Peer-inspired Student Performance Prediction in Interactive Online Question Pools with Graph Neural Network

Haotian Li, Huan Wei, Yong Wang, Yangqiu Song, Huamin Qu
Department of Computer Science and Engineering, HKUST, Hong Kong SAR, China
{hlibg,hweiad,ywangct}@connect.ust.hk;{yqsong,huamin}@cse.ust.hk

ABSTRACT

Student performance prediction is critical to online education. It can benefit many downstream tasks on online learning platforms, such as estimating dropout rates, facilitating strategic intervention, and enabling adaptive online learning. *Interactive online question pools* provide students with interesting interactive questions to practice their knowledge in online education. However, little research has been done on student performance prediction in interactive online question pools. Existing work on student performance prediction targets at online learning platforms with predefined course curriculum and accurate knowledge labels like MOOC platforms, but they are not able to fully model knowledge evolution of students in interactive online question pools. In this paper, we propose a novel approach using Graph Neural Networks (GNNs) to achieve better student performance prediction in interactive online question pools. Specifically, we model the relationship between students and questions using student interactions to construct the *student-interaction-question* network and further present a new GNN model, called R^2GCN, which intrinsically works for the heterogeneous networks, to achieve generalizable student performance prediction in interactive online question pools. We evaluate the effectiveness of our approach on a real-world dataset consisting of 104,113 mouse trajectories generated in the problem-solving process of over 4,000 students on 1,631 questions. The experiment results show that our approach can achieve a much higher accuracy of student performance prediction than both traditional machine learning approaches and GNN models.

KEYWORDS

Student performance prediction, graph neural networks, online question pools

ACM Reference Format:
Haotian Li, Huan Wei, Yong Wang, Yangqiu Song, Huamin Qu. 2020. Peer-inspired Student Performance Prediction in Interactive Online Question Pools with Graph Neural Network. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3340531.3412733

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412733

1 INTRODUCTION

With the rapid growth of online education in the past few years, various online learning platforms (e.g., interactive online question pools) have become increasingly popular. Student performance prediction, which aims to predict students' future grades in the assignments and exams [7], is of significant importance for online education. For example, with an accurate student performance prediction, the online learning platforms can better estimate the course dropout rate and take appropriate measures to further increase the student retention rate [17]. The course instructors can recommend suitable learning materials to different students [20]. Extensive research has been conducted on student performance prediction on online learning platforms, which mainly consists of static models and sequential models [7]. Static models consider the static information of students (e.g., historical scores and learning activities) to predict their future performance [8, 11, 17], where the underlying relationship between different learning materials (e.g., courses, videos, questions) are totally ignored. Sequential models, such as Deep Knowledge Tracing (DKT) [16] and its variant methods [1, 14], further capture the sequential relationship of the learning materials. However, the sequential models are mainly applied to MOOC platforms and intrinsically rely on accurate labeling of the tested knowledge for each question.

Interactive online question pools, an essential part of online learning, attempts to make it a joyful process for students to practice their knowledge on a collection of interactive questions. For instance, Math Playground[1], Learnlex[2], and LeetCode[3] enable students to practice their mathematics or programming skills. However, interactive online question pools are different from MOOC platforms and there is *no predefined sequential order* for the learning materials (i.e., questions). Students often need to freely choose which question to answer next. It makes the prior sequential models only able to partially model the student knowledge evolution. Also, an accurate label of the tested knowledge for each question is also not necessarily available in interactive online question pools [22]. Therefore, we are motivated by the crucial research question: *how can we achieve effective student performance prediction in interactive online question pools?*

In this paper, we propose a novel GNN-based approach to model student knowledge evolution and predict student performance in interactive online question pools. Specifically, we first build a heterogeneous large graph, consisting of questions, students, and the interactions between them, to extensively model the complex relationship among different students and questions. This is inspired by prior studies [9, 10] and they have shown that the academic

[1]https://www.mathplayground.com/
[2]https://learnlex.com/
[3]https://leetcode.com/

performance of a student is correlated with the performances of other students (i.e., *peers*), especially those students with similar learning behavioral patterns. Then, we further formalize student performance prediction as a semi-supervised node classification problem on this heterogeneous graph. The classification results are the student score levels (4 score levels in our experiment) on each question. Moreover, we propose a novel GNN model, *Residual Relational Graph Neural Network (R^2GCN)*, for student performance prediction in interactive online question pools. Its model architecture is adapted from Relational-GCN (R-GCN) [18] and further incorporates a residual connection to different convolutional layers and original features. We conduct detailed evaluations of our approach on a real-world dataset consisting of 104,113 mouse trajectories generated in the problem-solving process of over 4,000 students on 1,631 questions, which are collected by our industry collaborator Trumptech [4] from their K-12 interactive online math question pool *Learnlex*. The results show that our approach outperforms other methods in terms of both accuracy and weighted F1 score. Detailed insights and observations are also discussed. In summary, the major contributions are as follows:

- **Question Formulation.** We formulate the student performance prediction in interactive online question pools as a semi-supervised node classification problem on a large heterogeneous graph that captures the underlying relationship among questions and students. New mouse movement features are also introduced to better delineate student-question interactions.
- **Model Architecture.** We propose a new convolutional graph neural network model, R^2GCN, to achieve student performance prediction in interactive online question pools, which intrinsically works for heterogeneous networks.
- **Detailed Evaluations.** We conduct detailed evaluations of our approach on a real-world dataset. The results demonstrate its capability of achieving a better prediction accuracy than both traditional machine learning models (e.g., Logistic Regression (LR) and Gradient Boosted Decision Tree (GBDT) model), and R-GCN [18].

2 RELATED WORK

The related work of this paper can be categorized into two groups: Graph Neural Networks and student performance prediction.

Graph Neural Networks. Graph Neural Networks (GNNs) are the deep neural networks adapted from the widely-used Convolutional Neural Networks (CNNs) and specifically designed for graphs. They have shown powerful capability in dealing with complicated relationships in a graph. Representative methods of GNNs include Graph Convolutional Network (GCN) [12], GraphSAGE [6], R-GCN [18], Message Passing Neural Network (MPNN) [5], Gated Graph Neural Network (GGNN) [13], and Heterogeneous Graph Attention Network (HAN) [21]. Among them, R-GCN and HAN are specifically designed for heterogeneous graphs. MPNN and GGNN perform graph convolution on graphs with multi-dimensional edge features. However, little research has been conducted on heterogeneous graphs with multi-dimensional edge features.

GNNs have been applied in various applications, such as recommender systems [23], social networks analysis [15], and molecular property prediction [13]. Very few studies have been done in the field of online learning and education. A recent study on college education by Hu and Rangwala [7] proposed a GNN-based approach called Attention-based Graph Convolutional Network (AGCN) which utilizes a GCN to learn graph embedding of the network of frequently taken prior courses and then applies the attention mechanism to generate weighted embedding for final predicted grades. However, their method is limited to the graph with only one type of nodes (i.e., courses) and edges (i.e., the connection of courses taken in continuous 2 semesters), which cannot be applied to student performance prediction in interactive online question pools due to their intrinsically complex relationships among questions and students.

Student performance prediction. Student performance prediction is an important task in educational data mining. For example, it can contribute to recommending learning material [20] and improving student retention rates [17] in online learning platforms. According to the study by Hu and Rangwala [7], prior studies on student performance prediction mainly include static models and sequential models. Static models refer to traditional machine learning models such as GBDT [22], Supporting Vector Machine (SVM) [3], and LR [19], which make predictions on student performances based on the static patterns of student features. On the contrary, sequential models [1, 14, 16] are proposed to better capture the temporal evolutions in students' knowledge or the underlying relationship between learning materials. However, sequential models cannot be directly applied to student performance prediction in interactive online question pools. They rely on the accurate labeling of the questions' tested knowledge, which is not always available in online question pools. Also, these models do not distinguish different questions of the same knowledge label and will predict the same result for different questions with the same knowledge label.

A recent study [22] conducted student performance prediction in interactive online question pools by introducing new features based on student mouse movement interactions to delineate the similarity between questions. However, their approach implicitly requires that the questions must have similar question structure designs and involve drag-and-drop mouse interactions, which may not always hold. In this paper, we aim to propose a more general approach for student performance prediction in interactive online question pools that can work for question pools with several hundred or thousand questions of different types.

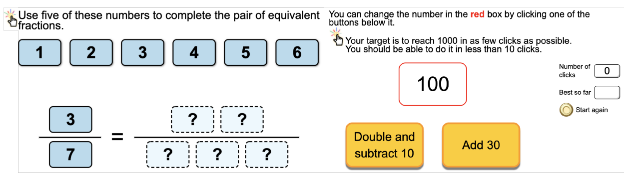
3 BACKGROUND

In this study, our data is collected from Learnlex, an interactive online question pool developed by Trumptech, a leading educational technology company in Hong Kong. This platform contains around 1,700 interactive math questions and has served more than 100,000 K-12 students during the last decade. Different from questions provided on MOOC platforms, these interactive questions could be freely browsed and answered by students without predefined orders and are merely assigned fuzzy labels, *grade*, *difficulty*, and *math dimension*. *Grade* indicates the targeted grade of students and ranges from 0 to 12. *Difficulty* is an index of five difficulty levels

[4] https://www.trumptech.com/en

(i.e., 1 to 5). *Math dimension* is a fuzzy math concept indicating the knowledge tested in that question.

Apart from these labels, the mouse movement interactions of students in their problem-solving process are also collected. According to our empirical observation, there are mainly two types of mouse movement interactions during students' problem-solving processes, i.e., *drag-and-drop* and *click*, as shown in Figure 1. Figure 1(a) is an example question that needs students to drag blue blocks on the top to appropriate locations to fulfill the requirement (*drag-and-drop*). The question in Figure 1(b) asks students to click the yellow buttons to complete a given task (*click*).



(a) (b)

Figure 1: Two examples of interactive questions on Learnlex.

When a student finishes a question, the platform will assign a discrete score between 0 and 100 to the submission. The possible scores of a question are often a fixed number of discrete values depending on what percentages a student can correctly answer the question, and the majority of the questions can have at most four possible score values. Therefore, we map the raw scores in historical score records to 4 score levels (0-3) to guarantee a consistent score labeling across questions. Also, only the score of a student's first trial on a question is considered in our experiment.

On this platform, we collected 2 parts of data, i.e., the historical score records and the mouse movement records. There are 973,676 entries from September 13, 2017 to January 7, 2020 in the historical score records, and each entry includes a score value, student ID, question ID, and the timestamp. The mouse movement records document the raw types of the mouse events (i.e., *mousemove, mouseup,* and *mousedown*), the corresponding timestamps, and positions of mouse events of all the students working on the interactive online question pool from April 12, 2019 to January 6, 2020. A *mouse trajectory* is a series of raw mouse events that are generated during a student's problem-solving process. In total, we collected 104,113 mouse trajectories made by 4,020 students on 1,617 questions.

4 THE PROPOSED METHOD

We propose a peer-inspired approach for student performance prediction in interactive online question pools. It extensively considers the historical problem-solving records of both a student and his/her peers (i.e., other students working on the question pool) to better model the complex relationship among students, questions, and student performances and further enhance student performance prediction, which is achieved by using a GNN-based model. Figure 2 shows the framework of our approach, which consists of three major modules: data processing & feature extraction, network construction, and prediction. The module of *data processing & feature extraction* is designed to process the related data and extract features of the historical data that will be further used for network construction and student performance prediction. We considered three types of features: *statistical features of students* reflecting students' past performance, *statistical features of questions* indicating the question popularity and their average scores, and *mouse movement features* representing the characteristics of students' problem-solving behaviors. The *network construction* module builds a network consisting of both students and questions, where the interactions between them are also considered and further integrated into the network. This network also incorporates the three types of features to extensively model the various performance of different students on different questions. Finally, the constructed network, along with the extracted features, is input into the *prediction* module, where we propose R^2GCN, a novel Residual Relational Graph Neural Network model that is adapted from R-GCN [18] by adding residual connections to hidden states, to predict a student's score level on the unattempted questions in interactive online question pools.

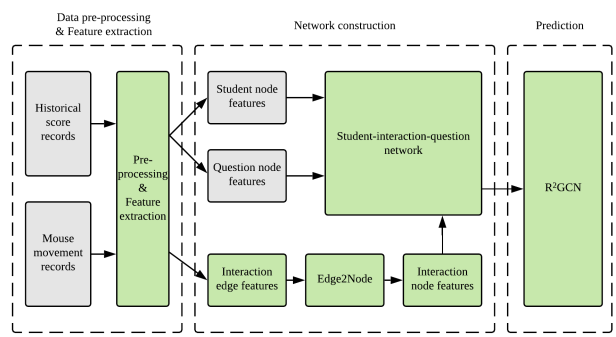


Figure 2: The framework of the proposed method. The blocks highlighted in green are our major contributions.

4.1 Feature Extraction

As discussed above, the feature extraction module mainly extracts three types of features: statistical features of students, statistical features of questions, and mouse movement features.

Statistical features in Tables 1 and 2 are extracted from historical score records. Statistical features of students mainly contain students' past performance on various types of questions to reflect the students' ability on a certain type of questions, for example, the average score of first trials on numeric questions of grade 8 and difficulty 3. Statistical features of questions are extracted to show the popularity and real difficulty level of them, for example, the proportion of trials getting score level 2 on the question.

Table 1: Statistical features of students.

Feature Name	Explanation	Example
# Total trials	Number of a student's total trials.	-
# 2nd trials	Number of a student's 2nd trials.	-
% Trials in [math dimension× grade×difficulty]	Percentage of trials on questions of certain math dimension, grade, and difficulty.	% Trials on spatial questions of grade 5 and difficulty 5.
Mean1stScore in [math dimension× grade×difficulty]	Mean score of 1st trials on questions of certain math dimension, grade, and difficulty.	Mean1stScore on numeric questions of grade 8 and difficulty 3.

For mouse movement features, we mainly consider two types of basic mouse movement interactions in interactive online question pools: *click* and *drag-and-drop*. However, despite their differences, both of them start with mouseup and end with mousemove, as

Table 2: Statistical features of questions. The star sign * indicates categorical features encoded by one-hot encoding.

Feature Name	Explanation	Example
Math dimension*	Question's related topic.	Numeric
Grade*	Grade of target students.	12
Difficulty*	Question's difficulty level.	4
# Total trials	Number of trials on a question.	-
# 2nd trials	Number of 2nd trials on a question.	-
% Trials in [score level]	Percentage of trials in each score level.	% Trials in 2.

Table 3: New mouse movement features based on GCs and mouse movement timestamps. The star sign * indicates categorical features encoded by one-hot encoding.

Feature Name	Explanation
1stGCTimeLength	Time length between entering the question and the 1st GC.
1stGCTimePercent	Percentage of the duration of the 1st GC.
1stGCEventStartIdx	Number of mouse events before the 1st GC.
1stGCEventPercent	Percentage of mouse events before the 1st GC among all.
1stGCEventEndIdx	Number of mouse events when the 1st GC ends.
GCCount	Total number of GCs.
GCPerSecond	Average number of GCs per second.
AvgTimeBtwGC	Average time between GC.
MedTimeBtwGC	Median value of time between GCs.
StdTimeBtwGC	Standard deviation value of time between GCs.
OverallDistance	Total mouse trajectory length.
InteractionHour*	Time point when students solves the problem, e.g., 13:00.

shown in Figure 3(d). Thus, they are considered as *generalized clicks (GCs)* in this paper. We analyze the GCs in students' mouse trajectories in their problem-solving process and further propose a set of new mouse movement features, as shown in Table 3. These features are mainly designed to reflect the first GC made by students when they try to answer the question. First GCs can reveal the information of questions, for example, the required type of mouse movement interaction. They also reflect the problem-solving behaviors of students, for example, read the description first or try to play with the question first. Apart from these new mouse movement features, some representative features regarding *think time* introduced in prior studies [22] are also extracted in this paper to comprehensively delineate students' learning behaviors, for example, the length of thinking time before answering the question.

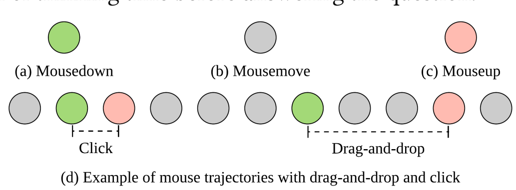


Figure 3: An illustration of the mouse interactions *click* and *drag-and-drop*. (a)-(c) indicate raw mouse events.

4.2 Network Construction

It is challenging to model the relationship between questions in an interactive online question pool, since there is no curriculum or predefined question order that every student need to follow and can help model the relationship among questions. We propose using students' mouse movement interactions with the attempted questions as a bridge to construct the dependency relationship between different questions and build a problem-solving network. When conducting performance prediction for a student, the mouse movement records of his/her peers (i.e., other students) are all considered. Figure 4(a)-(b) illustrates the problem-solving network is a heterogeneous network composed of students nodes (S), questions nodes (Q), and interaction edges with multi-dimensional mouse movement features mentioned above.

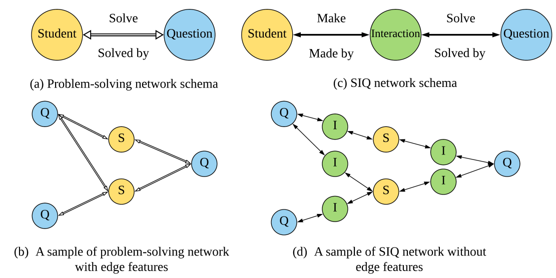


Figure 4: The problem-solving network and SIQ network.

Inspired by the recent progress of GNNs [24], we propose using GNNs to model the relationship among questions, students, and the interactions between them, which forms intrinsically a heterogeneous network with multi-dimensional edge features. However, there are no GNN models designed for such kind of heterogeneous networks. Inspired by the method of breaking up edges and transform a homogeneous network to a bipartite network [24], we conduct a transformation named *Edge2Node* to transform the mouse movement interaction (i.e., edges) between students and questions into "fake nodes", and further build a Student-Interaction-Question (*SIQ*) Network to model the complex relationships among different questions and students, as shown in Figure 4(d). *SIQ* network is the basis of applying our GNN-based approach to student performance prediction in interactive online question pools.

4.3 Residual Relational Graph Neural Network

To model the relationship of questions, students, and interactions, we construct a heterogeneous *SIQ* network to feed into GNN models. R-GCN [18] is one of the most widely used models to perform message passing on heterogeneous networks due to its good scalability and excellent performance. However, it does not fully make use of the hidden states. In GNNs, hidden states could be considered as the message aggregation results of near neighbors and R-GCN directly transforms them to next hidden states, which leads to a possible information loss.

One of the popular methods to handle this issue is to add residual connections between different level of hidden states and such an approach has been successfully applied to enhancing the performance of GCN [4]. Also, prior research [2] has also shown that integrating linear transformation of original simple features to the output layer can help improve the performance of deep models. Therefore, we propose a new model structure, Residual Relational Graph Convolutional Network (R^2GCN) to enhance traditional R-GCN [18] structure by adding residual connections between different hidden layers and also integrating the original statistical features of questions into the model.

Figure 5 shows the framework of the proposed model, R^2GCN. It consists of parallel input layers for feature transformation of different types of nodes to the same shape, consequential R-GCN layers for message passing, residual connections to hidden states and original features for capturing different levels of information and the output layer for final prediction. We will introduce the structure of our model from the perspectives of several key functions, message and update function in the message passing phase and readout function in the prediction phase. In the input network, n, p, and R represents the number of node types, the target type of nodes, and all relation types respectively. Since the *SIQ* network contains three types of nodes, n is 3 in our experiments.

Message function. The message function transmits and aggregates messages from all neighbor nodes $\mathcal{N}_i$ to center node i in the message passing phase. In each R-GCN layer, the received message $M_i^{(l+1)}$ of node i in layer $l+1$ is defined as

$$M_i^{(l+1)} = \sum_{r \in R} \sum_{j \in \mathcal{N}_i^r} w_{i,j} W_r^{(l)} h_j^{(l)}, \tag{1}$$

where W_r is the weight matrix of relation r belonging to all relations R and $h_j^{(l)}$ is the hidden state of node j after layer l, and $w_{i,j}$ indicates the weight of the message from node j. Here we use average function to reduce the messages transmitting on the same type of edges and sum function to reduce messages of different types of edges. $w_{i,j}$ is set as the multiplicative inverse of the number of nodes in $\mathcal{N}_i^r$ in this paper.

Update function. The update function updates the center node i's hidden state $h_i^{(l)}$ after layer l with the message $M_i^{(l+1)}$ generated by Equation (1) in the message passing phase. In our model, to preserve the original hidden state of the center node i, the update function is defined as

$$h_i^{(l+1)} = \sigma(M_i^{(l+1)} + W_0^{(l)} h_i^{(l)} + b), \tag{2}$$

where W_0 denotes the weight matrix of center node i's hidden state, b denotes the bias and σ is the activation function.

Readout function. The readout function transforms the final hidden state to the prediction result. Different from the original R-GCN model, the readout function of R^{2}GCN adds residual connections to both hidden states and original features. The readout function for node p_i of type p is defined as

$$\hat{y_{p_i}} = f(Concat(F_{p_i}, h_{p_i}^0, h_{p_i}^1, ..., h_{p_i}^k)), \tag{3}$$

where $\hat{y_{p_i}}$ is the predicted result, f is the function to transform the concatenated hidden states to final result, F_{p_i} denotes the original input features of the corresponding question node and k represents the number of consequential R-GCN layers.

5 EXPERIMENT

We conducted experiments to evaluate the prediction accuracy and weighted F1 score of students' performance on each question and there are 4 question score levels. We compared the proposed approach with 5 baseline approaches including 3 classical machine learning models without peers' mouse movement features.

5.1 Data Processing

In our experiment, we first extract a portion of the original dataset with records from April 12, 2019 to June 27, 2019 (denoted as ***short-term dataset***) for extensively evaluating the proposed approach in terms of prediction accuracy, the influence of labeled dataset size and the influence of the topological distance between questions in training, validation, and test set. In short-term dataset, there are 43,274 mouse trajectories made by 3,008 students on 1,611 questions.

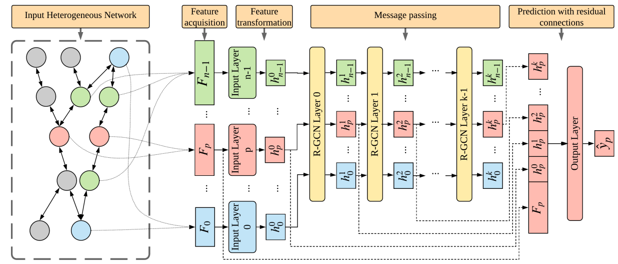


Figure 5: The framework of our proposed R^{2}GCN model. The input heterogeneous network contains n types of nodes and nodes of type p is our target for prediction. The model has k R-GCN layers. All rounded rectangles represent various layers and other rectangles represent 2D tensors. Dash lines represent residual connections. Each color represents a type of nodes and grey denotes other types of nodes. F represents features of nodes. h represents hidden states.

Taking into account that too few labeled data will make it difficult to train the GNN models, we only conducted experiments for the students who have finished at least 70 questions. Therefore, we gained 47 students in total, which are tested in our experiments for the short-term dataset. Also, we further use all the records from April 12, 2019 to January 6, 2020 (denoted as ***long-term dataset***) to further evaluate the performance of our approach. We extend the range of filtered students to those who have finished at least 20 questions. Thus, there are in total 1,235 students in this dataset.

For each student s, we use 70% of his/her problem-solving records in the early time period as the training set, the next 15% records as the validation set, and the last 15% as the test set. When processing his/her dataset, the timestamp of the split timestamp between training and validation set is recorded as t_1^s and the split timestamp between validation and test set is recorded as t_2^s. Thus, the *SIQ* network for student s is built with all students' problem-solving records between April 12, 2019 and t_1^s. Each student has a personalized network of different sizes, which helps provide better performance prediction for different students. All the statistical features assigned to student and question nodes in *SIQ* networks are extracted from records before April 12, 2019. Since t_1^s is always later than that date, the leakage of validation and test data in the training process can be avoided.

5.2 Baselines

We compared our approach with both the state-of-the-art GNN models and other traditional machine learning approaches for student performance prediction to extensively evaluate the performance of our approach. These baselines are as follows:

R-GCN: a classical GNN model proposed for networks with various types of edges. We test R-GCN with two variants of our networks. R-GCN (without E2N) represents the input of R-GCN model is a problem-solving network in Figure 4.2(b) without edge features. R-GCN (with E2N) denotes the input is the *SIQ* network with *Edge2Node* transformation.

GBDT: a tree model utilizing the ensemble of trees. To verify the effectiveness of integrating peer information into student performance prediction in our approach, we only consider statistical features of students and questions in Tables 1 and 2 in GBDT.

SVM: a model constructing a hyperplane or hyperplanes to distinguish samples. Similar to GBDT, only statistical features of students and questions in Tables 1 and 2 are fed into SVM.

LR: a classical linear model with a logistic function to model dependent variables. We use features in Tables 1 and 2 for LR.

5.3 Detailed Implementation

Our GNN models are mainly implemented using PyTorch and DGL[5], while GBDT, LR, and SVM are implemented with Sci-kit Learn. For our model R^{2}GCN, we use three parallel input layers to transform original features of 3 types of nodes, as shown in Figure 5. Then we use 3 sequential R-GCN layers with a hidden size of 128. The final two layers of our model are fully-connected neural networks with a hidden size of 128. The activation function used in our model is ReLU. All the GNN-based models in our experiments use Adam as the optimizer and cross entropy as the loss function. We empirically set the learning rate as $1e$-4 and weight decay rate as $1e$-2. The early stopping mechanism is applied to our GNN models. The maximum number of training epochs is set as 400 and the early stopping patience is set as 100 epochs. For GBDT, we set the number of trees as 250, the max depth as 5, and the learning rate as $1e$-3. For SVM, we use Radial Basis Function (RBF) kernel and the regularization parameter is set as 1. To gain reliable results, we trained and tested every model for 10 times and report the average performance.

Table 4: The basic statistics of all students' *SIQ* networks. Q, S, I denote question nodes, student nodes, and interaction nodes respectively. Label shows average number of labeled Q nodes in all students' networks.

		Type	Mean	Max	Min	Label
Short-term	Node	S	2,338	2,873	782	-
		I	28,962	42,409	5,608	-
		Q	1,547	1,608	1,121	89
	Edge	*I-Q, Q-I, I-S & S-I*	28,962	42,409	5,608	-
Long-term	Node	S	2,749	3,556	89	-
		I	52,485	92,573	450	-
		Q	1,540	1,597	419	58
	Edge	*I-Q, Q-I, I-S & S-I*	52,485	92,573	450	-

5.4 Evaluation Metrics

We use three different metrics to evaluate models comprehensively. Here we use s, n_c^s, n^s, $W\text{-}F1^s$ to denote a student, the number of correctly predicted questions for a student, the number of questions in the test set, and the weighted F1 of the prediction results.

Average personal accuracy (AP-Acc) evaluates a model's average prediction accuracy on different students:

$$AP\text{-}Acc = \frac{1}{N} \times \sum_{s=1}^{N} \frac{n_c^s}{n^s}. \tag{4}$$

Overall accuracy (O-Acc) evaluates a model's average prediction accuracy on all predicted questions:

$$O\text{-}Acc = \sum_{s=1}^{N} n_c^s / \sum_{i=1}^{N} n^s. \tag{5}$$

Average personal weighted F1 (APW-F1) evaluates a model's average weighted F1 score on different students:

$$APW\text{-}F1 = \frac{1}{N} \times \sum_{s=1}^{N} W\text{-}F1^s. \tag{6}$$

[5]https://www.dgl.ai/

5.5 Short-term Dataset

5.5.1 Prediction accuracy. In our experiments, we follow the prior research on Graph Neural Networks [5, 12, 18] and compare the model prediction accuracy of our approach with that of other baseline approaches on the test data. Table 5 shows the results of our experiments. Among all the methods, R^{2}GCN performs best across different metrics, which demonstrates the effectiveness of our proposed model. Also, the peer-inspired GNN models (i.e., R^{2}GCN, R-GCN) outperform all traditional machine learning models, which confirms the advantages of our peer-inspired student performance prediction for interactive online question pools. Moreover, the comparison of R-GCN (with E2N) and R-GCN (without E2N) could justify that using the same model, our approach of constructing the *SIQ* network outperforms the original problem-solving network.

Figure 6 further shows the box plots with a beam display to show the detailed accuracy distribution of all the 47 students. The box plots indicate that our method can achieve the highest median accuracy. Moreover, from the distribution of personal accuracy on the right of each box, we can learn that our R^{2}GCN has the least number of students whose accuracy is lower than 0.4, which indicates that extremely lower cases are fewer.

Table 5: The comparison between our method R^{2}GCN and the baseline models on *short-term dataset.*

Model	AP-Acc	O-Acc	APW-F1
R^{2}GCN	**0.6642**	**0.6662**	**0.6148**
R-GCN (with E2N)	0.6302	0.6331	0.5737
R-GCN (without E2N)	0.6151	0.6198	0.5508
GBDT	0.5687	0.5750	0.4398
SVM	0.5734	0.5805	0.4470
LR	0.5928	0.5961	0.5414

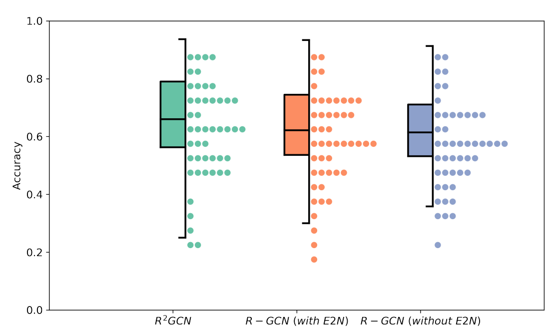


Figure 6: The prediction result of each student with three GNN-based methods, R^{2}GCN, R-GCN (with E2N), and R-GCN (without E2N). The dots on the right of the box plots represent students and their vertical positions denote the approximate accuracy.

5.5.2 Size of labeled data. We further investigate the influence of the training data size on the final prediction accuracy. To maintain the consistency of network structure, test set, and validation set, we choose to keep 40%, 60%, and 80% of records in the training set to conduct this experiment. The experiment results of R^{2}GCN is shown in Figure 7. It is easy to find that the prediction accuracy increases with the growth in training dataset size and finally reaches a relatively stable prediction accuracy around 0.66. This result is reasonable since there will be more ground-truth labels to guide the training of GNNs when more questions are finished.

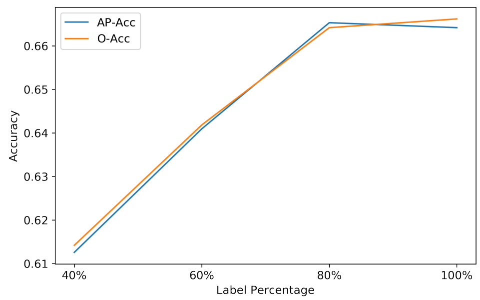


Figure 7: The change of AP-Acc and O-Acc along with the change of size of training labels.

5.5.3 Topological distance among training, validation, and test set. Apart from the number of training labels, the student performance prediction can also be influenced by the topological distance between the test set, and the training or validation set. Thus, we further calculate the average shortest distance in the *SIQ* network between questions in training set, test set, and validation set. These average distances are represented by $\overline{d}_{(train,test)}$, $\overline{d}_{(train,val)}$, and $\overline{d}_{(test,val)}$ respectively. Since the interaction nodes are derived from interaction edges, to simplify our analysis, we remove those nodes and use the problem-solving network in Figure 4.2 to calculate the shortest path distance with NetworkX. The average shortest distance is calculated as follows:

$$\overline{d}_{(\mathrm{X},\mathrm{Y})} = \frac{1}{\|\mathrm{X}\|\|\mathrm{Y}\|} \sum_{i \in \|\mathrm{X}\|} \sum_{j \in \|\mathrm{Y}\|} d_{(x_i, y_j)}, \tag{7}$$

where $\mathbf{X}$ and $\mathbf{Y}$ denote two sets of questions and $d_{(x_i,y_j)}$ is the shortest path distance between x_i and y_j.

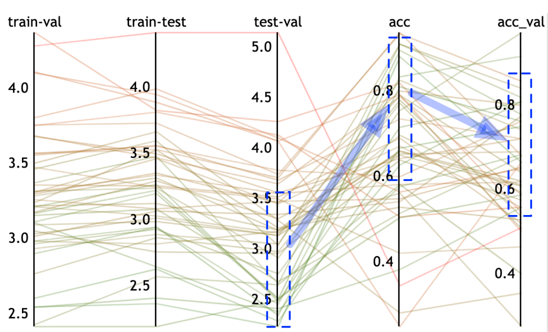


Figure 8: The influence of topological distance among training, validation, and test set on the prediction accuracy. The first three axes represent $\overline{d}_{(train,val)}$, $\overline{d}_{(train,test)}$, and $\overline{d}_{(test,val)}$ of each student. The last two axes denote the average personal accuracy of each student on their test and validation set, respectively. Each line represents the data of one student and the line color indicates the value of $\overline{d}_{(test,val)}$ (the green to red scheme denotes the small to large values).

We use parallel coordinates plot (PCP) to show the influence of the average distances on the student performance prediction accuracy, as shown in Figure 8. We use 5 parallel y-axes to encode the three average distances and two accuracy scores (i.e., test accuracy and validation accuracy), respectively. Each line represents a student in the dataset. Here we also use a green-to-red color scheme to encode $\overline{d}_{(test,val)}$ with green indicating lower $\overline{d}_{(test,val)}$ and red indicating high $\overline{d}_{(test,val)}$. It is easy to recognize that there is a negative correlation between the average distance from test to validation set $\overline{d}_{(test,val)}$ and the accuracy *acc*. Also, we could notice the test accuracy of students with a larger $\overline{d}_{(test,val)}$ is usually lower than their validation accuracy. Such an observation is probably because a large average distance between questions in the test set and questions in the validation set indicates the dissimilarity between them, making the early stopping point not always the best point of achieving the best test accuracy.

5.6 Long-term Dataset

To further evaluate the effectiveness and generalizability of our method, we compare the performance with the baseline methods on the long-term dataset that covers the problem-solving records of more students than the short term dataset. The results in Table 6 indicates that when the number of labeled data is limited, our approach can still achieve high accuracy and F1 score.

Table 6: The comparison between our method R^{2}GCN and baseline models on *long-term dataset.*

Model	AP-Acc	O-Acc	APW-F1
R^{2}GCN	**0.5507**	**0.5671**	**0.5050**
R-GCN (with E2N)	0.5100	0.5313	0.4605
R-GCN (without E2N)	0.5119	0.5296	0.4535
GBDT	0.4836	0.4610	0.3686
SVM	0.4973	0.4718	0.3801
LR	0.4881	0.4904	0.4322

6 DEPLOYMENT

Currently, we are working with engineers from Trumptech to deploy and further evaluate our proposed peer-inspired student performance prediction approach on their interactive online math question pool, Learnlex. Our approach has already been integrated into it and is still under active testing and iterative improvement, which is expected to be finished in September, 2020. After we finish the initial integration of our approach into Learnlex, one senior engineer commented that *"the prediction accuracy is impressive compared with other state-of-the-art approaches. It provides student performance prediction on each question, which can hardly be finished by prior knowledge tracing-based algorithms."* During the deployment process, one key issue we faced is how to handle the *cold start problem*, as our R^{2}GCN model intrinsically requires that there should be some existing training data for newly-registered students on Learnlex. To cope with this issue, Learnlex will ask each newly-registered student to finish 15 starting questions that are carefully selected by question designers by considering the question popularity (i.e., how many prior students have worked on it), the grade of the student, and the question difficulty level. By combining the newly-registered student's interaction data on the starting questions and prior students' existing problem-solving interaction data, we build the initial heterogeneous *SIQ* network for a new student and further train a R^2GCN model to predict his/her performances on other questions, which can achieve the same level of prediction accuracy as shown in Table 6.

Once the testing and deployment is finally finished, our approach will work as the core component of question recommendation module on Learnlex, which can recommend appropriate questions that satisfy the requirements of different students. According to the market share estimation by Trumptech, Learnlex will be used by more than 100,000 students in the coming three years, which means that over 100,000 students will benefit from the personalized online learning from our approach in the near future.

7 DISCUSSION

Generalizability of our method. Our method has the potential to be applied on touch screen devices since basic gestures are similar to mouse movement interactions (e.g., Tap, Drag). However, there are some unique gestures on touch screens (e.g., Swipe, Flick), so we may need to add other features related to such gestures to extract more information. Also, our method can be applied to other online scenarios with rich mouse interactions of users (e.g., E-sports).

Cold start problem. Cold start is a general problem for prediction and a set of training data is needed to train the prediction model. Our method based on GNN also suffers from this problem. It requires that the questions and students have enough records. To handle this issue, we ask the newly-registered students to finish 15 carefully-selected starting questions before R^2GCN is used to predict their performances on other questions, as discussed in Section 6. Some more advanced method can be further explored to cope with it.

8 CONCLUSION

Student performance prediction is important for online education. However, most existing work targets at MOOC platforms in which a predefined order of the learning materials (e.g., course videos and questions) is available. Little research has been done on interactive online question pools with no predefined question order. In this paper, we propose a novel approach based on graph neural networks to predict students' score level on each question in interactive online question pools. Specifically, we formalize the student performance prediction in interactive online question pools as a node classification problem on a large heterogeneous network consisting of questions, students, and the interactions between them, better capturing the underlying relationship among questions and students. Then, we propose a novel GNN model R^2GCN by adapting the classical R-GCN model and further incorporating a residual connection between different convolutional layers. Our detailed evaluations on a real-world dataset collected from a K-12 mathematics online question pool show that our approach outperforms both traditional machine learning models (e.g., Logistic Regression (LR) and Gradient Boosted Decision Tree (GBDT) model) and the classical R-GCN model [18], in terms of accuracy and F1 score.

In future work, we plan to extend the proposed student performance prediction approach using GNN to other interactive online question pools designed for touch screen devices like iPad, and further evaluate our model on larger datasets collected with more students and questions. Also, it will be interesting to further explore whether the proposed approach can be adapted to other analysis tasks in online education, e.g., detecting cheating behaviors.

ACKNOWLEDGMENTS

This work is partially sponsored by Innovation and Technology Fund (ITF) with No. ITS/388/17FP. Y. Wang is the corresponding author. We would like to thank Trumptech (Hong Kong) Limited for providing the dataset and giving us feedback on the proposed method, Zhihua Jin for suggestions and Min Xu for proofreading.

REFERENCES

[1] Penghe Chen, Yu Lu, Vincent W. Zheng, and Yang Pian. 2018. Prerequisite-Driven Deep Knowledge Tracing. In *ICDM*. 39–48.

[2] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. In *DLRS@RecSys*. 7–10.

[3] Ali Daud, Naif Radi Aljohani, Rabeeh Ayaz Abbasi, Miltiadis D. Lytras, Farhat Abbas, and Jalal S. Alowibdi. 2017. Predicting Student Performance using Advanced Learning Analytics. In *WWW*. 415–421.

[4] Nima Dehmamy, Albert-László Barabási, and Rose Yu. 2019. Understanding the Representation Power of Graph Neural Networks in Learning Graph Topology. In *NeurIPS*. 15387–15397.

[5] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. 2017. Neural Message Passing for Quantum Chemistry. In *ICML*. 1263–1272.

[6] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NeurIPS*. 1024–1034.

[7] Qian Hu and Huzefa Rangwala. 2019. Academic Performance Estimation with Attention-based Graph Convolutional Networks. In *EDM*.

[8] Suhang Jiang, Adrienne E. Williams, Katerina Schenke, Mark Warschauer, and Diane K. O'Dowd. 2014. Predicting MOOC Performance with Week 1 Behavior. In *EDM*. 273–275.

[9] Valentin Kassarnig, Andreas Bjerre-Nielsen, Enys Mones, Sune Lehmann, and David Dreyer Lassen. 2017. Class Attendance, Peer Similarity, and Academic Performance in a Large Field Study. *PloS one* 12, 11 (2017).

[10] Valentin Kassarnig, Enys Mones, Andreas Bjerre-Nielsen, Piotr Sapiezynski, David Dreyer Lassen, and Sune Lehmann. 2018. Academic Performance and Behavioral Patterns. *EPJ Data Science* 7, 1 (2018), 10.

[11] Gregor E. Kennedy, Carleton Coffrin, Paula G. de Barba, and Linda Corrin. 2015. Predicting Success: How Learners' Prior Knowledge, Skills and Activities Predict MOOC Performance. In *LAK*. 136–140.

[12] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[13] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. 2016. Gated Graph Sequence Neural Networks. In *ICLR*.

[14] Hiromi Nakagawa, Yusuke Iwasawa, and Yutaka Matsuo. 2019. Graph-Based Knowledge Tracing: Modeling Student Proficiency Using Graph Neural Network. In *WI*. 156–163.

[15] Hao Peng, Jianxin Li, Qiran Gong, Yangqiu Song, Yuanxing Ning, Kunfeng Lai, and Philip S. Yu. 2019. Fine-grained Event Categorization with Heterogeneous Graph Convolutional Networks. In *IJCAI*. 3238–3245.

[16] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J. Guibas, and Jascha Sohl-Dickstein. 2015. Deep Knowledge Tracing. In *NeurIPS*. 505–513.

[17] Zhiyun Ren, Huzefa Rangwala, and Aditya Johri. 2016. Predicting Performance on MOOC Assessments using Multi-Regression Models. In *EDM*. 484–489.

[18] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *ESWC*. 593–607.

[19] Nguyen Thai-Nghe, Lucas Drumond, Artus Krohn-Grimberghe, and Lars Schmidt-Thieme. 2010. Recommender System for Predicting Student Performance. In *RecSysTEL*. 2811–2819.

[20] Khushboo Thaker, Paulo Carvalho, and Kenneth Koedinger. 2019. Comprehension Factor Analysis: Modeling Student's Reading Behaviour: Accounting for Reading Practice in Predicting Students' Learning in MOOCs. In *LAK*. 111–115.

[21] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. 2019. Heterogeneous Graph Attention Network. In *WWW*. 2022–2032.

[22] Huan Wei, Haotian Li, Meng Xia, Yong Wang, and Huamin Qu. 2020. Predicting Student Performance in Interactive Online Question Pools Using Mouse Interaction Features. In *LAK*. 645–654.

[23] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *SIGKDD*. 974–983.

[24] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. 2018. Graph Neural Networks: A Review of Methods and Applications. *ArXiv* abs/1812.08434 (2018).