% |
| CDL | 0.7617 | - | 0.7459 | - |
| Average improvement | 6.33% | | 6.31% | |

- In terms of RMSE, CDL is the best performer among the five methods on both PAM and Algebra 2005-2006. On PAM, the improvement of CDL by RMSE is from 3.23% over KGEB-CF to 10.59% over User-CF, with an average of 6.33%. On Algebra 2005-2006, the improvement of CDL by RMSE is from 1.88% over DKT$^+$ to 14.44% over User-CF, with an average of 6.31%. In both cases, User-CF is the worst performer because it is one of the earliest algorithms for exercise recommendation. Its variant KS-CF performed much better on both cases. KGEB-CF, by incorporating CF into the processing, performed even better than KGEB-CF on both cases. Hence, collaborative filtering (CF) alone seems an outdated method for exercise recommendation.

- DKT$^{+}$ seems the second best method behind CDL, but its performances seemed dependent on the database. It performed marginally below CDL on PAM but with an obvious margin below CDL on Algebra 2005-2006. Hence, we infer that the causal inference with deep learning combined into CDL should have made a positive contribution to the stable and superior performance of CDL over the four baseline methods.

### 4.2. Comparison on accuracy and recall

Tables 5 and 6 show the statistics of the performance of CDL and the baselines in ranking recommendations. In the Top-N recommendation, CDL is superior to other baselines in accuracy and recall. On PAM, the average improvements of CDL are 9.81%, 8.01%, 9.49% and 5.56% for P@5, P@10, R@5 and R@10, respectively. In Algebra 2005-2006, CDL also brought improvements for P@5, P@10, R@5 and R@10, but they were not as high as those on PAM.

**Table 5.** Comparison of precision and recall on PAM

| Method | PAM | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | P@5 | CDL improvement | P@10 | CDL improvement | R@5 | CDL improvement | R@10 | CDL improvement |
| User-CF | 0.493 | 15.82% | 0.481 | 11.43% | 0.049 | 14.29% | 0.079 | 10.13% |
| KS-CF | 0.514 | 11.09% | 0.496 | 8.06% | 0.049 | 14.29% | 0.081 | 7.41% |
| DKT$^{+}$ | 0.529 | 7.94% | 0.497 | 7.85% | 0.053 | 5.67% | 0.085 | 2.35% |
| KGEB-CF | 0.547 | 4.39% | 0.512 | 4.69% | 0.054 | 3.70% | 0.085 | 2.35% |
| CDL | **0.571** | - | **0.536** | - | **0.056** | - | **0.087** | - |
| Average improvement | 9.81% | | 8.01% | | 9.49% | | 5.56% | |

**Table 6.** Comparison of precision and recall on Algebra 2005-2006

| Method | Algebra 2005-2006 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | P@5 | CDL improvement | P@10 | CDL improvement | R@5 | CDL improvement | R@10 | CDL improvement |
| User-CF | 0.502 | 8.23% | 0.496 | 6.65% | 0.048 | 12.50% | 0.069 | 14.50% |
| KS-CF | 0.518 | 5.60% | 0.512 | 3.32% | 0.048 | 12.50% | 0.072 | 9.72% |
| DKT+ | 0.532 | 2.82% | 0.516 | 2.52% | 0.050 | 8.00% | 0.077 | 2.78% |
| KGEB-CF | 0.538 | 1.67% | 0.523 | 1.15% | 0.053 | 2.00% | 0.074 | 6.76% |
| CDL | **0.547** | - | **0.529** | - | **0.054** | - | **0.079** | - |
| Average improvement | 4.58% | | 3.41% | | 8.75% | | 8.44% | |

### 4.3. Comparison of performances of CDL with and without causal inference

In order to prove the effect of causal inference on the performance of CDL, experiments were carried out without including the causal inference in the algorithm. The results are shown in Table 7, along with the results from the full CDL with the inclusion of the causal inference. There was an observable difference between the algorithms with and without causal inference, and the full CDL

consistently outperformed the one without causal inference on both databases. Hence, the causal inference has made a positive contribution to the improvement of CDL over other existing methods.

**Table 7.** Comparison of performances of CDL with/without causal inference

| Dataset | Metric | Method | |
|---|---|---|---|
| | | CDL-Without-CI | CDL-CI(CDL) |
| PAM | P@5 ↑ | 0.572 | **0.582** |
| | P@10 ↑ | 0.507 | **0.545** |
| | R@5 ↑ | 0.052 | **0.058** |
| | R@10 ↑ | 0.078 | **0.091** |
| Algebra 2005-2006 | P@5 ↑ | 0.569 | **0.578** |
| | P@10 ↑ | 0.499 | **0.539** |
| | R@5 ↑ | 0.051 | **0.055** |
| | R@10 ↑ | 0.073 | **0.089** |

### 4.4. The influence of knowledge path in the causal intervention

By setting the length of the knowledge path to 3, 4, 5 and 6, the experimental results with CDL by RMSE are shown in Figure 3. If an exercise contains 3 to 5 knowledge points, CDL performed with a level of RMSE below 0.8. This indicates a relatively lower difficulty level for the subject. If an exercise contains 6 or more knowledge points, meaning a higher difficulty level for the subject that may require knowledge from multiple areas, the RMSE shows a tendency of exponential increase for CDL even still with a low value.

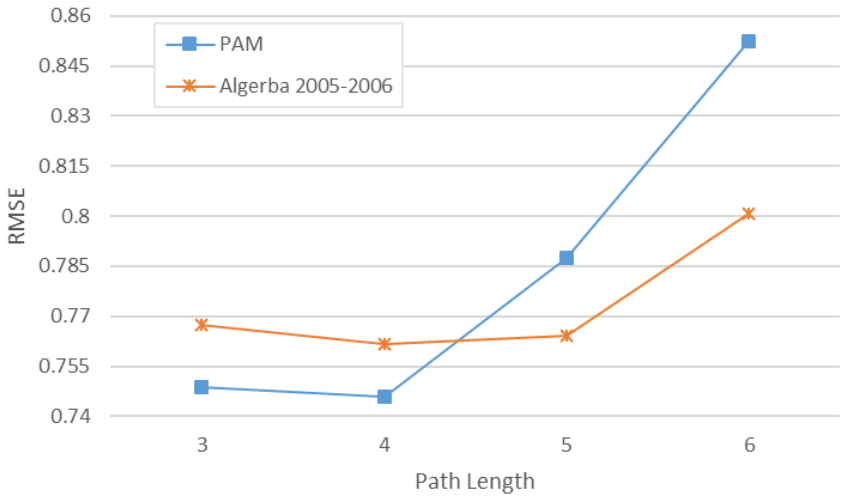



**Figure 3.** Influence of the length of the knowledge path

### 4.5. Influence of hyperparameters

This experiment is mainly about the influence of hyperparameters on the performance of CDL. As CDL uses a deep learning algorithm, the hyperparameters mainly include the dimension of embedding layers, epochs and the number of interaction layers.

### *4.5.1. Dimension of embedding*

The value range of the embedding dimension is 50 to 300, and the incremental step size is 50. The results are shown in Figure 4. A larger embedding size corresponded to a smaller RMSE. In other words, the embedded size is basically inversely proportional to RMSE. However, when the embedding size is around 230, the RMSE is the lowest for both datasets.

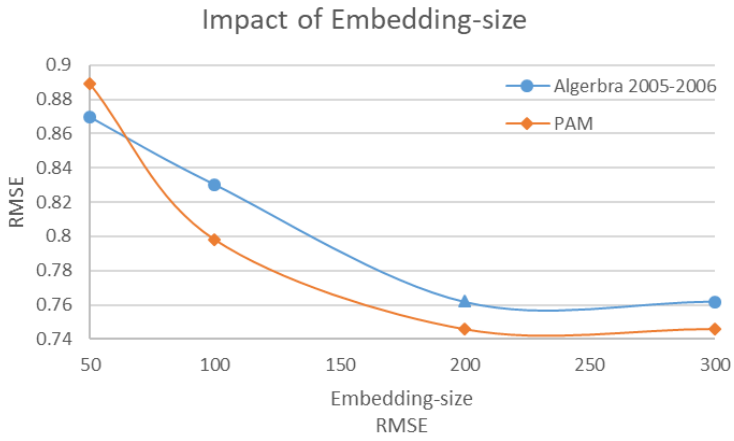



**Figure 4.** RMSE with different embedding sizes

### *4.5.2. Epochs*

The value range of the epochs is 0 to 300, and the step size is 50. The results are shown in Figure 5. On both datasets, the trend of RMSE is decreasing with the increase in the number of epochs.

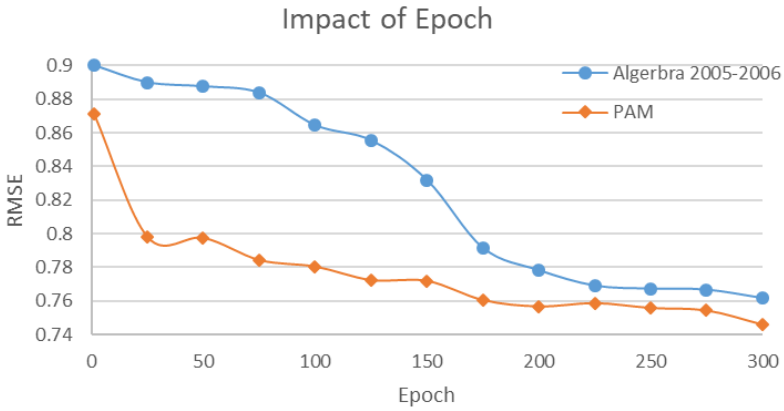
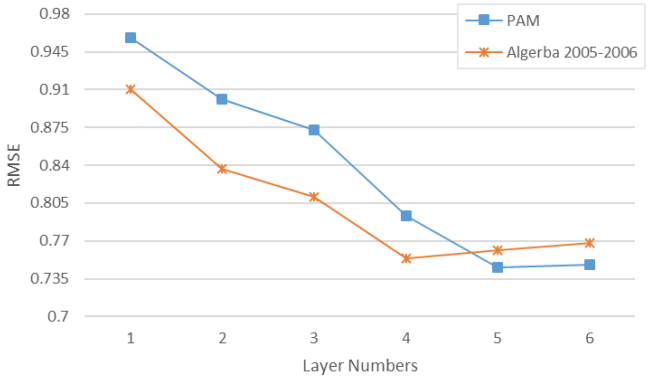



**Figure 5.** RMSE with different epochs

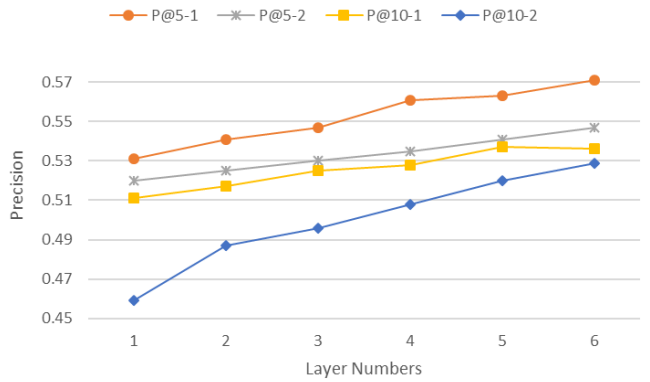
### *4.5.3. Number of interaction layers*

The value range of interaction layers is 1 to 6, and the step size is 1. In order to facilitate the representation in Figure 6, PAM is represented by 1, and Algebra 2005-2006 is represented by 2, so P@5-1 indicates that the data is based on PAM. The performance of CDL is shown in Figure 6.

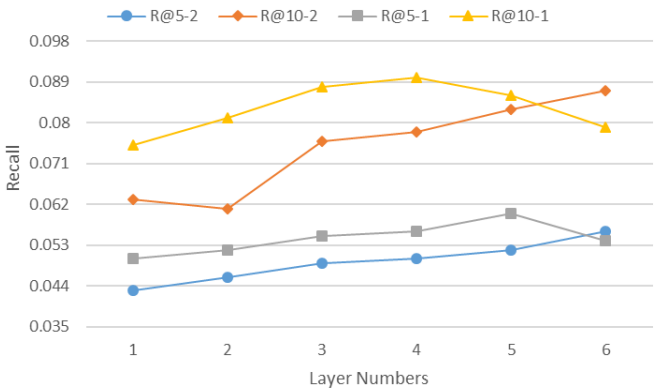
More layers corresponded to a lower RMSE (Figure 6a). However, this trend may invert after reaching a certain number of layers. For example, with Algebra 2005-2006, the RMSE increased from the lowest value, with 4 layers, once the number of layers increased.




(a) Effects of different layers on RMSE




(b) Effects of different layers on precision




(c) Effects of different layers on recall

**Figure 6.** Impact of interaction layers

Precision and recall are basically the same for both datasets. More layers corresponded to better performance (Figure 6b). However, for the recall, when the number of interaction layers reached 5, the performance seemed to peak on PAM. More layers over 5 may decrease the performance on recall on PAM (Figure 6c). This may be related to the relatively smaller size of the PAM dataset compared with Algebra 2005-2006.

## 6. Conclusions

We proposed a personalized exercise recommendation method, causal deep learning (CDL), based on the integration of causal inference and deep learning. The concept of CDL is to use students' records of attempted exercises and the score rates to study the impact of exercise knowledge points on students' achievements in a specific knowledge area, from which more relevant exercises can be recommended to the corresponding students who are likely to need improvement of skills in and understanding of the subject areas indicated by the chosen exercises. In this model, we first use deep learning to train and generate initial feature representations for the students and the exercises. Causal inference intervention algorithms are then used to fine tune these feature representations. The second time, deep learning is used to predict individual students' score ratings on exercises, from which the Top-N ranked exercises are recommended to corresponding students to achieve targeted improvement. Experiments of CDL and four baseline methods on two real-world datasets were conducted to validate the performance of CDL. In terms of RMSE, precision and recall rate, CDL consistently outperformed the existing methods in all measures on the two datasets.

Although the inclusion of causal inference has brought observable improvement in exercise recommendations by CDL, it should be able to make more significant contributions to the improvement in exercise recommendation by better capture of the casual relationships. In future projects, we would like to continue our efforts to better capture the causal inference for deep learning.

## Acknowledgments

This research was supported by the Jilin Provincial Development and Reform Commission (No. 2022C046-5) Research on Higher Education Teaching Reform in Jilin Province in 2020 and the Project Library of Changchun Humanities and Sciences College in 2022. We are grateful to the reviewers for their critical and constructive feedback on the original manuscript. We also appreciate the guidance and assistance provided by Prof. Jun Shen and Prof. William Guo during the revision. All these improved this article significantly.

## References

1. Vie, J. and Kashima, H., Knowledge tracing machines: Factorization machines for knowledge tracing. *The Thirty-Third AAAI Conference on Artificial Intelligence*, 2019, 33(01): 750–757. https://doi.org/10.1609/aaai.v33i01.3301750
2. Walker, A., Recker, M. and Lawless, K., Collaborative information filtering: A review and an educational application. *International Journal of Artificial Intelligence in Education*, 2004, 14(1): 3–28.
3. Hsu, M., A personalized English learning recommender system for ESL students. *Expert Systems with Applications*, 2008, 34(1): 683–688. https://doi.org/10.1016/j.eswa.2006.10.004
4. Milicevic, A., Vesin, B. and Ivanovic, M., E-learning personalization based on hybrid recommendation strategy and learning style identification. *Computers & Education*, 2011, 56(3): 885–899. https://doi.org/10.1016/j.compedu.2010.11.001
5. Segal, A., Katzir, Z. and Shapira, B., EduRank: A collaborative filtering approach to personalization in e-learning. *Proceedings of the 7th International Conference on Educational Data Mining*, 2014, 68–75.

6. Toledo, R. and Mota, Y., An e-learning collaborative filtering approach to suggest problems to solve in programming online judges. *International Journal of Distance Education Technologies*, 2014, 12(2): 51–65. https://doi.org/10.4018/ijdet.2014040103
7. Wu, D., Lu, J. and Zhang G., A fuzzy tree matching-based personalized e-learning recommender system. *IEEE Transactions on Fuzzy Systems*, 2015, 23(6): 2412–2426. https://doi.org/10.1109/TFUZZ.2015.2426201
8. Dwivedi, P. and Bharadwaj, K., Effective trust-aware e-learning recommender system based on learning styles and knowledge levels. *Journal of Educational Technology & Society*, 2013, 16(4): 201–216.
9. Jiang, C., Feng, J. and Sun, X., Personalized exercises recommendation algorithm based on knowledge hierarchical graph, ReKHG. *Computer Engineering and Applications*, 2018, 54(10): 234–240.
10. Gong, T. and Yao, X., Deep exercise recommendation model. *International Journal of Modeling and Optimization*, 2019, 9(1): 18–23. https://doi.org/10.7763/IJMO.2019.V9.677
11. Piech, C., Bassen, J. and Huang, J., Deep knowledge tracing. *NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems*, 2015, 505–513.
12. Zhang, L., Xiong, X. and Zhao, S., Incorporating rich features into deep knowledge tracing. *Proceedings of the Fourth ACM Conference on Learning at Scale*, 2017, 169–172. https://doi.org/10.1145/3051457.3053976
13. Su, Y., Liu, Q. and Liu, Q., Exercise-enhanced sequential modeling for student performance prediction. *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI-18)*, 2018, 2435–2443.
14. Wang, L., Angela, S. and Liu L., Deep knowledge tracing on programming exercises. *Proceedings of the Fourth*. *ACM Conference on Learning at Scale*, 2017, 201–204. https://doi.org/10.1145/3051457.3053985
15. Yeung, C. and Yeung, D., Addressing two problems in deep knowledge tracing via prediction-consistent regularization. *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*, 2018, 41–50. https://doi.org/10.1145/3231644.3231647
16. Didelez, V. and Pigeot, I., Judea Pearl: Causality: Models, reasoning, and inference. *Politische Vierteljahresschrift*, 2001, 42(2): 313–315. https://doi.org/10.1007/s11615-001-0048-3
17. Louizos, C., Shalit, U. and Mooij, J., Causal effect inference with deep latent-variable models. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, 6449–6459.
18. Van, D., Causal reasoning and inference making in judging the importance of story statements. *Child Development*, 1989, 60(2): 286–297.
19. Joachims, T., Swaminathan, A. and De, R., Deep learning with logged bandit feedback. *International Conference on Learning Representations*, 2018, 1–12.
20. Swaminathan, A. and Joachims, T., Counterfactual risk minimization: Learning from logged bandit feedback. *Proceedings of the 32nd International Conference on Machine Learning*, 2015, 814–823. https://doi.org/10.1145/2740908.2742564
21. Swaminathan, A. and Joachims, T., The self-normalized estimator for counterfactual learning. *Advances in Neural Information Processing Systems*, 2015, 3231–3239.
22. Lisa, W., Angela, S., Larry. L. and Chris, P., Deep knowledge tracing on programming exercises. *Proceedings of the Fourth ACM Conference on Learning*, 2017, 201–204. https://doi.org/10.1145/3051457.3053985

23. Guy, S. and Asela, G., Evaluating recommendation systems. *Recommender Systems Handbook*, 2011, 257–297. https://doi.org/10.1007/978-0-387-85820-3_8
24. Gang, L. and Tianyong, H., User-based question recommendation for question answering system. *International Journal of Information and Education Technology*, 2012, 2(3): 243–246. https://doi.org/10.7763/IJIET.2012.V2.120
25. Shah, K., Zafar, A. and Irfan, U., Recommender systems: Issues, challenges, and research opportunities. *Information science and applications (ICISA) 2016*, 2016, 1179–1189. https://doi.org/10.1007/978-981-10-0557-2_112
26. Ming, Z., De-sheng, Z., Ran, T., You-Qun, S., Xiang, Y. and Qian, W., Top-N collaborative filtering recommendation algorithm based on knowledge graph embedding. *Proceedings of the 14th International Conference of the Knowledge Management in Organizations*, 2019, 122–134. https://doi.org/10.1007/978-3-030-21451-7_11

**Author's biography**

**Dr. Suhua Wang** is an associate professor of Changchun Humanities and Sciences College. She specializes in deep learning recommendation system algorithm improvement. Her research interest is how to combine recommendation algorithms with teaching methods to improve students' learning efficiency.

**Dr. Zhiqiang Ma** is a professor of Changchun Humanities and Sciences College. He is specialized in artificial intelligence. His research interests include deep learning and STEM education.

**Hongjie Ji** received the B.S. degree in computer science and technology from the Northeast Normal University, Changchun, China, in 2020. She is currently pursuing the master's degree with Northeast Normal University, Changchun. Her current research interests include recommender systems and machine learning.

**Tong Liu** is an undergraduate at Northeast Normal University, Changchun, China. Her major is computer science and technology. Her research interests are natural language processing and recommendation systems.

**Anqi Chen** is studying for a bachelor's degree in the School of Information Science and Technology of Northeast Normal University. She is now majoring in computer science and technology. Her main research interests are artificial intelligence and recommendation systems.

**Dawei Zhao** is an associate professor of Changchun Humanities and Sciences College. He is specialized in online education. His research interests include online education platforms and educational psychology.