RESEARCH-ARTICLE

Prerequisite-Enhanced Category-Aware Graph Neural Networks for Course Recommendation

JIANSHAN SUN, Hefei University of Technology, Hefei, Anhui, China

SUYUAN MEI, Hefei University of Technology, Hefei, Anhui, China

KUN YUAN, Hefei University of Technology, Hefei, Anhui, China

YUANCHUN JIANG, Hefei University of Technology, Hefei, Anhui, China

JIE CAO, Hefei University of Technology, Hefei, Anhui, China

Published: 28 February 2024
Online AM: 29 January 2024
Accepted: 22 January 2024
Received: 31 July 2023

ACM Transactions on Knowledge Discovery from Data, Volume 18, Issue 5 (June 2024)
https://doi.org/10.1145/3643644
EISSN: 1556-472X

Prerequisite-Enhanced Category-Aware Graph Neural Networks for Course Recommendation

JIANSHAN SUN, School of Management, Hefei University of Technology, Hefei, China and Key Laboratory of Process Optimization and Intelligent Decision Making, Ministry of Education, Hefei, China
SUYUAN MEI, School of Management, Hefei University of Technology, Hefei, China
KUN YUAN, School of Management, Hefei University of Technology, Hefei, China and Intelligent Interconnected Systems Laboratory of Anhui Province (Hefei University of Technology), Hefei, China
YUANCHUN JIANG, School of Management, Hefei University of Technology, Hefei, China and Key Laboratory of Philosophy and Social Sciences for Cyberspace Behaviour and Management, Hefei, China
JIE CAO, School of Management, Hefei University of Technology, Hefei, China

The rapid development of Massive Open Online Courses (MOOCs) platforms has created an urgent need for an efficient personalized course recommender system that can assist learners of all backgrounds and levels of knowledge in selecting appropriate courses. Currently, most existing methods utilize a sequential recommendation paradigm that captures the user's learning interests from their learning history, typically through recurrent or graph neural networks. However, fewer studies have explored how to incorporate principles of human learning at both the course and category levels to enhance course recommendations. In this article, we aim at addressing this gap by introducing a novel model, named Prerequisite-Enhanced Catory-Aware Graph Neural Network (PCGNN), for course recommendation. Specifically, we first construct a course prerequisite graph that reflects the human learning principles and further pre-train the course prerequisite relationships as the base embeddings for courses and categories. Then, to capture the user's complex learning patterns, we build an item graph and a category graph from the user's historical learning records, respectively: (1) the item graph reflects the course-level local learning transition patterns and (2) the category graph provides insight into the user's long-term learning interest. Correspondingly, we propose a user interest encoder that employs a gated graph neural network to learn the course-level user interest embedding and design a category transition pattern encoder that utilizes GRU to yield the category-level user interest embedding. Finally, the two fine-grained user interest embeddings are fused to achieve precise course prediction. Extensive experiments on two real-world datasets demonstrate the effectiveness of PCGNN compared with other state-of-the-art methods.

This work was supported by the National Natural Science Foundation of China (72271083, 72342011, 72101076, 72101072), the Fundamental Research Funds for the Central Universities of China (Grant No. PA2023IISL0090, JZ2023YQTD0075).
Authors' addresses: J. Sun, School of Management, Hefei University of Technology, 193 Tunxi Road, Hefei, Anhui, 230009, China, and Key Laboratory of Process Optimization and Intelligent Decision Making, Ministry of Education, Hefei, Anhui, 230009, China; e-mail: sunjs9413@hfut.edu.cn; S. Mei and J. Cao, School of Management, Hefei University of Technology, 193 Tunxi Road, Hefei, Anhui, 230009, China; e-mails: 1330135114@qq.com, cao_jie@hfut.edu.cn; K. Yuan (Corresponding author), School of Management, Hefei University of Technology, 193 Tunxi Road, Hefei, Anhui, 230009, China, and Intelligent Interconnected Systems Laboratory of Anhui Province (Hefei University of Technology), Hefei 230009, Anhui, P.R. China; e-mail: yuankun@hfut.edu.cn; Y. Jiang, School of Management, Hefei University of Technology, 193 Tunxi Road, Hefei, Anhui, 230009 China, and Key Laboratory of Philosophy and Social Sciences for Cyberspace Behaviour and Management, Hefei, Anhui, 230009, China; e-mail: ycjiang@hfut.edu.cn.

ACM 1556-4681/2024/02-ART112
https://doi.org/10.1145/3643644

CCS Concepts: • **Information systems** → **Data mining**; *Decision support systems*; Users and interactive retrieval;

Additional Key Words and Phrases: Course recommendation, MOOCs, prerequisite relationship, graph neural network

ACM Reference Format:
Jianshan Sun, Suyuan Mei, Kun Yuan, Yuanchun Jiang, and Jie Cao. 2024. Prerequisite-Enhanced Category-Aware Graph Neural Networks for Course Recommendation. *ACM Trans. Knowl. Discov. Data.* 18, 5, Article 112 (February 2024), 21 pages. https://doi.org/10.1145/3643644

1 INTRODUCTION

The proliferation of **Massive Open Online Courses** (**MOOCs**) platforms such as Udemy, Coursera, and edX provides low-cost opportunities for everyone to access massive superior online courses offered by worldwide prestigious universities for continuous learning and skills improvement [3]. However, due to the limitation of their knowledge and cognitive level, users may struggle to find suitable courses that meet their learning requirements, which instead decreases their learning efficiency and enthusiasm, particularly in the context of the increasing number of online courses [6, 19]. Therefore, the dilemma of online learning users demands an efficient course recommender system to alleviate the course overload problem and draw up personalized course learning plans for users of all backgrounds and learning stages for high-efficiency learning.

In contrast to conventional recommendation scenarios, users' learning decisions heavily correlate with their current capacity and future development goals [26]. Traditional course recommendation methods mainly utilize machine learning and collaborative filtering techniques to capture users' preferences based on user profiles, course attribution, and historical learning data without considering the learning order. Kabbur et al. [14] proposed the **factored item similarity model** (**FISM**) which embeds each course with an embedding vector and averages the embeddings of all history courses to the user's preference. Aher et al. [1] constructed a course recommender system based on K-means clustering and Apriori association rule algorithms for newly enrolled students. Jing et al. [13] developed a collaborative filtering-based course recommendation algorithm based on user interest, demographic profiles, and course prerequisite relationships. While considering the varying user interest and improved ability as more courses are learned, more deep neural network-based recommendation methods are proposed to model the sequential learning behaviors. For example, **neural attentive item similarity** (**NAIS**) [9], **neural attentive session-based recommendation** (**NASR**) [16], and session-based recommendation with **graph neural networks** (**GNNs**) [29] can be used to extract users' preferences and recommend the next suitable course. However, the diverse user interests and noisy courses in the historical learning data hinder the further accuracy improvement of course recommendation methods. To address these challenges, Zhang et al. [33] proposed a hierarchical reinforcement learning algorithm to remove noisy courses in user profiles for better course recommendation. Lin et al. [17] designed **Dynamic Attention and Hierarchical Reinforcement Learning** (**DARL**) course recommendation framework which adaptively adjusts attention weight to capture the user's dynamic interests in sequential learning behaviors.

Despite the great performance on course recommendations, existing methods still suffer from two limitations. ***First***, most methods have not fully exploited the course-level prerequisite relationships when recommending the next suitable course. In reality, the prerequisite relationship refers to the learning order of two courses in which one course builds a foundational knowledge base for the other course. For instance, as depicted in Figure 1, programming courses can only be

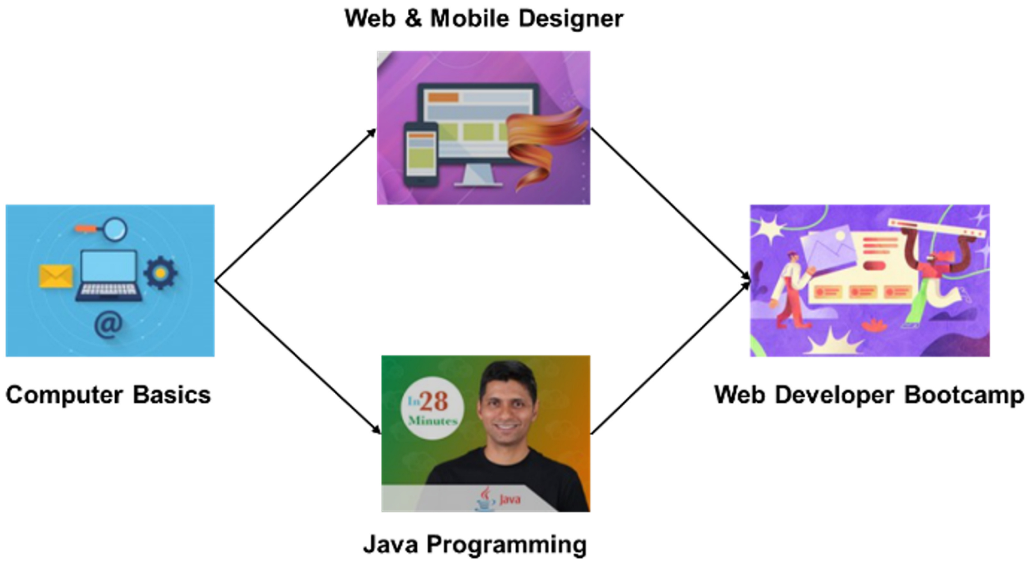


Fig. 1. An example of the prerequisite relationship between courses.

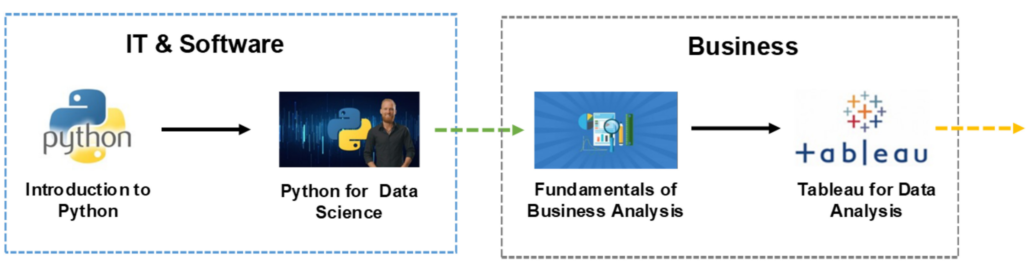


Fig. 2. Learning behavior sequence of a user.

taken by users who possess basic computer knowledge. As a result, *"Computer Basics"* serves as the prerequisite for *"Web & Mobile Designer"* and *"Java Programming"*. Although some research has attempted to address this issue, these heuristic methods simply count the conditional probability of each ordered course pair, which is insufficient to effectively model the complex learning process.

Second, they have also overlooked the indicative category-level information that reflects users' learning transition patterns within and between categories. Specifically, users tend to concentrate their learning interests within the same category for some time. Once their requirements have been met, they will turn to some specific categories following some transition patterns. We illustrate this phenomenon with an example of a true user's learning process, as shown in Figure 2. In pursuit of becoming a business analyst, the user first takes two Python-related courses in the *"IT & Software"* category to acquire programming skills, before turning to business analysis courses in the *"Business"* category. In this example, the transition pattern of the learning process between the two categories obeys the rule of becoming a professional business analyst. Once the category of the future course is determined, it assists to narrow the candidate course range and reduce prediction difficulty. Therefore, incorporating the principles of human learning at both course and category levels benefits the course recommender system accurately capturing users' dynamic interests and achieving precise course recommendations.

To this end, we propose a **Prerequisite-enhanced Category-aware Graph Neural Network** (**PCGNN**) model for accurate course recommendation. The main idea is to integrate the course-level prerequisite relationship and category-level learning interest transition patterns into deep GNNs to effectively capture the users' learning interests and recommend suitable courses. Specifically, (1) for the course-level prerequisite relationship, we first construct a weighted directed prerequisite graph including course and category entities and multiple types of relations based on the massive learning records of all users. Subsequently, we introduce the TransE model, a widely used embedding method in the knowledge graph, to pre-train the prerequisite graph and obtain the course and category embeddings. (2) Then to infer the user's learning interest from their sequential learning behaviors, we propose a **User Interest Encoder** (**UIE**) that transfers the learning sequence into a course graph and models it with **Gated Graph Neural Network** (**GGNN**). (3) In light of the indicative user's interest transition pattern exhibited in the category sequences, we have also developed a Category-enhanced **Transition Pattern Encoder** (**TPE**) as an auxiliary task for the next course' category prediction and facilitate narrowing down the range of potential courses. Ultimately, (4) we integrate the learned course-level learning interest and category-level transition pattern to recommend the most suitable course for the user. To validate the effectiveness of PCGNN, extensive experiments have been conducted on two real-world datasets, demonstrating the superiority of our model compared with several state-of-the-art methods. Furthermore, our model performs well for users with different lengths of the learning sequence, validating its generalizability for the data sparsity problem.

The remainder of this article is organized as follows. Section 2 briefly concludes the related work. In Section 3, we conduct data exploration about users' learning behavior and formally define the problem. Then we introduce the details of the proposed model in Section 4. The experimental results are reported in Section 5. Finally, we conclude the article in Section 6.

2 RELATED WORK

In this section, we summarize the related work in two aspects including course recommendation and sequential recommendation.

2.1 Course Recommendation

With the rapid development of online education platforms, the urgent need for an effective way of personalized course recommender system has attracted the attention of many researchers. Without consideration of the order of learning behaviors, early studies focus on leveraging machine learning and collaborative filtering techniques to capture users' learning interests [13, 14]. Huang et al. [12] improved the user multi-similarity algorithm by integrating users' academic information, social information, and interactive information for course recommendation. Aher et al. [1] constructed a course recommender system based on K-means clustering and Apriori association rule algorithms for newly enrolled students. To provide personalized course needs, Chen et al. [4] employed the association rule algorithm and collaborative filtering technology to recommend courses for users to obtain higher scores. Wu et al. [30] integrated the user-based collaborative filtering algorithm with user scores and course attributes. Noting the increase of learner capacity through learning courses, Tian et al. [25] combined the capacity tracing model with collaborative filtering to quantify the knowledge state or ability of learners and recommend appropriate courses for learners. However, this stream of research lacks adaptivity and flexibility to recommend courses as the increase of users' capacities through learning courses.

With the spur of deep learning, recent methods have incorporated deep learning techniques for more accurate course recommendations [7]. Fan et al. [5] proposed a deep learning method with a multi-attention mechanism for interpretable MOOC recommendations. Zhao et al. [34] extracted course prerequisite relations from course-video captions and further combined them with a neural attention network for course recommendation. Zhang et al. [32] fused learners' features and course content attributes into deep belief networks and finetuned them with feedback. Considering the noisy interactions in the historical learning records, Zhang et al. [33] propose a hierarchical reinforcement learning method, which can remove noisy courses without expert annotation. To balance the tradeoff between exploration and exploitation, Lin et al. [17] designed a hierarchical reinforcement learning model with a dynamic recurrent mechanism that explores the user's future preferences for course recommendation. Despite the significant achievements made by previous research, these content-based methods cannot trace the learners' learning interests in the continuous learning process and fail to accurately recommend courses.

2.2 Sequential Recommendation

When considering the order of the learning process, the course recommendation can be regarded as a sequential recommendation problem, in which session dependencies are modeled to predict the next item. Traditional methods [2, 22, 23] mainly employ Markov Chain to capture the sequential patterns in the session which only model the first-order transitions and limit the capacity of high-order dependencies modeling. As deep learning technology continues to advance, more **Recurrent Neural Network** (**RNN**)-based methods are proposed for the sequential recommendation. Hidasi et al. [10] proposed the first work called GRU4Rec which applies multi-layer GRU to model item sequences. On this basis, NARM [16] integrated the attention mechanism into GRU to better learn representative item transition patterns for session recommendation. Afterward, LSTM [35], BERT [24], Transformer [15], and other variants of RNNs [11] have been introduced for the sequential recommendation, bringing further performance improvement. However, although RNNs-based methods have shown superior competence in sequential recommendation, they are capable of local dependency modeling via the chronology of the given sequence, yet fall short in non-sequential patterns, which limits their further application.

To address it, a series of GNNs-based methods have been proposed which transfer the session sequence into a session graph and further employ the powerful graph modeling capacity of GNNs to capture the complex transition patterns [20, 27]. SRGNN [29] firstly constructs session graphs from all session sequences, which are input to the Gated GNN to learn the global preference embeddings and further combine them with current interest embeddings as the session latent embeddings. Based on SRGNN, GC-SAN [31] proposes a graph contextualized self-attention model to learn the long dependencies in the session. Wang et al. [28] propose a novel GCE-GNN model to exploit the item transitions from all sessions and learn two levels of item embeddings from the session graph and global graph for user preference inference. Liu et al. [18] build the CaSe4SR model that utilizes an item graph and a category graph to learn item-level and category-level representations to augment sequential recommendation via two Gated GNNs. Although GNN-based methods have achieved significant results, however, they mainly model the item-transition patterns for general sequential recommendation scenarios, while do not apply to varying learning interests modeling of online learning course recommendation. In contrast, we argue that the user's varying learning interests are reflected in both courses and corresponding category sequences, and further propose the PCGNN model following the principles of human learning for accurate next-course recommendations.

3 PRELIMINARY AND PROBLEM DEFINITION

In this section, we first conduct data exploration on users' online learning behaviors to unveil the course-level prerequisite relationships and category-level learning transition patterns, and then give a formal definition of the course recommendation problem.

Table 1. Notation

Notation	Description
u, v, c	user u, course v, and category c
M, N, N_c	the size of users, courses and categories
S_u, E_u	the learning sequence and embedding sequence for user u
C_u, F_u	the category sequence and embedding sequence for user u
s_u^s	the learning interest representation of user u
s_u^c	the category transition pattern representation of user u
P	the inversed position embedding matrix
l	the length of the sequence for user u
h, r, t	the head, relation, and tail nodes in a knowledge graph
e_h, e_r, e_t	the corresponding embeddings of h, r, t
e_i	the embedding of course v_i
f_i	the embedding of category c_i
$d \in \mathbb{N}$	the dimension of the embedding
λ_{pg}	the weight parameter of the prerequisite graph loss function

3.1 Problem Definition

Suppose we have a set of users $U = \{u_i\}_{i=1}^{M}$, and a set of courses $V = \{v_j\}_{j=1}^{N}$, where M and N denote the total number of users and courses, respectively. Meanwhile, the category set possessed by courses is denoted as $C = \{c_k\}_{k=1}^{N_c}$. For each user u, we extract all learning courses and further sort them in chronological order as the historical learning sequence according to the timestamp, which is denoted by $S_u = \{v_{u,1}, v_{u,2}, v_{u,3}, \ldots, v_{u,l}\}$, where l is the length of the sequence.

In this work, we aim at constructing a course recommendation model which can predict the next course that user u is most likely interested in when given the historical learning sequence S_u. Our notation is summarized in Table 1.

3.2 Data Exploration

We analyze a real-world MOOC dataset collected from Udemy to explore the users' online learning behaviors. Udemy is an educational platform that provides users with opportunities to access free or paid courses in various categories. The dataset consists of 130,019 users, 23,581 courses belonging to 15 categories, and 1,094,914 user-course learning records with timestamps. The average length of the user's learning record is 8.42. More details about the dataset are concluded in Section 5.

3.2.1 Course Prerequisite Relationship. As we know that education is systematic engineering, in which all courses are designed according to the content relevancy and learning difficulty in compliance with the principles of human learning, i.e., course prerequisite relationship. However, for MOOC platforms, course offerings depend on the interests of experts, which floods the platform with massive and disorganized courses and further sets obstacles for users to scientifically learn. Concerning this, we examine whether the courses on Udemy also exhibit the course prerequisite relationship. Specifically, we sort each user's learning record in chronological order as a sequence according to the timestamp. We consider any two adjacent courses in the sequence to form a directed pair, denoting that the user switches to the next course after finishing the previous one. Then we go through all users' learning records and count the occurrence of each directed pair. Table 2 lists the top 5 directed pairs with the highest occurrence. From the table, we observe that some certain directed pairs manifest with greater frequency than their inverse directed

Table 2. Top 5 with the Highest Occurrence between Courses

rank	previous course	previous category	next course	next category	occurrence
1	CSharp tutorial for beginners	Development	CSharp Intermediate: classes, interfaces and OOP	Development	850
2	React Redux	Development	React Redux tutorial	Development	842
3	CSharp Intermediate: classes, interfaces and OOP	Development	CSharp advanced	Development	789
4	AWS concepts	IT & Software	AWS essentials	IT & Software	690
5	Understand javascript	Development	Understand nodejs	Development	622

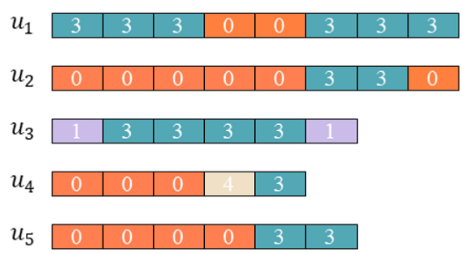


Fig. 3. The visualization of five randomly selected category sequences. Best views in color.

counterparts. For example, the majority of users exhibit a learning pattern for selecting "*CSharp Intermediate: classes, interfaces, and OOP*" after completing the "*CSharp tutorial for beginners*" learning. However, fewer users perform the inverse operation, which violates the foundational principles of learning. We filter out all directed pairs with occurrences exceeding 30 and obtain 929 pairs. Based on these, we conclude that MOOC platforms like Udemy exits the course prerequisite relationship hidden in the users' learning records.

3.3 Category-Level Learning Transition Patterns

Noting the indicative learning signals disclosed from the category sequences, we also explore the learning transition patterns from the course category level. First, we randomly select five users' learning records and visualize the corresponding category sequences in Figure 3. Different colors denote distinctive course categories. From the figure, we can see that (1) from the intra-category view, most users show a consistent learning pattern in one category. To check the universality, we further adopt a sliding window approach with a window size of 4 to divide the category sequences into a collection of subsequences. We examine all category subsequences and find that 84.5% of them belong to the same category. This observation is quite reasonable since learners must take a series of courses belonging to the same category before be a specialist in that domain.

Besides, (2) we also observe that after a period of learning in one category, some users will turn to another category and restart another period of learning. To further explore the inter-category transition patterns, similar to Section 3.2.1, we also count the occurrence of adjacent pairwise categories in the category sequence and list several pairs with the highest occurrence in Table 3. From the table, we infer that some specific transition patterns exist between certain pairwise categories, probably due to the interdisciplinary knowledge required by some jobs. Just like business analysts need to simultaneously have both programming skills and commercial background knowledge, the learners must learn two categories of courses step by step.

In summary, the course-level prerequisite relationship and the category-level learning transition patterns provide opportunities for us to precisely capture the user's short-term learning interest and long-term learning goal from the historical learning sequence, which benefits constructs a

Table 3. Several Most Weighted Edges between Categories

previous category	next category	Edge weight
Design	Development	17,167
	Business	5,216
	Marketing	4,437
Health&Fitness	Personal Development	4,139
	Business	1,928
Photography	Design	1,589
	Development	1,530

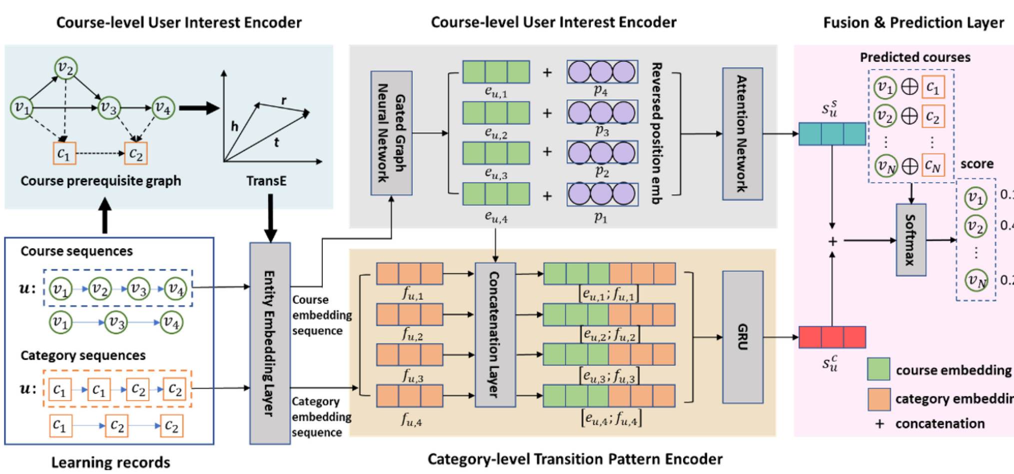


Fig. 4. The overall framework of the PCGNN model.

personalized course recommender system for users with any background at any learning stage for high-efficiency learning.

4 METHODOLOGY

Motivated by the observations from the real-world MOOC data analysis, in this section, we propose a PCGNN model, which exploits the course-level prerequisite relationship and category-level learning transition patterns for accurate course recommendation. The overall framework of PCGNN is presented in Figure 4, which consists of four key components: (1) **Course Prerequisite Graph Pre-training (PGP)**. First, based on the ordered massive learning sequences, we construct a course prerequisite graph, in which courses, categories, and their multiple types of relations are depicted and further pre-trained by the TransE model. (2) **Couser-level UIE**. Taking the user's historical learning sequence as input, we employ a GGNN to model the learning behaviors dependencies and further learn the user's learning interest embedding based on all learned courses by position-aware attention mechanism. (3) **Category-level TPE**. Besides, considering the category information provides additional clues to reveal the user's learning interest, we leverage a GRU module to capture the user's category-level transition pattern from the category sequence. (4) **Fusion & Prediction Layer**. Finally, the user's learning interest embedding and the next course's category embedding are concatenated and codetermine the next interested course. The loss function of the main recommendation task as well as the auxiliary course PGP task is jointly optimized for accurate course recommendation.

4.1 Course Prerequisite Graph Pre-Training

A typical difference between the course recommendation and other session-based recommendation tasks is the inherent course prerequisite relationship supported by learning theory. We also confirm this characteristic of MOOC in the previous section. To better depict the course prerequisite relationship, we propose to construct a course prerequisite graph $\mathcal{G}_g = (\mathcal{V}_g, \mathcal{E}_g)$ based on the directed pairs in Section 3.2.1 on all learning sequences. In the graph, $\mathcal{V}_g = \{V, C\}$ denotes the graph node set that contains both courses and categories, and $\mathcal{E}_g = \{\mathcal{E}^v, \mathcal{E}^{vc}, \mathcal{E}^c\}$ represents the edge set that contains three different kinds of relations between nodes. For example, for two courses v_i and v_j with the corresponding categories c_i and c_j, the edge $\mathcal{E}^v_{ij} = (v_i, r_v, v_j)$ indicates the course prerequisite relation r_v between v_i and v_j; $\mathcal{E}^{vc}_{ij} = (v_i, r_{vc}, c_j)$ denotes the belonging relation r_{vc} of course v_i to category c_i; and $\mathcal{E}^c_{ij} = (c_i, r_c, c_j)$ is the category transition relation r_c between c_i and c_j. Additionally, for each edge $e \in \mathcal{E}_g$, we count the occurrence as the importance weight and further only keep edges whose occurrences exceed a predefined threshold for efficiency consideration. In this manner, $\mathcal{G}_g$ is constructed as a heterogeneous directed weighted graph.

Intuitively, preserving the graph prerequisite relationship is beneficial for capturing the user's learning interest. To this end, we propose to leverage TransE [33] model, a widely used knowledge graph embedding method, to learn the course and category entities embeddings as well as their multiple relations. More specifically and generally, given a triplet relation (h, r, t), the basic idea of TransE is to learn embeddings of entities and relations by optimizing the translation principle $e_h + e_r \approx e_t$, where $h, t \in \mathcal{V}_g$ and $r \in \{r_v, r_{vc}, r_c\}$ (the set of relations), and e_h, $e_r, e_t \in \mathbb{R}^d$ are embeddings of h, t, r. Following the energy-based framework, the energy score of the triplet is formulated as follows:

$$g(h, r, t) = \|e_t - e_h - e_r\|_2^2 \,. \tag{1}$$

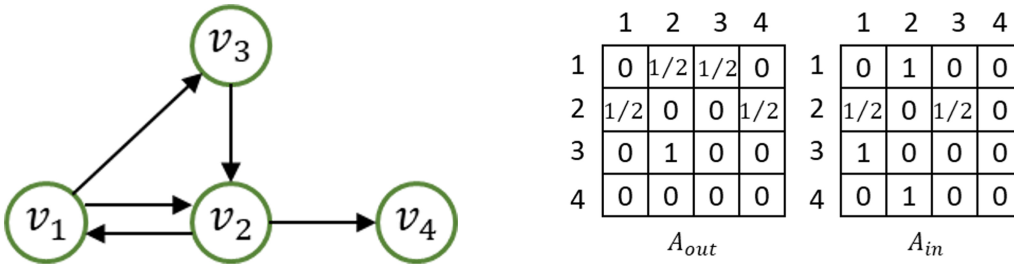
A higher score of $g(h, r, t)$ indicates the triplet is less likely to exist in the real world, and vice versa. To learn the embeddings of each entity and relation, the TransE method optimizes a margin-based ranking loss which regards true triplets should yield a lower score than the broken triplets:

$$\mathcal{L}_{kg} = \sum_{(h,r,t) \in T} \sum_{(h,r,t') \notin T} max(\alpha + g(h, r, t) - g(h, r, t'), 0), \tag{2}$$

where (h, r, t') is a broken triplet derived from the true triplet $(h, r, t) \in T$ in which we randomly replace t with t'; T is the ground-truth triplet set; $\alpha > 0$ is a margin hyper-parameter. By applying the TransE to model the fine-grained triplets of course-course prerequisite relation $\mathcal{E}^v_{ij}$, course-category belonging relation $\mathcal{E}^{vc}_{ij}$, and category-category transition relation $\mathcal{E}^c_{ij}$, we inject the multiple entities and relations into embedding vectors while increasing the model expressiveness ability.

4.2 Course-Level User Interest Encoder

Given user u's historical learning sequence S_u, it naturally appeals to capture the sequential learning interest via sequence modeling methods like LSTM, which is a typical neural-based sequence model capable of modeling long-term dependency for consecutive items. However, in a real learning scenario, the user's decision of the next enrolled course is more complicated, which is not only correlated with the current consecutive course but also the non-adjacent previous ones. To address this, inspired by the work [8], we consider the learning sequence from a graph view and transfer it into a session graph. The emerging GGNN has shown powerful capacity in modeling session connection relationships between nodes on graph data. Therefore, in this section, we



(a) learning sequence

(b) outgoing and ingoing matrix

Fig. 5. An example of a learning sequence and adjacent matrix.

propose a UIE which leverages the GGNN to capture the user's course-level learning interest from the transferred course session graph. In what follows, we first present how to learn course representations via GGNN, and then obtain the course-level user learning interest representation with position-aware attention mechanism.

Specifically, for course $v_{u,i}$ in user u's historical learning sequence $S_u = \{v_{u,1}, v_{u,2}, v_{u,3}, \ldots, v_{u,l}\}$, let $e_{u,i} \in \mathbb{R}^d$ denotes the embedding vector with the dimension of d, which is initialized with course pre-trained embedding in Section 4.1. Then we obtain the user u's learning sequence embedding $E_u = \{e_{u,1}, e_{u,2}, e_{u,3}, \ldots, e_{u,l}\}$. Then we construct a directed session graph $\mathcal{G}_u = \{V_u, \mathcal{E}_u\}$ from S_u to capture course transition patterns, in which $(v_{u,i}, v_{u,i+1}) \in \mathcal{E}_u$ denotes that user u learns $v_{u,i+1}$ after $v_{u,i}$ consecutively. To describe the directed graph, as shown in Figure 5, we build the adjacent matrix A_u consisting of an out-edge matrix A_u^{out} and in-edge matrix A_u^{in} to represent the contextual dependency relationships of any course in the personalized learning process.

4.2.1 GGNN-Based Course Representation Learning. Since the directed session graph reveals the user's course-level learning transition pattern, we propose to leverage GGNN which consists of information propagation and information aggregation layers to encode the contextual dependency relationships for course representation learning. Here, we mainly describe the operation at t-step, which can be generalized to multiple layers.

Information Propagation: Let $E_u^{(t-1)} = [e_{u,1}^{(t-1)}, \ldots, e_{u,l}^{(t-1)}]$ denotes the sequence representation at $(t-1)$-step. Then the information propagation layer can be expressed as

$$a_{u,i}^{(t)} = Concat\left(A_{u,i}^{in}\left(E_u^{(t-1)}W_{in} + b_{in}\right), A_{u,i}^{out}\left(E_u^{(t-1)}W_{out} + b_{out}\right)\right), \quad (3)$$

where $A_{u,i}^{in}, A_{u,i}^{out} \in \mathbb{R}^{1\times l}$ are ith row of the ingoing and outgoing matrix, $W_{in}, W_{out} \in \mathbb{R}^{d\times d}$ are the parameter matrices, and $b_{in}, b_{out} \in \mathbb{R}^d$ are the bias vectors.

Information Aggregation: After obtaining the neighborhood information $a_{u,i}^{(t)}$ at t-step, it is further combined with the self-embedding $e_{u,i}^{(t-1)}$ to update $v_{u,i}$'s representation at t-step. Note that the aggregated neighborhood information is valuable but noisy. Therefore, it is necessary to filter out noises before fusion. Based on this, we apply the **Gated Recurrent Units** (**GRUs**) to implement the aggregation as follows:

$$z_{u,i}^{(t)} = \sigma\left(W_{z1}a_{u,i}^{(t)} + H_{z1}e_{u,i}^{(t-1)}\right), \quad (4)$$

$$r_{u,i}^{(t)} = \sigma\left(W_{r1}a_{u,i}^{(t)} + H_{r1}e_{u,i}^{(t-1)}\right), \quad (5)$$

$$\tilde{e}_i^{(t)} = \tanh\left(W_{h1} a_{u,i}^{(t)} + H_{h1}\left(r_{u,i}^{(t)} \odot e_{u,i}^{(t-1)}\right)\right), \tag{6}$$

$$e_{u,i}^{(t)} = \left(1 - z_{u,i}^{(t)}\right) \odot e_{u,i}^{(t-1)} + z_{u,i}^{(t)} \odot \widetilde{e}_{u,i}^{(t)}, \tag{7}$$

where $W_{z1}, W_{r1}, W_{h1} \in \mathbb{R}^{2d \times d}$, $H_{z1}, H_{r1}, H_{h1} \in \mathbb{R}^{d \times d}$ are learnable parameter matrices, $\sigma(\cdot)$ is a sigmoid function, and $\bigodot$ represent element-wise multiplication. $z_{u,i}^{(t)} \in \mathbb{R}^d$ and $r_{u,i}^{(t)} \in \mathbb{R}^d$ denote the update gate and reset gate, respectively, controlling what information should be preserved and filtered during the fusion process. The final course representation $e_{u,i}^{(t)}$ is the tradeoff between the previous self-course representation $e_{u,i}^{(t-1)}$ and new filtered neighborhood information $\widetilde{e}_{u,i}^{(t)}$. The iteration step of aggregation and fusion starts at $t = 0$, and stops when it reaches a pre-defined depth T. Finally, we obtain the sequence representation $E_u^{(T)} = [e_{u,1}^{(T)}, \ldots, e_{u,l}^{(T)}]$.

4.2.2 Position-Aware User Learning Interest Representation Learning. After T-layer updating process, we have obtained each course's representation by incorporating its neighborhood courses. Then we consider how to fuse these course representations to infer the user's long-term learning interest representation. A straightforward solution is to use the mean pooling/sum method on the sequence representation. However, each course contributes differently when forming the overall learning interest representation. Intuitively, the more recent courses in the sequence are more representative of the user's current interest and thus should be allocated with higher importance. To this end, we propose a position-aware attention mechanism that incorporates position information and course representation to make a better inference of the user's learning interest representation. Specifically, for the sequence representation $E_u^{(T)} = [e_{u,1}^{(T)}, e_{u,2}^{(T)}, \ldots, e_{u,l}^{(T)}]$, we introduce an inversed position matrix $P = [p_l, p_{l-1}, \ldots, p_1]$ to describe the position information in the sequence, where $p_i \in \mathbb{R}^d$ denotes the position embedding corresponding to the position i. Note that position matrix P is reversed, i.e., the last course $v_{u,l}$ corresponds to the 1st position, and the penultimate course $v_{u,l-2}$ is in the 2nd position. Therefore, it allocates stable position embeddings for the last several courses and enables the model to memorize that positions toward the end are more important. Thus, we integrate the course representation $e_{u,i}^{(T)}$ and the corresponding position embedding p_{l-i+1} and obtain the fused representation $h_{u,i}$:

$$h_{u,i} = \tanh\left(W_1 \cdot Concat\left(e_{u,i}^{(T)}, p_{l-i+1}\right) + b_1\right). \tag{8}$$

Since the last course $v_{u,l}$ is a significant indicator of the user's next learning choice, we learn the importance of each course through a soft-attention mechanism with a combination of the last $h_{u,l}$:

$$\alpha_i = q^T\left(W_2 h_{u,l} + W_3 h_{u,i} + b_2\right), \tag{9}$$

where $W_1 \in \mathbb{R}^{2d \times d}$, $W_2, W_3 \in \mathbb{R}^{d \times d}$ and $b_2, q \in \mathbb{R}^d$ are learnable matrixes and vectors, respectively.

Finally, the user learning interest representation e_u^s inferred from the learning sequence can be obtained by a linear combination of its course representations:

$$s_u^s = \sum_{i=1}^{l} \alpha_i e_{u,i}^{(T)}. \tag{10}$$

In this way, the proposed UIE not only considers the user's personalized course learning transition pattern in the learning sequence but also effectively differentiates the contribution of each course with its chronological order.

4.3 Category-Level Transition Pattern Encoder

Although the proposed UIE is dedicatedly designed, however, the dynamic user interests and complicated course transition patterns still bring severe challenges. Naturally, resorting to additional auxiliary information about users' learning behaviors could assist in precisely capturing the user's learning interest. As seen in the data exploration, except for the course sequence which explicitly expresses the user's learning interest, the category transition patterns in both intra- and inter-category views also provide new clues to reveal it. This phenomenon benefits narrowing down the category range of the next candidate course and further improving the next course prediction, which is less considered in previous research. Therefore, in this section, we present how to design a category-level TPE to enhance the course recommendation.

Specifically, for the learning sequence $S_u = \{v_{u,1}, v_{u,2}, \dots, v_{u,l}\}$, the corresponding category sequence is constructed as $C_u = \{c_{u,1}, c_{u,2}, \dots, c_{u,l}\}$, where $c_{u,i}$ is the category of course $v_{u,i}$. Let $f_{u,i} \in \mathbb{R}^d$ represents the embedding of category $c_{u,i}$, and then obtain the category sequence embedding $F_u = \{f_{u,1}, f_{u,2}, \dots, f_{u,l}\}$. As above mentioned that the course category information is also conducive to indicating the user's learning interest, which has shown obvious short-term dependencies in intra- and inter-categories, such as the category *Design* to *Design* and *Design* to *Development*. GRU is a typical neural-based sequence model which specializes in dependency modeling and has been successfully applied in the field of natural language processing and session recommendation. Thus, we apply GRU to capture the transition patterns of the user's learning interest on the category level. Considering that the course category is closely bound up with the course content, we concatenate the course embedding $e_{u,i}$ and category embedding $f_{u,i}$ as input, i.e., $x_{u,i} = [e_{u,i}, f_{u,i}]$, to GRU to predict the next course category, which can be formulated as follows:

$$z_{u,i} = \sigma\left(W_{z2}\left[x_{u,i}, s^c_{u,i-1}\right]\right), \tag{11}$$

$$r_{u,i} = \sigma\left(W_{r2}\left[x_{u,i}, s^c_{u,i-1}\right]\right), \tag{12}$$

$$\widetilde{s}^c_{u,i} = \tanh\left(W_{h2}\left[x_{u,i}, r_{u,i} \odot s^c_{u,i-1}\right]\right), \tag{13}$$

$$s^c_{u,i} = \left(1 - z_{u,i}\right) \odot s^c_{u,i-1} + z_{u,i} \odot \widetilde{s}^c_{u,i}, \tag{14}$$

where $W_{z2}, W_{r2}, W_{h2} \in \mathbb{R}^{2d\times d}$ are learnable parameter matrices. $s^c_{u,i} \in \mathbb{R}^d$ is the hidden state, $z_{u,i} \in \mathbb{R}^d$ and $r_{u,i} \in \mathbb{R}^d$ denote the update and reset gate vectors, respectively. We take the category sequence representation $s^c_u = s^c_{u,l}$, denoting the next course category the user will take.

4.4 Fusion & Prediction Layer

Through the proposed two encoders, we have obtained the course-level and category-level user learning interest representations s^s_u and s^c_u, respectively. To better describe the user's learning interest, we combine the course-level and category-level representations $s_u = [s^s_u,\ s^c_u]$ as the user presentation. For each candidate course v_k as well as the category c_k, the corresponding course embedding can be denoted as $s_k = [e_k, f_k]$. Based on these, we use the dot product and softmax function to compute the user learning interest score $\widehat{y}_k$ for course k:

$$\widehat{y}_{uk} = softmax\left(s^T_u \cdot s_k\right), \tag{15}$$

where $\widehat{y}_i$ represents the probability that the course k is the next course that the user will learn. Then the cross-entropy loss function is defined to optimize the predictions:

$$\mathcal{L}_{main}\left(\widehat{y}\right) = -\sum_{u=1}^{M}\sum_{k=1}^{N} y_{uk}\log\left(\widehat{y}_{uk}\right) + \left(1 - y_{uk}\right)\log\left(1 - \widehat{y}_{uk}\right), \tag{16}$$

Table 4. Dataset Statistics

Statistics	Udemy	XueTangX
Users	130,019	35,705
Courses	23,581	895
Interactions	1,094,914	301,733
Categories	15	23
Average length	8.42	8.45

where y_{uk} denotes one hot encoding vector of the ground truth that user u will learn the course k given the historical learning sequence S_u.

Finally, the loss functions of the main course recommendation task as well as the auxiliary prerequisite pre-training task are joined to optimize the model parameters Θ:

$$\mathcal{L}_{total} = \mathcal{L}_{main} + \lambda_{pg}\mathcal{L}_{pg} + \lambda_2 \|\Theta\|_2 . \tag{17}$$

5 EXPERIMENTS

We conduct extensive experiments on two real-world datasets to validate the effectiveness of our proposed model compared with several state-of-the-art baselines under different conditions for course recommendation.

5.1 Experiment Setup

5.1.1 Datasets. We evaluate our model on two real-world datasets named Udemy[1] and XueTangX.[2] Udemy (Dessi et al., 2018) is an online learning and teaching platform aiming to improve learners' job-related skills. Within Udemy, any learner can choose free or paid courses provided by experts from different fields. XueTangX is a MOOC platform in China that has more than 3,000 high-quality courses from Tsinghua University, Massachusetts Institute of Technology, Stanford University, and other first-class universities. Both two datasets contain the users' learning records, including the user ID, course ID with category, and associated timestamp. To filter noisy data, we remove sequences with length 1 and courses appearing less than five times. After filtering, the detailed data statistics are shown in Table 4.

Following previous studies [28, 29], we use the leave-one-out strategy for evaluation. Specifically, for each user's learning sequence, we leave the last course as the testing data, the penultimate course as the validation data, and all the remaining data as the training data. We organize the course prerequisite graph from the training data and pre-train it with TransE model, and further utilize the pre-trained embeddings to initialize the courses and categories in two encoders.

5.1.2 Evaluation Metrics. For each user in the testing data, the model outputs a top-K recommendation list based on the predicted probability $\widehat{y}_{uk}$. To evaluate the effectiveness of the top-K recommendation of the model, We adopt two widely used metrics, HR@K (Hit Ratio) [33] and MRR@K (Mean Reciprocal Rank) [18]. By default, we set $K = 10$.

— HR@K measures the ratio of ground-truth courses appearing in the top-K recommendation list. The larger HR indicates the higher accuracy of the model. The metric is calculated as follows:

$$HR@K = \frac{NumberOfHits@K}{N}, \tag{18}$$

[1] https://www.udemy.com
[2] http://www.xuetangx.com

where $NumberOfHits@K$ counts the total number of ground-truth courses appearing in the top-K recommendation list. N represents the size of the testing set.

— MRR is the average reciprocal ranking of the ground-truth item in the top-K recommendation list. If the ground-truth item is not in the top-K list, then the value is set to 0. Specifically, MRR can be calculated as follows:

$$MRR@K = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i}. \tag{19}$$

Where $rank_i$ is the ranking position of the ground-truth item in the top-K recommendation list. The larger MRR is, the better performance the model achieves.

5.1.3 Baselines. To demonstrate the effectiveness of our proposed PCGNN model, we compare it with the three groups of recommendation methods, including traditional recommendation methods, RNN-based sequential recommendation methods, and GNN-based sequential recommendation methods.

Traditional recommendation methods merely leverage the user-item interactions, which can be used to evaluate the effectiveness of sequential modeling:

— **BPR [21]:** This method is a classical MF model using implicit feedback for recommendation with a pairwise loss.
— **LightGCN [8]:** This method is a simple yet effective GNN-based collaborative filtering recommendation model that simplifies the nonlinear transformation of GCN.

RNN-based sequential recommendation methods capture the sequential correlation and item transition pattern via RNN, which includes:

— **GRU4Rec [10]:** This method applies GRU to model sequences.
— **SASRec [15]:** This method utilizes a self-attention mechanism to identify which items are more relevant to represent the sequence.
— **NARM [16]:** This method explores a hybrid encoder framework with an attention mechanism to capture the global sequence behavior and local purpose.

GNN-based methods can capture the complex item transition patterns by modeling session graphs, which consist of:

— **SRGNN [29]:** The method transforms each session sequence into a session graph, and further employs GNN and traditional self-attention network to learn the sequence representation.
— **GCSAN [31]:** This method improves SRGNN with a multi-layer self-attention mechanism.
— **CASE4SR [18]:** This method incorporates an additional category transition graph to augment GNN-based sequential recommendation.

5.1.4 Parameter Settings. In the experiment, we use Pytorch, an open-source deep learning library, to implement our model. For a fair comparison, the hyper-parameters of all methods are set either by the suggestions of the original article or optimized by our experiments. For our model, the hyper-parameters are chosen by grid search as follows: we vary the embedding size d = {8,16,32,64,80,96,128}, the learning rate lr in {0.001,0.002,0.003,0.01,0.02,0.05}, the weight of PGP task λ_{pg} in {1,2,3,4,5,6}. Besides, we set the batch size $B = 2048$, and the L2 regularization term $\lambda_2 = 0.0001$. The iteration step of UIE module $T = 2$.

5.2 Overall Performance Comparison

The overall performance of all models on the two datasets are shown in Table 5 with the following observations and conclusions:

Table 5. The Overall Performance of All Methods on Two Datasets

Datasets	Udemy		XueTangX	
Methods	HR@10	MRR@10	HR@10	MRR@10
BPR	0.1302	0.0545	0.3881	0.1843
LightGCN	0.1364	0.0573	0.3808	0.1756
GRU4Rec	0.1676	0.0684	0.4254	0.2507
SASRec	0.1705	0.06	0.4094	0.1902
NARM	0.1703	0.0689	0.4074	0.1895
SRGNN	0.1635	0.0702	0.4281	0.2438
GCSAN	0.1652	0.0651	0.4227	0.2677
CASE4SR	0.1741	0.0777	0.4315	0.2846
PCGNN	**0.1842**	**0.0843**	**0.44**	**0.3026**
Impv.	**5.80%**	**8.50%**	**1.90%**	**6.30%**

(1) **Our PCGNN achieves the best results.** From the comparison, we observe that PCGNN consistently outperforms other baselines on two datasets in terms of two metrics with great improvements, i.e., 5.8% HR impv. For Udemy and 6.3% MRR impv. for XueTangX. This superior performance can be attributed to the effectiveness of the course-level and category-level user learning interest modeling in PCGNN which accurately captures the learner's fine-grained learning interest for personalized course recommendations. The further distinct contributions of the two parts are examined in Section 5.4.
(2) **Sequence pattern modeling does matter the performance.** From the comparison of traditional recommendation methods (BPR and LightGCN) with other sequential methods, we infer that without consideration of the chronological order of the courses is incapable of capturing the user's varying learning interests. Besides, generally speaking, GNN-based methods (e.g., SRGNN, CASE4SR) which transfer sequences into session graphs benefit capture the complex transition patterns between courses and thus obtain better performance than RNN-based methods (e.g., GRU4Rec, SASRec, and NARM).
(3) **Auxiliary category sequence task further improves the learning interest modeling.** Note that both PCGNN and CASE4SR leverage the category sequence as auxiliary signals, and achieve the best among all baselines. This phenomenon illustrates the necessity and importance of category transition patterns for user interest modeling. Moreover, there exists a significant difference between PCGNN and CASE4SR in exploiting the category sequence. CASER4SR directly constructs a category graph with the GGNN model, while PCGNN combines the course and category with GRU, which is proven to be more reasonable and effective to reveal the user's category-level learning interest.

5.3 Performance w.r.t. Sequence Length

To examine the robustness of our proposed PCGNN model to data sparsity issue which is ubiquitous in most recommender systems, we partition the users into three groups based on the learning sequence length, i.e., Short-group with range [2,10], Medium-group with range [11,20], and Long-group with range [20,). Then three models are independently trained and evaluated on the three datasets. For a better comparison, we select a set of methods with better performance including GRU4Rec, NARM, SRGNN, and CASE4SR as baselines. Due to article limits, we mainly report the results on Udemy which also shows a similar trend on XueTangX. The detailed results are shown in Table 6.

Table 6. The Model Performance w.r.t. Different Sequence Lengths on the Udemy Dataset

Sequence Length	Short-group		Medium-group		Long-group	
Methods	HR@10	MRR@10	HR@10	MRR@10	HR@10	MRR@10
GRU4Rec	0.1571	0.0652	0.0937	0.0347	0.0394	0.0133
NARM	0.1682	0.0693	0.0989	0.0358	0.0442	0.0136
SRGNN	0.1642	0.0762	0.0942	0.0371	0.0433	0.0145
CASE4SR	0.1719	0.0766	0.1032	0.0399	0.0474	0.0160
PCGNN	**0.1797**	**0.0832**	**0.1069**	**0.0454**	**0.0556**	**0.0171**
Impv.	4.50%	8.60%	3.60%	13.80%	14.70%	6.90%

Obviously, from the table, we observe that with the increase of the sequence length, the performance of all methods in terms of the two metrics drops to varying degrees. This phenomenon is reasonable since users in the Short Group are new learners with clear learning intentions; while users in the Long Group belong to skilled learners and tend to learn courses according to their varying interests, presenting diverse course transition patterns that are difficult to model. Among them, PCGNN outperforms other baselines with large gaps in all three groups, showing its potential in addressing the data sparsity issue in the course recommendation scenario. We ascribe this superiority to the overall learning principles hidden in the course prerequisite graph and the additional guidance from the category sequence modeling. Furthermore, CASE4SR consistently performs better than SRGNN and NARM, which also indicates the importance of the category sequence. Merely relying on the course sequence, GRU4Rec fails to capture the user's learning interest transition patterns and performs the worst.

5.4 Ablation Study of Model Components

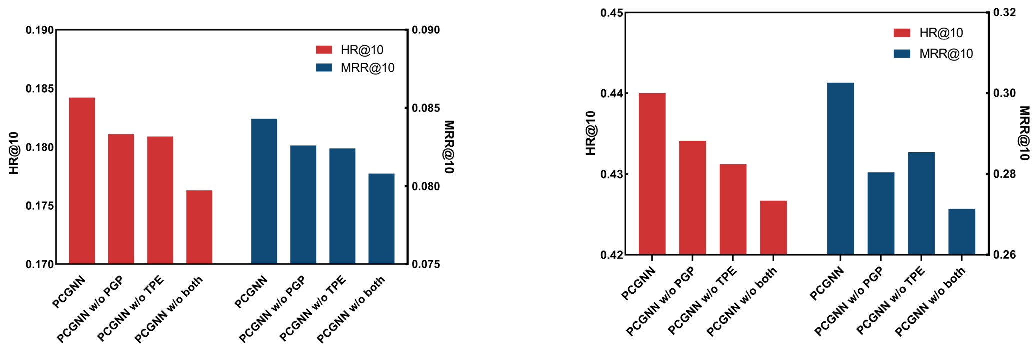
In this subsection, we conduct ablation experiments to demonstrate the effectiveness and necessity of each proposed module in PCGNN from (i) different components, and (ii) aggregation operation of the two representations.

Effect of different components. First, we examine the effectiveness of different components in PCGNN, including the Course PGP, Course-level UIE, and Category-level TPE. We iteratively remove different components of PCGNN and obtain three variants:

— *PCGNN w/o PGP*: denotes the removal of the Course PGP module of PCGNN;
— *PCGNN w/o TPE*: removes the Category-level TPE of PCGNN;
— *PCGNN w/o both*: discards both two components of PCGNN and leaves alone the Course-level UIE.

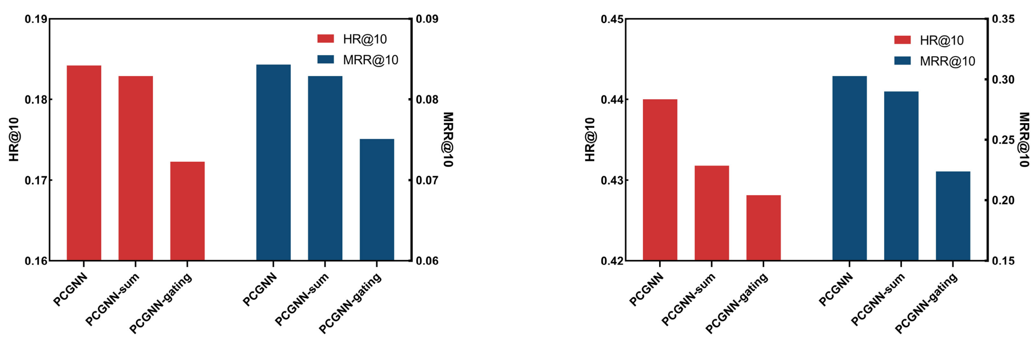
We report the evaluation results on the two datasets in Figure 6. Overall speaking, any removal of a component incurs performance deterioration compared with the original PCGNN model, demonstrating the effectiveness of each designed component for course recommendation. Interestingly, we observe that the performance drop ratio of *PCGNN w/o TPE* is comparable to *PCGNN w/o PGP*, suggesting that both two components contribute equally to the model performance. When the two designed components are removed, *PCGNN w/o both* further degrades in contrast to *PCGNN w/o TPE* and *PCGNN w/o PGP*, indicating the two components provide complementary information to identify the user's learning interest.

Impact of aggregation operations. Since the course-level learning sequence representation and category-level transition pattern representation jointly depict the user's overall learning interest, it is meaningful to study how to effectively combine the two representations to achieve better performance. Therefore, we further compare the concatenation operation in PCGNN with two different aggregation operations, i.e., sum pooling and gating mechanisms.



(a) Udemy (b) XueTangX

Fig. 6. The ablation performance of different components on Udemy and XueTangX.



(a) Udemy (b) XueTangX

Fig. 7. The ablation performance of different aggregation operations on Udemy and XueTangX.

— *PCGNN-sum:* for the sum pooling, we directly add the course-level sequence representation s_u^s and category-level representation s_u^c:

$$s_u = s_u^s + s_u^c. \tag{20}$$

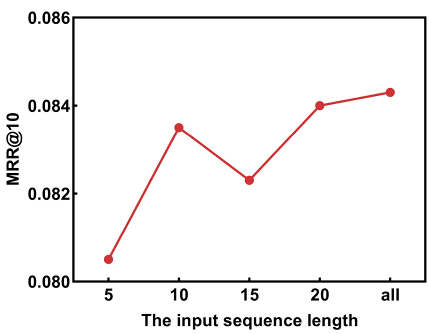
— *PCGNN-gating*: for the gating mechanism, we utilize a gating mechanism to balance the information between the two representation vectors:

$$r_u = \sigma\left(W_1 s_u^s + W_2 s_u^c\right), \tag{21}$$

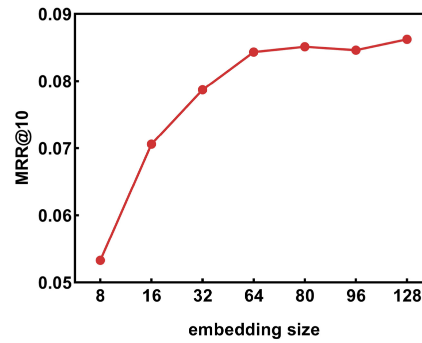
$$s_u = r_u s_u^s + (1 - r_u)\, s_u^c, \tag{22}$$

where σ is the sigmoid function, and W_1, $W_2 \in \mathbb{R}^d$ are the weight matrices. r_u represents the importance of two vectors.

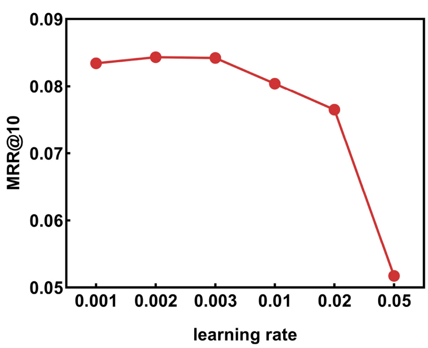
Figure 7 shows the performance of different aggregation operations on the two datasets. From the figure, we can observe that PCGNN with concatenation operation performs the best on both two datasets in terms of the two metrics. In contrast, the *PCGNN-sum* performs worse than PCGNN but better than *PCGNN-gating*. This is probably because the sum pooling confuses the complementary information in the two representations, while too many parameters in the gating mechanism bring about overfitting.



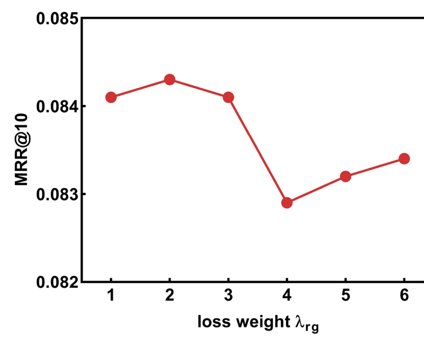
(a) Input sequence length



(b) Embedding size



(c) Learning rate



(d) Loss weight of the graph pre-training

Fig. 8. The performance of PCGNN w.r.t. different hyper-parameters.

5.5 Parameter Sensitivity

We further investigate the model's performance *w.r.t.* different hyper-parameters including the input sequence length l, embedding size d, learning rate lr, and PGP loss weight λ_{rg}. We omit other hyper-parameters since they have less impact on the model's performance. Due to space limitations, we mainly show the MRR results on the Udemy dataset in Figure 8.

Impact of input sequence l. The input sequence length determines how much information is utilized for sequential modeling. We vary the length of each user's learning sequence in the range of {5, 10, 15, 20, all} and report the results in Figure 8(a). As shown in the figure that with the increase of the sequence length, the performance of our model displays a gradually increasing trend, indicating the superiority of our model in being capable of capturing long-term learning interest. Therefore, we take the whole sequence to PCGNN for sufficient learning interest modeling.

Impact of embedding size d. The embedding size greatly influences the model's expressiveness. We empirically set the embedding size as 8, 16, 32, 64, 80, 96, and 128 and show the performance trend in Figure 8(b). From the figure, we can observe that with the embedding size increases, PCGNN shows an increasing trend and eventually stabilizes when $d \geq 64$. Considering the computational complexity of the high embedding size, we set $d = 64$.

Impact of learning rate lr. Besides, we also examine the impact of the learning rate on the model's performance. We gradually increase the learning rate from 0.001 to 0.05 and display the results in Figure 8(c). From the figure, we infer that a too-large or small learning rate will deteriorate the training effect. To this end, we set $lr = 0.002$.

Impact of pre-training loss weight λ_{rg}. The PGP task aims at capturing the course prerequisite relationships and lays a foundation for complex learning interest modeling. λ_{rg} controls the tradeoff between the pre-training task and the main recommendation task in the loss function. Figure 8(d) shows the results when λ_{rg} increases from 1 to 6. We can observe that too small λ_{rg} is insufficient to restrain the course prerequisite relationships in the training process, while too large λ_{rg} makes the model can not focus on minimizing the recommendation loss. Therefore, $\lambda_{rg} = 2$ is a suitable weight to balance the two tasks.

6 CONCLUSIONS

In this article, we propose a Prerequisite-enhanced Category-aware Graph Neural Network model for accurate course recommendation, named PCGNN, which jointly considers the course-level prerequisite relationships and category-level learning interest transition patterns to capture the user's complex learning interest. Specifically, the course prerequisite graph is firstly constructed from the massive learning records and modeled through a TransE model to seize the intrinsic course relationships. Besides, the course category sequence, which provides a fine-grained clue to the user's learning interest, is also combined with the course sequential modeling for future course enrollment. Empirical experiments on two real-world datasets have demonstrated the effectiveness and superiority of PCGNN compared with other state-of-the-art methods under different sparsity conditions for accurate course recommendation.

Currently, we mainly focus on the course and category transition pattern modeling to boost course recommendation performance. Note that different learners show diverse learning states and trajectories even for the same learning intention. Therefore, in the future, we may explore fine-grained user learning state modeling and inference to achieve personalized course recommendations.

ACKNOWLEDGMENTS

We thank all editors and anonymous reviewers for their valuable and constructive comments that improve our article.

REFERENCES

[1] Sunita B. Aher and L. M. R. J. Lobo. 2013. Combination of machine learning algorithms for recommendation of courses in E-Learning System based on historical data. *Knowledge-Based Systems* 51 (2013), 1–14.

[2] R. Begleiter, R. El-Yaniv, and G. Yona. 2004. On prediction using variable order Markov models. *Journal of Artificial Intelligence Research* 22 (2004), 385–421.

[3] Fatiha Bousbahi and Henda Chorfi. 2015. MOOC-Rec: A case-based recommender system for MOOCs. *Procedia-Social and Behavioral Sciences* 195 (2015), 1813–1822.

[4] Xiaojuan Chen and Huiwen Deng. 2021. Research on personalized recommendation methods for online video learning resources. *Applied Sciences* 11, 2 (2021), 804.

[5] J. Fan, Y. C. Jiang, Y. Z. Liu, and Y. H. Zhou. 2022. Interpretable MOOC recommendation: A multi-attention network for personalized learning behavior analysis. *Internet Research* 32, 2 (2022), 588–605.

[6] Weibo Gao, Qi Liu, Zhenya Huang, Yu Yin, Haoyang Bi, Mu-Chun Wang, Jianhui Ma, Shijin Wang, and Yu Su. 2021. RCD: Relation map driven cognitive diagnosis for intelligent education systems. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 501–510.

[7] R. N. He, W. C. Kang, and J. McAuley. 2017. Translation-based Recommendation. In *Proceedings of the 11th ACM Conference on Recommender Systems (Recsys'17).* 161–169.

[8] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.* 639–648.

[9] Xiangnan He, Zhankui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, and Tat-Seng Chua. 2018. NAIS: Neural attentive item similarity model for recommendation. *IEEE Transactions on Knowledge and Data Engineering* 30, 12 (2018), 2354–2366.

[10] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939 (2015).

[11] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk. 2016. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems (Recsys'16).* 241–248.

[12] Xiaoxian Huang, Yong Tang, Rong Qu, Chunying Li, Chengzhe Yuan, Saimei Sun, and Bixia Xu. 2018. Course recommendation model in academic social networks based on association rules and multi-similarity. In *Proceedings of the IEEE 22nd International Conference on Computer Supported Cooperative Work in Design ((CSCWD)).* 277–282.

[13] Xia Jing and Jie Tang. 2017. Guess you like: Course recommendation in MOOCs. In *Proceedings of the International Conference on Web Intelligence.* 783–789.

[14] Santosh Kabbur, Xia Ning, and George Karypis. 2014. FISM: Factored item similarity models for top-n recommender systems. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 659–667.

[15] W. C. Kang and J. McAuley. 2018. Self-attentive sequential recommendation. In *Proceedings of the 2018 IEEE International Conference on Data Mining.* 197–206.

[16] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management.* 1419–1428.

[17] Y. G. Lin, S. B. Feng, F. Lin, W. H. Zeng, Y. Liu, and P. C. Wu. 2021. Adaptive course recommendation in MOOCs. *Knowledge-Based Systems* 224 (2021), 1–11.

[18] L. Liu, L. Wang, and T. Lian. 2021. CaSe4SR: Using category sequence graph to augment session-based recommendation. *Knowledge-Based Systems* 212 (2021), 1–12.

[19] Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Hui Xiong, Yu Su, and Guoping Hu. 2021. EKT: Exercise-aware knowledge tracing for student performance prediction. *IEEE Transactions on Knowledge and Data Engineering* 33, 1 (2021), 100–115.

[20] Yitong Pang, Lingfei Wu, Qi Shen, Yiming Zhang, Zhihua Wei, Fangli Xu, Ethan Chang, Bo Long, and Jian Pei. 2022. Heterogeneous global graph neural networks for personalized session-based recommendation. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining.* 775–783.

[21] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618 (2012).

[22] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web.* 811–820.

[23] G. Shani, D. Heckerman, and R. I. Brafman. 2005. An MDP-based recommender system. *Journal of Machine Learning Research* 6, 9 (2005), 1265–1295.

[24] F. Sun, J. Liu, J. Wu, C. H. Pei, X. Lin, W. W. Ou, and P. Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management.* 1441–1450.

[25] Xuetao Tian and Feng Liu. 2021. Capacity tracing-enhanced course recommendation in MOOCs. *IEEE Transactions on Learning Technologies* 14, 3 (2021), 313–321.

[26] Chao Wang, Hengshu Zhu, Chen Zhu, Xi Zhang, Enhong Chen, and Hui Xiong. 2020. Personalized employee training course recommendation with career development awareness. In *Proceedings of the Web Conference.* 1648–1659.

[27] Wen Wang, Wei Zhang, Shukai Liu, Qi Liu, Bo Zhang, Leyu Lin, and Hongyuan Zha. 2020. Beyond clicks: Modeling multi-relational item graph for session-based target behavior prediction. In *Proceedings of the Web Conference.* 3056–3062.

[28] Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xian-Ling Mao, and Minghui Qiu. 2020. Global context enhanced graph neural networks for session-based recommendation. In *Proceedings of the 43rd international ACM SIGIR Conference on Research and Development in Information Retrieval.* 169–178.

[29] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence.* 346–353.

[30] Wenbin Wu, Bixi Wang, Yan Liu, Lirong Yin, and Wenfeng Zheng. 2020. Higher education online courses personalized recommendation algorithm based on score and attributes. *Journal of Physics: Conference Series* 1673, 1 (2020), 1–8.

[31] C. F. Xu, P. P. Zhao, Y. C. Liu, V. S. Sheng, J. J. Xu, F. Z. Zhuang, J. H. Fang, and X. F. Zhou. 2019. Graph contextualized self-attention network for session-based recommendation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence.* 3940–3946.

[32] H. Zhang, T. Huang, Z. H. Lv, S. Y. Liu, and H. Yang. 2019. MOOCRC: A highly accurate resource recommendation model for use in MOOC environments. *Mobile Networks and Applications* 24, 1 (2019), 34–46.

[33] Z. Wang, J. Zhang, J. Feng, and Z. Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence.* 28, 1 (2014).

[34] Zhongying Zhao, Yonghao Yang, Chao Li, and Liqiang Nie. 2020. GuessUNeed: Recommending courses via neural attention network and course prerequisite relation embeddings. *ACM Transactions on Multimedia Computing, Communications, and Applications* 16, 4 (2020), 1–17.

[35] Y. W. Zhou, C. Q. Huang, Q. T. Hu, J. Zhu, and Y. Tang. 2018. Personalized learning full-path recommendation model based on LSTM neural networks. *Information Sciences* 444 (2018), 135–152.

Received 31 July 2023; accepted 22 January 2024