

RESEARCH

Open Access



Enhancing academic performance prediction with temporal graph networks for massive open online courses

Qionghao Huang^{1,2*} and Jili Chen^{1,2}

*Correspondence:
qhhuang@m.scnu.edu.cn

¹ Zhejiang Key Laboratory of Intelligent Education Technology and Application, Zhejiang Normal University, Yinbin Ave., Jinhua 321004, Zhejiang, China

² College of Education, Zhejiang Normal University, Yinbin Ave., Jinhua 321004, Zhejiang, China

Abstract

Educational big data significantly impacts education, and Massive Open Online Courses (MOOCs), a crucial learning approach, have evolved to be more intelligent with these technologies. Deep neural networks have significantly advanced the crucial task within MOOCs, predicting student academic performance. However, most deep learning-based methods usually ignore the temporal information and interaction behaviors during the learning activities, which can effectively enhance the model's predictive accuracy. To tackle this, we formulate the learning processes of e-learning students as dynamic temporal graphs to encode the temporal information and interaction behaviors during their studying. We propose a novel academic performance prediction model (APP-TGN) based on temporal graph neural networks. Specifically, in APP-TGN, a dynamic graph is constructed from online learning activity logs. A temporal graph network with low-high filters learns potential academic performance variations encoded in dynamic graphs. Furthermore, a global sampling module is developed to mitigate the problem of false correlations in deep learning-based models. Finally, multi-head attention is utilized for predicting academic outcomes. Extensive experiments are conducted on a well-known public dataset. The experimental results indicate that APP-TGN significantly surpasses existing methods and demonstrates excellent potential in automated feedback and personalized learning.

Keywords: Artificial intelligence, Temporal graph networks, Academic performance prediction

Introduction

With advancements in deep learning technologies, big data, and artificial intelligence (AI) technologies are now widespread in various domains, such as smart grids [1], telecommunications [2], and so on. Educational big data also profoundly influences and reshapes academic research and industrial applications in education [3, 4]. Massive Open Online Courses (MOOCs) [5], generating a large amount of diversified learning behavior data, has been a research hotspot in educational big data and *AI + Education* [6, 7]. Student academic performance prediction, as a fundamental technique in realizing intelligent educational applications [8], has received more and more attention in MOOCs [9]. Predicting student performance is a process that involves

estimating how students will fare in future evaluations or exams. This process is crucial in identifying students at risk of failing or dropping out, enabling timely intervention and support. Such a process holds significant importance in the context of massive open online courses [10].

Many research efforts have been devoted to predicting students' academic performance with machine learning techniques. For instance, traditional machine learning techniques successfully applied to academic performance prediction, e.g., logistic regression, random forest, artificial neural network, support vector machine [11, 12]. Deep neural network-based methods also make significant progress, e.g., recurrent neural networks [13, 14], convolutional neural networks [15], attention networks [8, 16]. However, most existing academic performance prediction methods exploit learning behavior data with simple feature engineering, e.g., using a statistic figure (number of occurrences) to denote the feature of a specific learning activity. These settings may result in severe value information loss due to inappropriate data structures. To encode the learning behavior data with graph structures [17], the most expressive data structure, which can retain valuable clues for performance predictions [18]. Furthermore, many kinds of research show that sequential patterns of learning behaviors or interaction activities can exhibit the academic states of students [14, 19, 20]. Thus, encoding the online learning behavior data in graph structures with a temporal property may better retain the value learning cues for their academic performance predictions. This work will demonstrate that this temporal graph structure is vital for academic performance prediction. Nevertheless, finding a suitable graph structure to encode students' learning cues and corresponding processing techniques remains challenging in the domain.

To bridge this gap, a novel model, APP-TGN, utilizing temporal graph neural networks, is introduced to predict academic performance for problem-solving. Specifically, within APP-TGN, a dynamic graph is constructed from the online learning activity logs. The generated graph is forwarded to a temporal graph network with low-high filters to learn potential academic performance variations encoded in dynamic graphs. Furthermore, a global sampling module is developed to mitigate the problem of false correlations in deep learning-based models. Finally, the learned representations from global sampling and local processing (with TGN) are passed through a multi-head attention module, predicting academic performances. The proposed approach's utility is assessed through comprehensive experimentation using the widely recognized public dataset, OULA [21], derived from a practical educational application. Specifically, the empirical study seeks to address three research questions: **(i)** *How does the proposed APP-TGN perform when predicting student academic performance in terms of accuracy, F1-score, and recall?* **(ii)** *What is the improvement in early predicting at-risk students when using APP-TGN against other state-of-the-art methods?* **(iii)** *What contribution does each proposed component of APP-TCN make to the final prediction performance in terms of accuracy?* The experimental results indicate that the proposed APP-TGN significantly surpasses existing methods and holds great potential for automated feedback and personalized learning in practical educational applications. Ablation studies also highlight the superiority and value of the proposed techniques within APP-TGN.

The main technical contributions of the paper are summarized as follows:

- A novel framework for predicting academic performance is introduced, utilizing temporal graph networks and local and global sampling techniques. This framework leverages temporal information and interaction behaviors to achieve high prediction accuracy in the model.
- An efficient temporal graph neural network with low-high filters is designed to deal with temporal-evolving dynamic graphs formed by complex learning interaction activities.
- To the best of our knowledge, this paper is the first work to formulate the academic performance prediction tasks as the problem of classifying temporal dynamic graphs. Furthermore, a data bias deduction module is also developed in APP-TGN to mitigate the issue of false correlations in deep learning-based models.

Literature review

Academic performance predictions

As an important research task in intelligent education, academic performance prediction has attracted the attention of many researchers. The initial discussion is dedicated to exploring research works that utilize traditional machine learning techniques.

Methods with traditional machine learning Academic performance prediction with traditional machine learning has been investigated for decades [22]. Three logistic regression models were developed by Marbouti et al. to pinpoint students who were at risk in the first grade engineering curriculum. These models were applied at three crucial junctures throughout the semester, and the findings underscored the significance of devising a prediction model tailored to a specific curriculum [11]. Ren et al. formulated a linear multiple regression approach tailored to individual students to forecast their academic performance in the curriculum. This was achieved by monitoring student participation in MOOCs. The approach effectively highlighted critical aspects of the student's learning behaviors and studied habits [23]. Chui and his team introduced a model known as the Reduced Training Vector-based Support Vector Machine (RTV-SVM) for identifying students who are marginal or at risk. By minimizing the number of training vectors, this model effectively cuts down the duration of training while maintaining accuracy [24]. To find at-risk students at an early stage and promote the realization of pedagogical and economic goal outcomes, Coussement et al. proposed a logit leaf model (LLM). They visualized it to balance predictive performance and comprehensibility, effectively improving the prediction of student dropout [25]. Riestra et al. utilized five algorithms, decision tree, naive Bayes, logical regression, multi-layer perceptron, and support vector machine, to anticipate student performance in the early stages of a course, based on an analysis of LMS log information available at the time of prediction. In addition, they employed a clustering algorithm to examine various patterns of cluster interaction [26]. Turabieh et al. introduced a method that enhances the Harris Hawkes optimization (HHO) approach. This method addresses the issue of premature convergence by managing population diversity. They also employed the k-nearest neighbor (kNN) method as a strategy for clustering, which allowed them to monitor the performance of HHO in adjusting population diversity [27]. In their work, Mubarak et al. put forward Sequential Logistic Regression along with Input Output Hidden Markov Model (IOHMM) for scrutinizing student learning behavior. This approach proves effective in

pinpointing students who are at risk of discontinuing their studies [28]. A model based on genetic programming was developed by Jiao et al. for forecasting student academic performance. This model demonstrated robust performance compared to conventional AI methods such as ANN and SVM [29]. In summary, traditional machine learning algorithms have a limited capacity for feature learning [30], which can hinder their ability to model students' complex learning processes accurately.

Methods with deep neural networks Research on applying deep neural networks has become increasingly popular in recent years [31, 32]. Yang et al. proposed the 1-channel & 3-channel learning image recognition based on convolutional neural networks for transforming students' curriculum participation into images for predictive analysis [15]. Giannakas et al. introduced a Deep Neural Network framework with two hidden layers in software engineering. This framework was designed to predict teams' performance early and demonstrated superior performance compared to traditional methods [33]. It was specifically tailored to handle two-category classification tasks. Wang and colleagues presented AS-SAN, Adaptive Sparse Self-Attention Network, which predicts the fine-grained performance of students in online courses [8]. Karimi et al. constructed a knowledge map using the DOPE, the Deep Online Performance Evaluation method. They employed recurrent neural networks for encoding sequence learning, which aids in predicting student performance in curriculum [34]. Waheed et al. utilized deep artificial neural networks in virtual learning environments for early intervention with at-risk students. This approach, which extracted features from clickstream data, outperformed baseline models such as logistic regression or support vector machines [35]. Du et al. introduced a comprehensive model that leverages Latent Variation Auto Encoder (LVAE) and Deep Neural Network (DNN) to address imbalances in education datasets. This approach enhances the model's capacity for early identification of students at risk [36]. Leveraging the growing popularity of graph neural networks [31], a novel pipeline, MTGNN [18], has been developed for predicting student performance. This innovative approach utilizes multi-topology graph neural networks, capitalizing on graph structures to mirror student relationships. Sun et al. [37] propose an adversarial reinforcement learning method for time-relevant scoring systems. They aim to optimize student scores within a limited time while minimizing detection risk. The attacking problem is formulated as a Markov decision process, and a deep Q-network is used for policy learning. Li et al. introduced a unique method, MVHGNN, for predicting students' academic performance [38]. This approach utilizes hypergraphs, meta-paths, and a CAT module to establish high-order relations between students and determine the weight of various behaviors. Despite their effectiveness, these models do not incorporate temporal learning process information in simulating learning performance, indicating potential areas for enhancement.

Graph neural networks in educational applications

Graph Neural Networks (GNNs) have garnered significant interest recently due to their exceptional ability to extract information from non-Euclidean spaces [39]. As a versatile tool compatible with various learning paradigms, such as graph prompt learning [40, 41], GNNs have been widely applied in a range of domains, including natural language processing, recommendation systems, and materials science [42–44]. In

line with the advancements in intelligent education, GNNs have also made their mark in the educational sector.

Cognitive diagnosis For instance, cognitive diagnosis, a fundamental aspect of intelligent education, assesses a student's grasp of specific knowledge areas [45]. Gao and colleagues introduced a unique framework for Cognitive Diagnosis driven by Relation maps (RCD), based on the interplay among students, exercises, and concepts. This framework successfully integrates both structural and interactive relationships [46]. Zhang et al. introduced a graph-based approach to knowledge tracing for cognitive diagnosis, known as GKT-CD [47]. They utilized Gated-GNN within GKT-CD to monitor students' knowledge records and dynamically ascertain their knowledge mastery abilities. Mao et al. proposed an approach for cognitive diagnosis that is aware of learning behavior (LCD). This method employs GCN to distill features from exercises and videos, thereby enhancing the depiction of students' knowledge proficiency [48]. The graph-based Cognitive Diagnosis model (GCDM), proposed by Su et al. facilitates the extraction of interactions between students, skills, and questions from heterogeneous cognitive graphs [49]. It also uncovers potential higher-order relations between these entities. The ICD, a cognitive diagnostic model proposed by Qi et al. uses three layers of neural networks to model the influence of exercises on concepts, the interaction between concepts, and the influence of concepts on exercises, aiming to address the interaction among knowledge concepts and the quantitative relation between exercises and concepts [50]. These models have shown the comparable capacity of graph neural networks in modeling the complex learning interaction among students.

Knowledge tracing Knowledge tracing is another important task in intelligent education, which aims to judge students' knowledge states by tracing their historical learning [51, 52]. In the work of Nakagawa et al., a Graph Neural Network was utilized for the first time to transform knowledge structures and apply graph networks for interactive feature extraction, leading to the creation of a unique approach to knowledge tracing known as GKT [53]. In the study by Yang et al., a unique approach was introduced, known as Graph-based Interaction Knowledge Tracing (GIKT). This approach leveraged a graph convolution network, allowing it to discern the correlation between questions and skills [54]. Tong et al. introduced a hierarchical graph knowledge tracing approach, HGKT, was introduced. This approach involved the construction of a hierarchical exercise graph, effectively capturing the dependencies in exercise learning [55]. Song et al. introduced a Joint graph convolutional network-based deep Knowledge Tracing (JKT) system that connects exercises across different concepts, grasps high-level semantic details, and enhances the model's interpretability [56]. Wu et al. introduced a session graph-based knowledge tracing (SGKT) that captures dynamic graphs through student interactions during a session and mimics the student response process. Additionally, they utilized a gated graph neural network to discern the knowledge states of students [57]. A Bi-Graph Contrastive Learning-based Knowledge Tracing (Bi-CLKT) model was proposed to obtain better concept representation through contrastive learning [58]. Some models with self-supervised methods and graph neural networks are also investigated [59, 60]. These studies highlight the importance of simulating complex interactions during learning to improve model prediction performance.

Other educational applications Graph neural networks are also widely used in other intelligent education fields [61]. Ying et al. introduced an efficient Graph Convolutional Network that produces node embeddings using random walks and graph convolution techniques [62], and this approach has demonstrated outstanding performance in large-scale network recommendation systems. To counter cold start plus data sparsity issues in recommender systems based on collaborative filtering, Wang et al. introduced a Knowledge Graph Convolutional Network [63]. This network adeptly identifies item correlations by exploring attributes linked in knowledge graphs. In addressing costliness plus rigidity in conventional Automatic Short Answer Grading (ASAG) tasks, Tan et al. employed a two-layer graph convolutional network, transforming a heterogeneous graph representing student responses, effectively resolving these issues [64]. Agarwal et al. proposed a Multi-Relational Graph Transformer (MitiGaTe) to mine the structural context of the sentence and achieved remarkable performance on the ASAG task [65]. Li et al. used interactive information to model the relationship between students and questions. They proposed a GNN model named R2GCN, which can be applied to heterogeneous networks to predict students' performance in interactive online question banks [66]. Li et al. leveraged interactive data for mapping relationships between students and questions, proposing an R2GCN GNN variant. This variant, applicable on heterogeneous networks, forecasts student performance for interactive online question banks [66]. A GNN model named R2GCN was proposed to model the relationship between students and questions using interactive information, this model can be applied to heterogeneous networks to predict student performance in interactive online question banks [67]. Asadi et al. suggest using graph neural networks to model irregular multivariate time series, which can achieve accuracy comparable or superior to hand-crafted features when applied to raw time series click streams [20]. These models demonstrate the promising performance of graph neural networks in these applications.

Methodology

An academic performance prediction model (APP-TGN) based on a revised low-high filtering temporal graph network is proposed in this section. The proposed APP-TGN considers temporal information and interaction behaviors to enhance the performance of model predictions. Furthermore, a data bias deduction module with global sampling techniques is developed to mitigate the problem of false correlations in deep learning-based models. The section introduces the details of the proposed APP-TGN. Firstly, a brief introduction of the framework of APP-TGN is presented, followed by an explanation of the different components of APP-TGN.

The framework of APP-TGN

Figure 1 illustrates the architecture of our solution with APP-TGN. It mainly consists of five main components: *Data Collection & Pre-processing*, *Dynamic Graph Construction*, *Global Sampling Module*, *Low-High Filtering Temporal Graph Networks(LHFTGN)*, *Academic Performance Representation & Prediction*.

Procedures of APP-TGN Data Collection & Pre-processing includes attribution selection, data cleaning, and data transformation. With the pre-processed data from online learning systems, a dynamic graph construction method is presented to provide

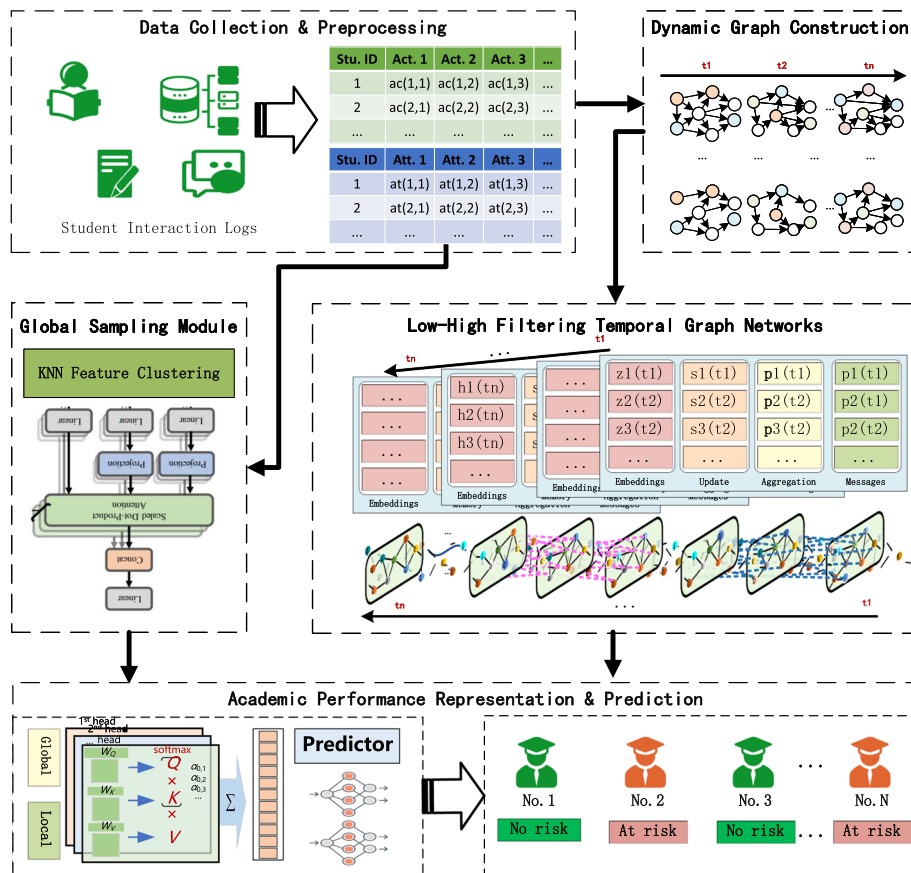


Fig. 1 The framework of the proposed APP-TGN for academic performance prediction

temporal graphs as the input for LHFTGN in *Dynamic Graph Construction*. After that, a revised temporal graph neural network with low-high filtering operators is applied to the generated dynamic graphs, from which a local representation of the academic performance is learned for the candid student. Meanwhile, a global representation of the group cognition is also obtained from *Global Sampling Module*. The local and global representations are concatenated and forwarded to a multi-head attention module to learn an unbiased academic performance representation. With an MLP-based classifier, the academic performances are predicted from the learned representations of these candidate students.

Data cleaning and pre-processing

To perform a training or prediction task for APP-TGN, we need to prepare well-format data from the interaction logs of learning management systems (LMS) to fulfill the requirement of APP-TGN through data cleaning and pre-processing. Usually, the data collection and pre-processing have several essential steps to obtain the desirable formatted data, such as attribute selection, data cleaning, data transformation, etc. Attribute selection refers to choosing a suitable subset of data to achieve better performance on a specific task, as there are many attribute features from the logs of LMS, not all of them can contribute to the model’s performance. Data cleaning is

to fix or remove incomplete or unreasonable data to produce a qualified dataset for model training or testing. More importantly, the data format or type may not fulfill the requirement of the model inputs. Some data transformation techniques are often employed to get the exact data types or structures for specific tasks. From Fig. 1, we can see that the input for APP-TGN can be divided into parts: Ones are used for generating dynamic graphs, and the others are forwarded to the global sampling module.

Data preparation for dynamic graph construction This paper mainly uses temporal dynamic graphs to encode the temporal information and learning behaviors to facilitate academic performance prediction. Thus, we need to prepare the candidate data to generate dynamic graphs. To generate a graph from the raw log data, the key is to determine the types of nodes and edges. As the target graph has a temporal property, we choose online activities as the nodes $V = \{v_1, v_2, \dots, v_{N_v}\}$, where v_i denotes the i th type of learning activities. The type of edges $E = \{e_1, e_2, \dots\}$ are usually the possible interactions between these nodes. We use the notation $ac(i, 1)$ to denote the required data to generate a dynamic graph. $ac(i, 1)$ represents a data unit from the sequence of learning activity logs of learner l_i . A sequence of learning activities for the learner l_i can be formulated by Eq. (1).

$$Ac(l_i) = \{ac(i, 1), ac(i, 2), \dots, ac(i, N_{aci})\}, \quad (1)$$

$$Fg(L) = \{Ac(l_1), Ac(l_2), \dots, Ac(l_M)\}, \quad (2)$$

where $L = \{l_1, l_2, \dots, l_m\}$, $Fg(L)$ represents a collection activity log data for a collection of learners L (with M learners), N_{aci} denotes the length of an activity log for the learner l_i . The following subsection will detail how these interaction activity logs are converted into dynamic graphs.

Data preparation for global sampling module A global sampling technique is applied in APP-TGN to mitigate the problem of false correlations in deep learning-based models. To achieve this goal, we must select the proper attributes to participate in the global sampling process. We use the notation $at(i, 1)$ to denote the i th chosen attribute (e.g., Gender, Region, Disability, Highest_education, etc.) from learning management systems. $at(i, 1)$ can be real-valued scalar or integer numbers obtained by a one-hot or multi-hot encoding method. Thus a record for the learner l_i can be formulated as follows:

$$At(l_i) = \{at(i, 1), at(i, 2), \dots, at(i, N_{ati})\}, \quad (3)$$

$$Fa(L) = \{At(l_1), At(l_2), \dots, At(l_M)\}, \quad (4)$$

where $Fa(L)$ represents selected attribute feature records for a collection of learners L (with M learners), N_{at} denotes that we choose N_{at} attribute features for the global sampling. Specifically, $Fa(L)$ is generated only from the training dataset, not all the raw data from LMS, through which the problem of predicting the current states with possible future information can be avoided.

Dynamic graph construction

The subsection details how to use the data from Eq. (2) to construct the dynamic temporal graphs as the input for low-high filtering temporal graph networks in APP-TGN. Temporal graphs are a kind of dynamic graphs that are temporally changing with node or edge events. In our setting, as mentioned in Eq. (1) and (2), we use a sequence of online learning activities to generate a temporal graph \mathcal{G} , the temporal graph can be formulated as follows:

$$\mathcal{G}(T) = \{\mathcal{V}(T), \mathcal{E}(T)\}, \quad (5)$$

$$\mathcal{V}(T) = \{i : \exists v_i(t) \in \Omega, t \in T\}, \quad (6)$$

$$\mathcal{E}(T) = \{(i, j) : \exists e_{ij}(t) \in \Omega, t \in T\}, \quad (7)$$

$$\Omega = \{x(t_1), x(t_2), \dots\}, \quad 0 \leq t_1 \leq t_2 \leq \dots, \quad (8)$$

where $x(t_i)$ denotes a node-wise or interaction event in the sequence of online learning activities Υ . A node-wise event $v_i(t)$ is an online learning activity from a collection of candidate online learning activities V . An interaction event is a directed temporal edge $e_{ij}(t)$ between node v_i (source) and node v_j (target), usually denoting the transition from the learning activity v_i to the learning activity v_j . $\mathcal{N}_i(T) = \{j : (i, j) \in \Omega(T)\}$ refers to the neighborhood of node $v_i(t)$ in time interval T .

To be specific, the number of the node types in a temporal graph $\mathcal{G}(T)$ is determined by the types of online learning activities, i.e., N_v , which means that we can see a temporal graph \mathcal{G} as a static graph with N_v nodes at a specific duration, denoted as $\mathcal{G}(t) = (\mathcal{V}[0, t], \mathcal{E}[0, t])$. Therefore, we can apply spectral-based or spatial-based techniques to obtain the temporal embedding $\mathbf{v}_i(t)$ of $v_i(t)$ in temporal graph convolutional operators. The features of node v_i are denoted as a tuple $(v_{i,1}, \dots, v_{i,j}, \dots)$, where $v_{i,j}$ denotes the j th feature of v_i , e.g., the type of learning activity, or the duration of the learning activity, and so on. The features of temporal edge $e_{ij}(t)$ are denoted as a tuple $(e_{i,j,1}, \dots, e_{i,j,k}, \dots)$, where $e_{i,j,k}$ denotes the k th feature of $e_{ij}(t)$, e.g. the timestamp of transition. Furthermore, we can define the node or edge features with different time intervals for efficient computation with dynamic graphs. Together with temporal graphs and their node or edge features, an effective temporal graph network is proposed to obtain the representation of a sequence of online learning activities in the following subsection.

Low-high filtering temporal graph networks

From Fig. 1, we can see that there are two crucial temporary representations of academic performance to reach the final representation, one is generated from temporal graph networks (locally), which is detailed in this subsection, the other is from the global sampling module (globally), detailed in the following subsection.

Following the conventions, we also adapt an encoder-decoder architecture to realize the temporal graph networks for a local representation of online learning activities. There may exist an over-smooth problem [67] in temporal graph learning after several

propagation operations with different online learning transitions. Thus, we propose an adaptive low-high filtering temporal graph neural network for problem-solving.

Propagation function From the process of dynamic graph construction, we know that dynamic graphs are temporal event-driven in online learning activities. Therefore, the transition between online learning activities is simulated as propagation functions in TGN and can be expressed as:

$$\mathbf{p}_i(t) = \text{pgf}\left(\mathbf{v}_i^s(t^-), v_i(t), \mathbf{v}_j^d(t^-), v_j(t), \Delta t, e_{i,j}(t)\right) * \sigma\left(\mathbf{v}_i^s(t^-), \mathbf{v}_j^d(t^-)\right), \quad (9)$$

$$\mathbf{p}_j(t) = \text{pgf}\left(\mathbf{v}_j^s(t^-), v_j(t), \mathbf{v}_i^d(t^-), v_i(t), \Delta t, e_{j,i}(t)\right) * \sigma\left(\mathbf{v}_j^s(t^-), \mathbf{v}_i^d(t^-)\right), \quad (10)$$

where $\mathbf{v}_i^s(t^-)$ denotes the memory representation of node v_i before time t , $v_i(t)$ is the raw feature, as a source node in the transition between online learning activities, $\mathbf{v}_j^d(t^-)$ for the destination one, σ is a learnable gate function. If the transition of activities is self-loop, the propagation is expressed as:

$$\mathbf{p}_i(t) = \text{pgf}\left(\mathbf{v}_i^{sd}(t^-), v_i(t), \Delta t\right) * \sigma\left(\mathbf{v}_i^{sd}(t^-)\right), \quad (11)$$

where *pgf* is the similar learnable propagation function as Eqn. (9) and (10).

Low-high filtering aggregator We will perform information aggregation several times after information propagation as Eqn (9), (10) and (11). Inspired by the work [68], we propose an adaptive low-high filtering aggregator for temporal graph networks for online learning interaction activities, which can be formulated as follows:

$$\hat{\mathbf{p}}_i = \sum_{j \in \{\mathcal{N}(v_i(t)) \cup i\}} \alpha_{ij}^L \left(\mathcal{F}_i^L \mathbf{p}_j \mathcal{F}_r^L\right) + \alpha_{ij}^H \left(\mathcal{F}_i^H \mathbf{p}_j \mathcal{F}_r^H\right), \quad (12)$$

where \mathcal{N} denotes the neighboring operator, α_{ij}^L and α_{ij}^H are coefficient to feature representation node v_i with the relation $\alpha_{ij}^L + \alpha_{ij}^H = 1$, \mathcal{F}_i^L and \mathcal{F}_i^H are low-high filters similar in [68], \mathcal{F}_r^L and \mathcal{F}_r^H are operators of element-wise attention mechanisms between \mathbf{p}_i and \mathbf{p}_j .

Memory updater and local representation As previously mentioned, one part of the final representation of student academic performances is generated locally from a temporal graph network. Thus, we first need to obtain the node-wise features of the online activities, which can be formulated as follows:

$$\mathbf{s}_i(t) = \text{upd}\left(\mathbf{s}_i(t^-), \hat{\mathbf{p}}_i\right), \quad (13)$$

where *upd* can be implemented by a learnable neural network, e.g., GRU or LSTM, and $\mathbf{s}_i(t)$ is the temporal state of node v_i at time step t . The local representation of student academic performance can be learned with Eqn. (13). It can be defined as:

$$\hat{\mathbf{z}}^L(T) = \text{CPooling}\left(\{\mathbf{s}_1(T), \mathbf{s}_2(T), \dots, \mathbf{s}_{N_v}(T)\}\right), \quad (14)$$

where *CPooling* denotes a column-wise average or max pooling technique to obtain the local representation of student academic performance, i.e., $\hat{\mathbf{z}}^L(T)$.

Global sampling module

With the collection of interaction features $FI(S)$, we can apply a K-means clustering algorithm to construct the target global interaction feature dictionary.

Global sampling Specifically, some datasets' whole interaction features may be too large to perform a clustering algorithm. We may choose a subset of them for constructing the dictionary, and we will note this in the experimental settings. The process to obtain the global interaction feature dictionary $Gdict$ for $FI(S)$ can be formulated as follows:

$$\mathbf{z}^G = Gdict(FI(S)) = K-Means(FI(S)), \tag{15}$$

where $Gdict(FI(S))$ is a matrix with the size of $N \times d_k$, and d_k is the dimension of interaction features. The optimization object to get N cluster-shaped dictionary is formulated by

$$\arg \min_{FI(S)} K-Means(FI(S)) = \sum_{n=1}^N \sum_{s_i \in S} \sum_{f^{(j)} \in In(s_i)} \|f^{(j)} - \mu^{(n)}\|^\delta, \tag{16}$$

where $f^{(j)}$ denotes $in(i, j) \in In(s_i)$, $\mu^{(n)}$ denotes the n th candidate vector of the global interaction feature dictionary, $\| * \|^\delta$ represents a distance function. A cosine similarity or Euclid distance function is often employed in the algorithm. This setting ensures that global and local sampling estimates are based on the same distribution.

Linear transformation layer The feature vectors \mathbf{z}^G from *Global Sampling* may not be in a well-aligned space to the features from TGN. Thus, we introduce a simple linear transformation layer to obtain a feature representation from a global perspective. The process can be formulated as follows:

$$\begin{aligned} Q^G &= FFN(\mathbf{z}^G, \{D_k, head\}), \\ K^G &= FFN(\mathbf{L}_i, \{D_k, head\}), \\ \hat{\mathbf{z}}^G &= \frac{Q^G \tilde{\otimes} K^G}{\sqrt{D_k}} \tilde{\otimes} \mathbf{z}^G, \end{aligned} \tag{17}$$

where $D_k, head$ are parameters for the attention mechanism, \mathbf{L}_i is a feature vector for student l_i , $\tilde{\otimes}$ is multiplication with broadcasting property.

Academic performance representation and prediction

As Fig. 1 shown, the final representation of academic performance is generated from a local branch of TGN and a global branch of the global sampling module. We apply a simplified multi-head attention mechanism to fuse these local–global features to obtain the academic performance representation. It can be defined as:

$$\begin{aligned} Q_i &= FFN(\hat{\mathbf{z}}_i^G), \\ K &= FFN(\mathbf{z}^L), \\ V_i &= \frac{Q_i \tilde{\otimes} K}{\sqrt{d_k}} \tilde{\otimes} \hat{\mathbf{z}}^L, \\ V &= CPooling(\{V_1, V_2, \dots, V_N\}), \end{aligned} \tag{18}$$

where z is the output of *CPooling* as in Eqn. (14). With the final representation z , an MLP-based classifier is applied to z to obtain the academic performance prediction of online candidate learners, i.e., $y = MLP(V)$, y is the predicted result on a given representation V . Following the convention of classification tasks with neural networks, a cross-entropy loss is utilized to train our APP-TGN model.

Case study

Research questions

A case study on the widely recognized OULA dataset [21] validates the superior performance of APP-TGN in forecasting student academic outcomes. The study aims to answer the following research questions:

- Question One (Q1): How does the proposed APP-TGN perform when predicting student academic performance in terms of classification accuracy, F1-score, and recall?
- Question Two (Q2): What is the improvement in early prediction of at-risk students when using APP-TGN against other state-of-the-art methods?
- Question Three (Q3): What contribution does each proposed component of APP-TCN make to the final prediction performance in terms of classification accuracy?

Dataset and baselines

Dataset A subset of the Open University Learning Analytics dataset (OULA) [21], specifically code-Module FFF (2013B, 2013J), is chosen for evaluation. The refined data encompasses academic records of 3897 students, encapsulating student details, online learning interaction logs, and academic performance. Figure 2 visually represents the spread of students’ grades. For the sake of simplicity in our study, students were categorized into three groups: *Pass* (encompassing *Pass* and *Distinction*), *Withdrawn*, and *Fail*, as depicted in (b). Besides the basic information (e.g. gender, region, highest_education) of students, Table 1 summarizes online learning activities to construct dynamic graphs.

Baselines The case study employs a variety of machine learning models as baselines to evaluate our proposed APP-TGN. They are - optimized multiple layer perception (OMLP) [69], ProbSAP [70], CNN-LSTM [71], graph neural networks MTGNN [18] and a modified multi-view graph transformer from [31] (noted as AP-GT), hybrid recurrent networks (HRNs) [72] and a variant of our model, denoted APP-TGN1, where APP-TGN

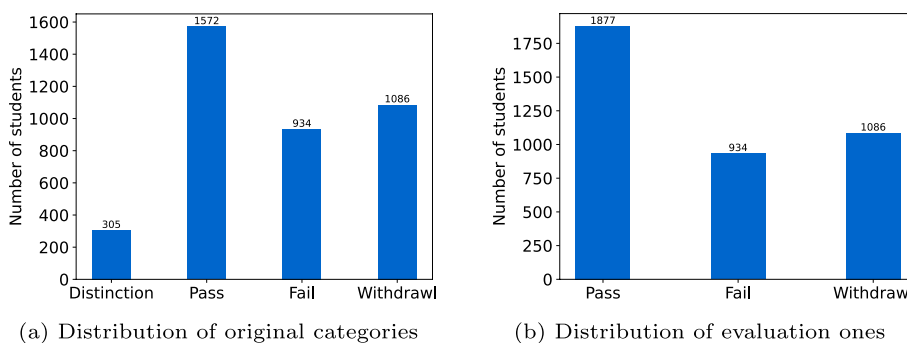


Fig. 2 Statistics of code-Module FFF in OULA

Table 1 Online learning activities to construct dynamic graphs

Activities	Descriptions
Homepage	Interaction with course homepage
Subpage	Interaction with the other sites embedded in the course
Url	Interaction with the links to audio/video contents
Forumng	Interaction with the discussion forum
Oucontent	Interaction with the contents of the assignment
Resource	Interaction with the search tool
Quiz	Interaction with the course quiz

substitutes the TGN module for TGN as per [73]. This variant serves the role of baseline models, and we contrast it against our newly introduced APP-TGN. Both the reference models and our APP-TGN are built using PyTorch and Python.

Experimental settings

Training and testing setup We partition the dataset, allocating 80% of the samples for training purposes and reserving the remaining 20% for testing. The training set undergoes further partitioning. Here, 90% of the samples form the training set, while the remaining portion aids in the process of identifying optimal hyper-parameters and model configurations. As for the sequential models like GRU, APP-TGN1, and APP-TGN, we will tune the hyper-parameter of the window size to achieve their best performance. To be specific, as we detail in Sect. "Data Cleaning and Pre-processing" and "Dynamic graph construction", the dynamic graph construction involves feature selection for the process, we choose the learning materials (denoted as id_site in the dataset) as the nodes. Not all the learning materials or learning activities are employed in the graph construction, the ones used in the process are summarized in Table 1. We cannot build a directed edge between nodes because each learning activity has no fine-grained timestamps. We suppose the materials or nodes used within a day have a non-directional edge between them. The raw features for a node are a tuple ($site_id, sum_click, date$).

Hyperparameter tuning and optimization In the APP-TGN framework, a thorough process of hyperparameter tuning and optimization was carried out. Different propagation and gate functions were experimented with for the low-high filtering temporal graph network module. The challenge lay in striking a balance between complexity and performance. The Identity function for propagation and a three-layer MLP for the gate function yielded the best results. Various configurations were tested for the low-high filter aggregator for the low and high filters. The primary challenge was to ensure the filters effectively captured the relationships among the neighboring vectors. The best performance was achieved when the low filter was set as the addition of neighboring vectors and the high filter as the subtraction of neighboring vectors. The linear transformation layer and the FNN function in the global sampling module were optimized. A three-layer MLP for the FNN function, with three heads and a D_k of 100, yielded the best results. For the academic performance representation and prediction module, a three-layer MLP was also used for the FNN function. The challenge was to ensure that the output vector had the

right dimensionality. An output vector with a dimensionality of 100 proved to be the best. ReLU activation functions were used throughout the entire process, and the APP-TGN was initialized with random parameters following a normal distribution with a standard deviation of 0.1. The main challenge was to prevent overfitting while achieving high performance. This setup provided a good balance between model complexity and performance. Overall, hyperparameter tuning and optimization was a complex task requiring careful experimentation and considering trade-offs between different factors. However, the effort was worthwhile as it significantly improved the performance of the APP-TGN framework.

Evaluation metrics The task of predicting student performance is approached as a binary classification problem. The metrics listed below serve as the basis for comparing performance:

- Classification Accuracy(ACC):

$$ACC = \frac{TP + TN}{TP + FP + FN + TN},$$

where TP, FP, FN, and TN denote the count of True Positive, False Positive, False Negative, and True Negative instances in the confusion matrix.

- Recall(REL):

$$REL = \frac{TP}{TP + FN},$$

where REL is the proportion that the model is accurately classifying the true positives;

- F1-score(F1):

$$F1 = \frac{2 * REL * PRE}{REL + PRE},$$

where F1 is the harmonic mean of REL (REL = TP / (TP + FN)) and PRE (Precision, defined as the proportion of true positives among predicted positives).

Results and discussions

This subsection details the empirical study results of APP-TGN and other baselines from two perspectives. The first experimental study is to answer the research question one, i.e., *How does the proposed APP-TGN perform when predicting student academic performance in terms of classification accuracy, F1-score, and recall?* The task in the experiments for the evaluated models is to exploit students' learning logs of the whole semester to predict their academic performances in the course, e.g., *Pass/Fail*, or *Pass/Withdrawn*. The second experimental study aims to answer the second research question, i.e., *What is the improvement in early prediction of at-risk students when using APP-TGN against other state-of-the-art methods?* The merits of APP-TGN in comparison to other baselines for the early identification of students at risk of not excelling in the initial weeks of the term are examined.

Academic performance prediction with whole online learning logs (Q1)

The experiment involves two distinct tasks: identifying students who might fail and those who might withdraw. To identify students who might fail, students are classified as either *Pass* or *Fail*. Similarly, to identify students who might withdraw, students are classified as either *Pass* or *Withdrawn*.

Table 2 reports the experimental results of the tasks, and we use bold font to denote the best performance. We find several observations in the following. Firstly, **superior performance of APP-TGN**: Our APP-TGN model outperforms the baseline models in both sub-tasks, achieving an accuracy of 83.22% in the *Pass/Fail* task and 77.06% in the *Pass/Withdrawn* task. Secondly, **advantage of graph-Based models**: Graph-based models (MTGNN, AP-GT, APP-TGN1, and APP-TGN) consistently surpass non-graph-based models (ProbSAP, CNN-LSTM, OMLP, HRNs) in all metrics, demonstrating their effectiveness in predicting academic performance. Thirdly, **comparison of AP-GT and MTGNN**: AP-GT and MTGNN, utilizing multiple graphs, show similar prediction performance. However, AP-GT performs slightly better due to its deep feature transformation after GNN representation, a technique also used in our model. Fourthly, **benefit of temporal graph structure**: Models incorporating the TGN module (APP-TGN and APP-TGN1) outperform static graph neural networks (AP-GT, MTGNN), indicating that a temporal graph structure can more effectively encode learning behavior data for academic performance prediction. In particular, **effectiveness of low-high filtering mechanism**: Our APP-TGN model, which includes a low-high filtering mechanism, surpasses the APP-TGN with a standard TGN module in three metrics, demonstrating the practical effectiveness of this mechanism. Our APP-TGN introduces a suitable graph structure with temporal property to encode the learning behavior data, which can capture academic states in their complex learning processes, so its predictive performance improves. Furthermore, **consistent performance across various training sizes**: As depicted in Fig. 3, our APP-TGN model maintains superior performance across various sizes of training sets, demonstrating its robust ability to discern students' academic states from learning behavior data. In summary, our APP-TGN model introduces a suitable graph structure with temporal property to encode the learning behavior data, which can capture academic states in their complex learning processes, thereby improving its predictive performance. Further experimental studies will scrutinize the effectiveness of the components of our APP-TGN model.

Table 2 Comparative analysis of baseline models and APP-TGN in identifying students who are at risk in terms of ACC, F1, and REL

Tasks	Metrics	ProbSAP	CNN-LSTM	OMLP	HRNs	MTGNN	AP-GT	APP-TGN1	APP-TGN
<i>Pass/Fail</i>	ACC	77.80	80.10	76.60	79.29	80.21	80.38	82.66	83.22*
	F1	65.41	67.92	64.98	65.74	69.89	70.18	71.23	73.11*
	REL	63.11	73.15	60.47	64.38	73.53	79.55	87.85	90.21*
<i>Pass/Withdrawn</i>	ACC	70.50	72.26	69.36	71.71	72.63	73.21	75.13	77.06*
	F1	72.02	75.94	71.81	72.65	76.03	76.48	77.21	79.70*
	REL	67.53	70.12	50.62	68.34	71.40	74.03	78.25	82.43*

An asterisk (*) denotes improvements that are statistically significant when compared to the best performing baseline. These improvements are validated by a two-sided t-test with a p-value less than 10^{-3} , confirming their statistical significance

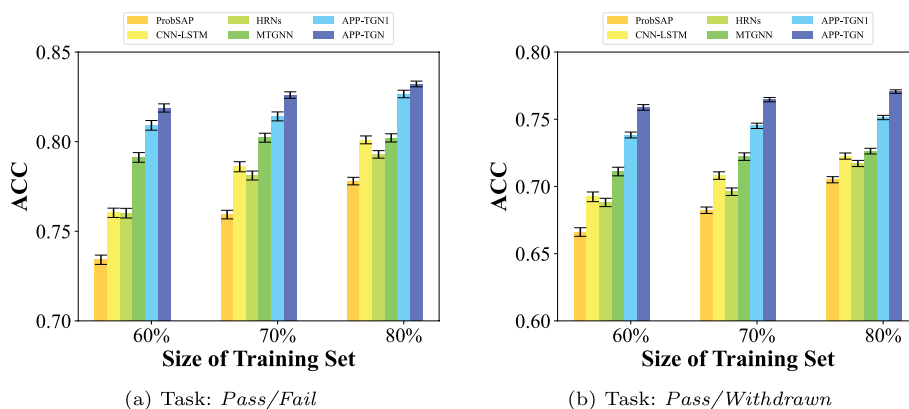


Fig. 3 ACC(%) of APP-TGN against other baselines for predicting at-risk students

Table 3 Comparisons between the baseline models and APP-TGN in **early** predicting at-risk students in terms of ACC(%)

Tasks	Duration	ProbSAP	CNN-LSTM	OMLP	HRNs	MTGNN	AP-GT	APP-TGN1	APP-TGN
<i>Pass/Fail</i>	Week5	69.89	71.42	69.13	72.31	72.88	72.81	75.10	77.12*
	Week10	70.81	73.31	70.26	73.22	73.25	73.52	75.89	78.76*
	Week15	72.09	74.31	71.66	74.23	74.64	75.06	77.58	80.05*
	Week20	72.89	76.10	72.11	75.83	76.33	76.91	79.11	81.65*
<i>Pass/Withdrawn</i>	Week5	62.88	63.48	61.92	65.72	64.98	65.06	68.88	72.11*
	Week10	63.37	64.82	62.11	66.30	65.75	66.13	69.86	73.35*
	Week15	65.53	68.09	63.28	67.86	67.04	67.58	70.89	74.43*
	Week20	66.18	69.17	64.26	68.84	69.56	70.01	72.08	75.13*

An asterisk (*) denotes improvements that are statistically significant when compared to the best performing baseline. These improvements are validated by a two-sided t-test with a p-value less than 10^{-3} , confirming their statistical significance

Early prediction for at-risk students with partial online learning logs(Q2)

The task of this experiment is to answer the second research question, i.e., *What is the improvement in early prediction of at-risk students when using APP-TGN against other state-of-the-art methods?* Early prediction of students’ performance is an important application in online learning management systems, as we can identify students at risk of failing or dropping out early. Some active invention policies or actions can be applied promptly, giving them enough time to improve their abilities and understanding. We have split the task into two sub-tasks: predicting early on whether students are at risk of failure, categorized as *Pass* or *Fail*, and identifying students who may drop out prematurely, categorized as *Pass* or *Withdrawn*. Following a similar experimental setting except for the duration (weeks 5, 10, 15, and 20) of learning logs for training and testing.

The comparison between the baseline models and APP-TGN in predicting at-risk students early is presented in Table 3. It is evident that APP-TGN consistently surpasses the other baseline models in accuracy across all learning periods. Among the baseline models, graph-based models, including AP-GT and MTGNN, exhibit competitive performance compared to non-graph-based models. This suggests that the graph-based approach, which captures complex interactions among learning activities, is beneficial for this prediction task. Interestingly, APP-TGN and its variant, APP-TGN1, outperform

AP-GT and MTGNN and perform better in different periods. This indicates that the techniques proposed in APP-TGN, such as temporal graph networks, are effective for early prediction tasks. Moreover, it is worth noting that the performance of all models improves over time, as more academic information becomes available. However, APP-TGN shows the most significant improvement, further highlighting its effectiveness in utilizing temporal information for prediction. Specifically, Fig. 4a illustrates how APP-TGN achieves an accuracy rate of 81.65% in predicting students who might fail, and Fig. 4b shows an accuracy rate of 71.13% in predicting students who might withdraw. These figures highlight the potential for early identification of students who are at risk. Moreover, Fig. 4 illustrates that APP-TGN surpasses other compared methods in early prediction, showcasing its high capacity for early intervention. This is important for addressing student issues promptly and encouraging their learning journey.

Effectiveness of APP-TGN (Q3)

This part aims to answer the second research question, i.e., What contribution does each proposed component of APP-TCN make to the final prediction performance regarding classification accuracy? As our APP-TGN consists of several significant components and hyper-parameters, we investigate their contribution to the performance of model predictions with ablation study and parameter sensitivities.

Effectiveness of different components of APP-TGN To evaluate the impact of different components of APP-TGN on the prediction performance, we introduce some notations to denote different ablation settings of APP-TGN: **APP-GS** denotes the APP-TGN without global sampling module, and takes L_i as z^G directly; **APP-LTL** denotes the global sampling module without a linear transformation layer; **APP-GRU** denotes the APP-TGN with a GRU network [13] as TGN module; **APP-TGN1** denotes the APP-TGN with a normal temporal graph network [73] as the TGN module. Table 4 shows the accuracy of different components of APP-TGN, and the numbers in the parentheses are deviations from the best prediction performance. We can make the following observations from the table. First, we can see that all main components of APP-TGN are important for the prediction performance for both Pass/Fail and Pass/Withdrawn classification, indicating that the proposed techniques can effectively capture the temporal and relational features of online learning behavior data. It shows that the APP-TGN model can provide a comprehensive and dynamic representation of students' academic

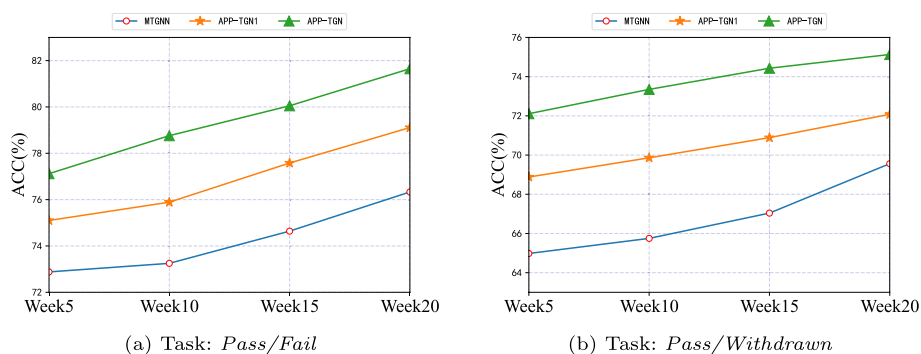


Fig. 4 APP-TGN against MTGNN, APP-TGN1 for early predicting at-risk students in terms of ACC(%)

Table 4 Effectiveness different components of APP-TGN in terms of ACC(%)

Methods	GS	LTL	TGN	TGN1	GRU	Pass/Fail	Pass/Withdrawn
APP-GS	×	×	✓	×	×	82.15(-1.07)	75.74(-1.32)
APP-LTL	✓	×	✓	×	×	82.68(-0.54)	76.08(-0.98)
APP-GRU	✓	✓	×	×	✓	81.11(-2.11)	74.21(-2.85)
APP-TGN1	✓	✓	×	✓	×	82.76(-0.46)	75.63(-1.43)
APP-TGN	✓	✓	✓	×	×	83.22(-0.00)	77.06(-0.00)

Table 5 ACC(%) of APP-TGN with different settings of time units to construct dynamic graphs

Tasks	2-days	4-days	6-days	8-days	10-days	12-days
Pass/Fail	82.16	82.75	83.22	82.18	81.70	81.22
Pass/Withdrawn	75.53	76.77	77.06	76.25	75.50	74.83

performance, which can help educators and students monitor and improve their learning outcomes. Second, we can see that the GS module can help reduce data bias due to the training dataset. Without the GS module, there is a 1.07% and 1.32% decrease in *Pass/Fail* and *Pass/Withdrawn*, respectively. This suggests that the GS module can enhance the APP-TGN model's generalization ability, making it more robust to different learning scenarios and student groups. Third, the APP-GRU model does not include a TGN module and, therefore, ignores interaction information between learning behavior data. This can result in a significant decrease in prediction performance. APP-GRU has the lowest prediction performance for both *Pass/Fail* and *Pass/Withdrawn* sub-tasks, at 81.11% and 74.21%, respectively. That is, the interaction information between learning behavior data is crucial for understanding students' academic performance, and the TGN module can effectively model such information. Fourth, APP-TGN1 and APP-TGN both have a TGN module in their models, but we can see that APP-TGN shows a better prediction performance over APP-TGN1 for two sub-tasks. The difference is that the TGN module in our APP-TGN adapts a low-high filtering information aggregation design. In contrast, the TGN module in APP-TGN1 adapts a conventional implementation [73], implying that the low-high filtering design is a better solution to capture more academic information during their learning processes. It demonstrates that the low-high filtering design can help the APP-TGN model distinguish between different learning behavior data levels, focusing on the most relevant and informative ones for academic performance prediction.

Parameter sensitivity in APP-TGN A parameter sensitivity analysis is performed on the main hyper-parameters in APP-TGN. Dynamic graph construction is crucial in APP-TGN, with the window size for updating a temporal graph being a critical hyperparameter that impacts prediction performance. Experimental results from various window size settings are presented in Table 5. The prediction performances of these two sub-tasks are pretty sensitive to these hyper-parameter settings. APP-TGN achieves the best performance at a window size of 6 days. For the two sub-tasks, the performance of APP-TGN decreases when the window size exceeds 6 days. This suggests that using a large window size to update a dynamic graph may result in information loss and poor graph

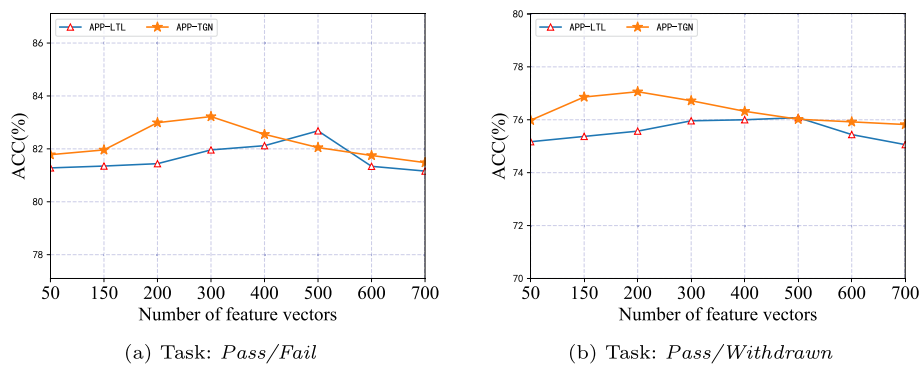


Fig. 5 ACC(%) of APP-TGN and APP-LTL with different settings of the number of feature vector N in a global sampling module

Table 6 ACC (%) of APP-TGN with different omissions of data fields in the dataset

Tasks	Homepage	Subpage	Url	Forumng	Oucontent	Resource	Quiz
Pass/Fail	82.90	82.56	82.39	81.54	82.33	82.70	81.67
Pass/Withdrawn	76.82	76.64	76.26	75.84	76.12	76.36	75.64

construction. It implies that the online learning logs of students are more informative and relevant when they are closer in time, and that older logs may not reflect students' current state and behavior. As the window size increases beyond 6 days, the performance worsens. Furthermore, from Table 4, we can see that the global sampling module plays a vital role in APP-TGN, which is an effective technique for reducing data bias. This means that the model can learn from a more representative and diverse set of students rather than focusing on a few dominant or frequent ones. The experimental results for APP-TGN and APP-LTL, concerning different hyper-parameter settings for the feature vectors N , are visualized in Fig. 5. As shown in Fig. 5a, APP-TGN delivers optimal performance with N as 300, while APP-LTL requires a larger amount of feature vectors, precisely 500, for optimal performance. Figure 5b also shows a similar result, demonstrating the effectiveness of the linear transformation layer in the global sampling module. The layer can help reduce the dimensionality and complexity of the feature vectors, making them more suitable for temporal graph networks.

Feature Importance and Contribution In the experiment conducted by us, the goal was to comprehend how different types of interactions influence student outcomes. Seven interaction features were utilized (as listed in Table 1), and an ablation study was carried out. This study involved the omission of one feature at a time from our APP-TGN model. The changes in prediction accuracy (%) for each performance category, resulting from this process, were documented and are displayed in Table 6. The analysis brought to light that the *Quiz* and *Forumng* features have a significant bearing on the performance prediction of the model. The accuracy experienced a considerable drop when these features were removed, suggesting their critical role in capturing students' learning behaviors and progress. It implies that future strategies for data collection could prioritize obtaining more detailed data concerning quizzes and forum interactions. Conversely, features such as *Homepage*, *Subpage*, *Resource*,

among others, had a less noticeable impact on the prediction accuracy. This could be attributed to the redundancy or lower relevance of these features for the task at hand. Hence, future enhancements to the model could consider exploring techniques for feature selection or transformation to minimize redundancy and boost the predictive power of the input features. Interestingly, it was also observed that the influence of each feature differs across various performance categories, indicating that different features might be capturing distinct aspects of student performance. For example, a feature that is highly predictive for one category (e.g., *Pass*) might not be as informative for another category (e.g., *Fail*). This insight could steer the development of models specific to each category or the application of multi-task learning techniques to harness the differential predictive power of the features. In conclusion, the comprehensive analysis of the importance and contribution of features offers valuable insights that can enhance the model's performance and guide future strategies for data collection.

Model Complexity and Computation Cost of APP-TGN The APP-TGN model is designed with computational efficiency in mind, making it suitable for handling large-scale MOOC data. The computational complexity of APP-TGN can be estimated by considering its components. A 1-layer GCN has a complexity of $\mathcal{O}(|E|d_i d_o)$ where $|E|$ is the number of edges, d_i is the input feature dimension, and d_o is the output feature dimension. A GAT-like layer [74] has a complexity of $\mathcal{O}(N_v d_i d_o + |E|d_o)$, where N_v is the number of activity types. The linear transformation attention in APP-TGN has a linear complexity of $\mathcal{O}(N_v)$, similar to Linformer [75]. The k-Means feature clustering in the global sampling module is pre-processed and remains constant during training and testing. Therefore, the overall complexity of APP-TGN can be estimated as $\mathcal{O}(S|E|d_i d_o + Sd_m d_o)$, where S the step size for prediction, d_m denotes the number of neurons in MLP for realizing learnable functions. Since the graph in each step is usually sparse, the computational cost of APP-TGN is similar when S is small. We report the FLOPs of several baselines and our APP-TGN (with a window size of 6 days). The FLOPs are as follows: OMLP - 0.151M, HRNs - 0.263M, CNN-LSTM - 0.924M, MTGNN - 1.705M, and APP-TGN - 0.6621M. Our computational cost is less than that of MTGNN. Compared to computer vision models like ResNet (1.8G FLOPs), the computational cost of these models is relatively small for this task and is not yet a significant concern. This further underscores the efficiency and scalability of APP-TGN for large-scale MOOC data.

Visualization of academic performance representations We visualize the academic representation of the category of *Pass/Withdrawn* in Fig. 6. Figure 6a shows the representations from the original feature spaces, where the features of *Withdrawn* and *Pass* overlap together in a feature space, making it difficult to classify a specific feature. Figure 6b displays the representations learned from our APP-TGN of the category of *Withdrawn* and *Pass*. It can be seen that most features learned by APP-TGN are separable in the feature space. Compared to those not learned by APP-TGN, Feature representations learned by it have a more structured form and clear category boundaries. Thus, our APP-TGN can effectively cluster students' academic performances within the same category, which can help educators identify students' learning patterns, strengths, and weaknesses and provide personalized feedback and intervention.

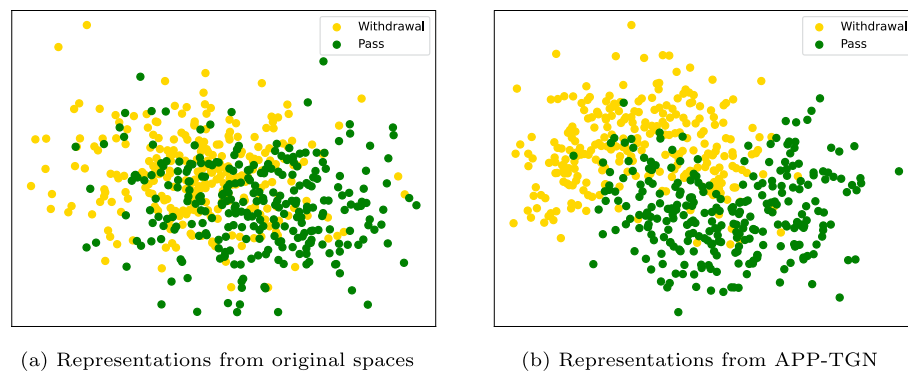


Fig. 6 Visualization of academic performance representations

Model Interpretability in Educational Context In this section, we discuss how the predictions of APP-TGN can be interpreted in an educational context based on the analysis of the model components and the experimental results. First, the dynamic graph construction module captures students' temporal information and interaction behaviors during their online learning activities, which reflect their learning processes and states. The temporal graphs can be visualized to show the patterns and transitions of different learning activities, such as watching videos, reading texts, or taking quizzes. Second, the low-high filtering temporal graph network module learns the potential academic performance variations encoded in the dynamic graphs, representing student knowledge and skills changes over time. The low-high filters can identify the nodes' and edges' important and relevant features in the temporal graphs, such as the frequency, duration, order, or correlation of the learning activities. Third, the global sampling module mitigates the problem of false correlations in deep learning-based models by incorporating students' demographic and contextual features, such as gender, region, disability, or highest education. The global sampling module can also provide a way to compare and contrast the performance of different groups of students based on these features. Finally, the academic performance representation and prediction module combines students' local and global representations and uses a multi-head attention mechanism to generate the final predictions of academic outcomes. The attention weights can be interpreted as the importance or relevance of different features or components for the prediction task. For example, the attention weights can indicate which types of learning activities or which demographic or contextual factors are more influential in predicting a specific student's performance or group of students. By providing these interpretations, APP-TGN can help educators and learners understand the factors and processes that affect students' academic performance in online courses and provide feedback and guidance for improving their learning outcomes.

Implications

This paper introduces APP-TGN, a new method that uses online learning logs to predict academic performance. APP-TGN does not rely on any existing framework but instead constructs a dynamic graph from the raw data and applies temporal graph networks to learn the academic performance representation and prediction. Our framework leverages temporal graph networks to capture the dynamic and complex relationships

between learning behaviors and academic outcomes. We also introduced a global sampling module to improve the representation learning for temporal graphs and a low-high filtering technique that eliminates the noise in online learning data. Our APP-TGN model achieved high accuracy rates in two prediction tasks, outperforming several baseline models by a significant margin. Specifically, in the experimental study of the first research question, our APP-TGN model achieved accuracy rates of 83.22% and 77.06% for two different tasks. These results represent statistically significant improvements over other models, with increases ranging from 1.23% to 8.29%. In the experimental study of the second research question, our APP-TGN model showed better statistically significant improvements over other models in early predicting at-risk students, with increases ranging from 2.99% to 12.97%. Our APP-TGN model is particularly effective in mining the dynamic relationship between learning behavior data and accurately predicting at-risk students. The third research question also demonstrates the effectiveness and superiority of our proposed techniques in APP-TGN. Overall, our model has great potential for use in automated feedback and personalized learning in real-world educational applications.

Limitations The APP-TGN prediction model has some limitations regarding data, algorithm, ethics, and generalizability. Firstly, there are few course interactions that form the model's basis and could benefit from more data. Secondly, the APP-TGN algorithm cannot learn incrementally, or interactively like other supervised AI methods. However, an APP-TGN with a more extensive database could be used for quasi-real-time analysis. Thirdly, ethical considerations such as the potential influence of AI-enabled models on student learning outcomes should be considered. Future work could deliver real-time predictions, timely alerts, and suggestions to ensure positive outcomes from AI prediction methods. Lastly, the prediction method must enhance its generalizability through empirical research in various educational contexts and by considering external factors like offline classroom activities or social interactions.

Conclusions

Student academic performance prediction is fundamental in implementing intelligent services for massive open online courses. The paper explores exploiting temporal information and interaction behaviors during learning activities to promote the performance of model predictions. We represent the learning processes of e-learning students as dynamic temporal graphs that capture the temporal information and interaction behaviors during their studying. We also introduce APP-TGN, a new method for academic performance prediction that utilizes temporal graph neural networks. Specifically, in APP-TGN, a dynamic graph is constructed from the online learning activity logs. Generated graphs are forwarded to a revised temporal graph network with low-high filters to learn potential academic performance variations encoded in dynamic graphs. Furthermore, a global sampling module is developed to mitigate the problem of false correlations in deep learning-based models. Finally, the learned representations from the global sampling and local processing (with TGN) are forwarded to a multi-head attention module to get the predicted academic performances. We perform a case study with a popular dataset from a real-world educational application that is publicly available. Empirical study results indicate that APP-TGN, which

we introduce, surpasses other methods by a large margin. The ablation study also reveals the effectiveness and superiority of our APP-TGN techniques.

Future work and extensions We intend to explore the following directions: (i) Heterogeneous Data Sources: The primary focus of our existing model is structured data derived from learning management systems. However, the nature of educational data is often heterogeneous, incorporating text from student essays, audio from spoken responses, and video from recorded presentations. Our goal is to broaden the scope of our model to accommodate these varied data types. For example, we could employ natural language processing techniques for text data analysis, while audio and video data might be processed using deep learning models tailored for these specific data types. (ii) Incorporation of Additional Educational Data: Beyond the data currently in use, there are other forms of educational data that could offer valuable insights. These include demographic information, data on student learning styles, and affective states. The integration of these supplementary data sources could enhance the precision of our predictions and provide a more comprehensive understanding of student performance. (iii) Forecasting of Additional Educational Outcomes: Although our present focus is on predicting academic performance, the model has the potential to be modified to forecast other vital educational outcomes. These might encompass student retention rates, degrees of student engagement, or even student satisfaction. Each of these outcomes holds significant importance in the educational context, and their accurate prediction could have substantial implications for educational institutions. (iv) Pretraining-fine-tuning Schema: We are also keen on investigating a pretraining-fine-tuning schema in APP-TGN for a range of educational analytical tasks. This would involve retraining the model on a large dataset to discern general patterns, followed by fine-tuning it on a specific task with a smaller dataset. This method has proven effective in various domains and could enhance the performance of our model.

Acknowledgements

The research project is supported by National Natural Science Foundation of China (No. 62207028), and partially by Zhejiang Provincial Natural Science Foundation (No. LY23F020009), Key R & D Program of Zhejiang Province (No. 2022C03106), and National Natural Science Foundation of China (No. 62007031, 62177016), and Zhejiang Province Education Science Planning Annual General Planning Project (Universities) (No. 2023SCG367), and Open Research Fund of College of Teacher Education, Zhejiang Normal University (No. jykf22006).

Author contributions

The first author: Writing—original draft, Conceptualization, Software, Investigation, Writing—review and editing. The second author: Data curation, Visualization, Writing—review and editing.

Availability of data and materials

The authors have no permission to share the dataset.

Declarations

Ethics approval and consent to participate

The study has no ethical issues, and no potentially identifiable personal information is presented in this study.

Competing interests

The authors declare that they have no Competing interests.

Received: 25 December 2023 Accepted: 7 April 2024

Published online: 13 April 2024

References

1. Huang C, Huang Q, Wang D. Stochastic configuration networks based adaptive storage replica management for power big data processing. *IEEE Trans Ind Inf.* 2019;16(1):373–83.

2. Alsaroah AH, Al-Turjman F. Combining Cloud Computing with Artificial intelligence and Its Impact on Telecom Sector. *NEU J Artif Intell Internet Things*. 2023;2(3).
3. Baig MI, Shuib L, Yadegaridehkordi E. Big data in education: a state of the art, limitations, and future research directions. *Int J Educ Technol High Educ*. 2020;17(1):1–23.
4. Wang J. Comprehensive test and evaluation path of college teachers' professional development based on a cloud education big data platform. *Int J Emerg Technol Learn (Online)*. 2023;18(5):79.
5. Morris W, Crossley S, Holmes L, Trumbore A. Using transformer language models to validate peer-assigned essay scores in massive open online courses (MOOCs). In: *LAK23: 13th international learning analytics and knowledge conference*; 2023; p. 315–323.
6. Zheng Y, Yin B. Big data analytics in MOOCs. In: *2015 IEEE international conference on computer and information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive intelligence and computing*. IEEE; 2015; p. 681–686.
7. Crompton H, Burke D. Artificial intelligence in higher education: the state of the field. *Int J Educ Technol High Educ*. 2023;20(1):1–22.
8. Wang X, Mei X, Huang Q, Han Z, Huang C. Fine-grained learning performance prediction via adaptive sparse self-attention networks. *Inf Sci*. 2021;545:223–40.
9. Liang J, Yang J, Wu Y, Li C, Zheng L. Big data application in education: dropout prediction in edx MOOCs. In: *IEEE second international conference on multimedia big data (BigMM)*. IEEE. 2016;2016:440–3.
10. Chen Y, Zhai L. A comparative study on student performance prediction using machine learning. *Educ Inf Technol*. 2023;28:1–19.
11. Marbouti F, Diefes-Dux HA, Strobel J. Building course-specific regression-based models to identify at-risk students. In: *2015 ASEE Annual Conference and Exposition*; 2015; p. 26–304.
12. Xu X, Wang J, Peng H, Wu R. Prediction of academic performance associated with internet usage behaviors using machine learning algorithms. *Comput Human Behav*. 2019;98:166–73.
13. Wang X, Wu P, Liu G, Huang Q, Hu X, Xu H. Learning performance prediction via convolutional GRU and explainable neural networks in e-learning environments. *Computing*. 2019;101(6):587–604.
14. Kukkar A, Mohana R, Sharma A, Nayyar A. Prediction of student academic performance based on their emotional wellbeing and interaction on various e-learning platforms. *Educ Inf Technol*. 2023;28:1–30.
15. Yang Z, Yang J, Rice K, Hung JL, Du X. Using convolutional neural network to recognize learning images for early warning of at-risk students. *IEEE Trans Learn Technol*. 2020;13(3):617–30.
16. Waheed H, Hassan SU, Nawaz R, Aljohani NR, Chen G, Gasevic D. Early prediction of learners at risk in self-paced education: a neural network approach. *Expert Syst Appl*. 2023;213: 118868.
17. Sun X, Cheng H, Liu B, Li J, Chen H, Xu G, et al. Self-supervised hypergraph representation learning for sociological analysis. *IEEE Trans Knowl Data Eng*. 2023. <https://doi.org/10.1109/TKDE.2023.3235312>.
18. Li M, Wang X, Wang Y, Chen Y, Chen Y. Study-GNN: a novel pipeline for student performance prediction based on multi-topology graph neural networks. *Sustainability*. 2022;14(13):7965.
19. Wang C, Fang T, Gu Y. Learning performance and behavioral patterns of online collaborative learning: impact of cognitive load and affordances of different multimedia. *Comput Educ*. 2020;143: 103683.
20. Asadi M, Swamy V, Frej J, Vignoud J, Marras M, Käser T. Ripple: Concept-based interpretation for raw time series models in education. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2023; 13. p. 15903–15911.
21. Kuzilek J, Hlosta M, Zdrahal Z. Open university learning analytics dataset. *Sci Data*. 2017;4(1):1–8.
22. Ouyang F, Wu M, Zheng L, Zhang L, Jiao P. Integration of artificial intelligence performance prediction and learning analytics to improve student learning in online engineering course. *Int J Educ Technol High Educ*. 2023;20(1):1–23.
23. Ren Z, Rangwala H, Johri A. Predicting performance on MOOC assessments using multi-regression models. *arXiv preprint arXiv:1605.02269*. 2016.
24. Chui KT, Fung DCL, Lytras MD, Lam TM. Predicting at-risk university students in a virtual learning environment via a machine learning algorithm. *Comput Human Behav*. 2020;107: 105584.
25. Coussemont K, Phan M, De Caigny A, Benoit DF, Raes A. Predicting student dropout in subscription-based online learning environments: The beneficial impact of the logit leaf model. *Decis Support Syst*. 2020;135: 113325.
26. Riestra-González M, del Puerto Paule-Ruiz M, Ortin F. Massive LMS log data analysis for the early prediction of course-agnostic student performance. *Comput Educ*. 2021;163: 104108.
27. Turabieh H, Azwari SA, Rokaya M, Alosaimi W, Alharbi A, Alhakami W, et al. Enhanced Harris Hawks optimization as a feature selection for the prediction of student performance. *Computing*. 2021;103(7):1417–38.
28. Mubarak AA, Cao H, Zhang W. Prediction of students' early dropout based on their interaction logs in online learning environment. *Interact Learn Environ*. 2022;30(8):1414–33.
29. Jiao P, Ouyang F, Zhang Q, Alavi AH. Artificial intelligence-enabled prediction model of student academic performance in online engineering education. *Artif Intell Rev*. 2022;55(8):6321–44.
30. Batool S, Rashid J, Nisar MW, Kim J, Kwon HY, Hussain A. Educational data mining to predict students' academic performance: a survey study. *Educ Inf Technol*. 2023;28(1):905–71.
31. Peng T, Liang Y, Wu W, Ren J, Pengrui Z, Pu Y. CLGT: A graph transformer for student performance prediction in collaborative learning. In: *Proceedings of the AAAI conference on artificial intelligence*; 2023; p. 15947–15954.
32. Huang Q, Zeng Y. Improving academic performance predictions with dual graph neural networks. *Complex and Intelligent Systems*. 2024;p. 1–19.
33. Giannakas F, Troussas C, Voyiatzis I, Sgouropoulou C. A deep learning classification framework for early prediction of team-based academic performance. *Appl Soft Comput*. 2021;106: 107355.
34. Karimi H, Derr T, Huang J, Tang J. Online academic course performance prediction using relational graph convolutional neural network. In: *Proceedings of the 13th international conference on educational data mining (EDM 2020)*; 2020. p. 444–450.
35. Waheed H, Hassan SU, Aljohani NR, Hardman J, Alelyani S, Nawaz R. Predicting academic performance of students from VLE big data using deep learning models. *Comput Human Behav*. 2020;104: 106189.

36. Du X, Yang J, Hung JL. An integrated framework based on latent variational autoencoder for providing early warning of at-risk students. *IEEE Access*. 2020;8:10110–22.
37. Sun X, Cheng H, Dong H, Qiao B, Qin S, Lin Q. Counter-empirical attacking based on adversarial reinforcement learning for time-relevant scoring system. *IEEE Trans Knowl Data Eng*. 2023. <https://doi.org/10.1109/TKDE.2023.3341430>.
38. Li M, Zhang Y, Li X, Cai L, Yin B. Multi-view hypergraph neural networks for student academic performance prediction. *Eng Appl Artif Intell*. 2022;114:105174.
39. Sun X, Yin H, Liu B, Chen H, Cao J, Shao Y, et al. Heterogeneous hypergraph embedding for graph classification. In: *Proceedings of the 14th ACM international conference on web search and data mining*; 2021; p. 725–733.
40. Sun X, Zhang J, Wu X, Cheng H, Xiong Y, Li J. Graph Prompt Learning: A Comprehensive Survey and Beyond. *arXiv preprint arXiv:2311.16534*. 2023.
41. Sun X, Cheng H, Li J, Liu B, Guan J. All in One: Multi-Task Prompting for Graph Neural Networks. In: *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '23*. New York, NY, USA: Association for Computing Machinery; 2023; p. 2120–2131. <https://doi.org/10.1145/3580305.3599256>.
42. Wu Z, Pan S, Chen F, Long G, Zhang C, Philip SY. A comprehensive survey on graph neural networks. *IEEE Trans Neural Netw Learn Syst*. 2020;32(1):4–24.
43. Zhou J, Cui G, Hu S, Zhang Z, Yang C, Liu Z, et al. Graph neural networks: a review of methods and applications. *AI Open*. 2020;1:57–81.
44. Zhang Z, Cui P, Zhu W. Deep learning on graphs: a survey. *IEEE Trans Knowl Data Eng*. 2020;34:249.
45. Wang S, Zeng Z, Yang X, Zhang X. Self-supervised Graph Learning for Long-tailed Cognitive Diagnosis. In: *Proceedings of the AAAI conference on artificial intelligence*; 2023; p. 110–118.
46. Gao W, Liu Q, Huang Z, Yin Y, Bi H, Wang MC, et al. 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*; 2021; p. 501–510.
47. Zhang J, Mo Y, Chen C, He X. GKT-CD: Make Cognitive Diagnosis Model Enhanced by Graph-based Knowledge Tracing. In: *2021 International joint conference on neural networks (IJCNN)*. IEEE; 2021; p. 1–8.
48. Mao Y, Xu B, Yu J, Fang Y, Yuan J, Li J, et al. Learning behavior-aware cognitive diagnosis for online education systems. In: *International conference of pioneering computer scientists, engineers and educators*. Springer; 2021; p. 385–398.
49. Su Y, Cheng Z, Wu J, Dong Y, Huang Z, Wu L, et al. Graph-based cognitive diagnosis for intelligent tutoring systems. *Knowl-Based Syst*. 2022;253:109547.
50. Qi T, Ren M, Guo L, Li X, Li J, Zhang L. ICD: a new interpretable cognitive diagnosis model for intelligent tutor systems. *Expert Syst Appl*. 2023;215:119309.
51. Chen Y, Wang S, Jiang F, Tu Y, Huang Q. DCKT: a novel dual-centric learning model for knowledge tracing. *Sustainability*. 2022;14(23):16307.
52. Abdelrahman G, Wang Q, Nunes B. Knowledge tracing: a survey. *ACM Comput Surv*. 2023;55(11):1–37.
53. Nakagawa H, Iwasawa Y, Matsuo Y. Graph-based knowledge tracing: modeling student proficiency using graph neural network. In: *IEEE/WIC/ACM international conference on web intelligence (WI)*. IEEE. 2019;2019:156–63.
54. Yang Y, Shen J, Qu Y, Wang K, Zhu Y, et al. GIKT: a graph-based interaction model for knowledge tracing. In: *Joint European conference on machine learning and knowledge discovery in databases*. Springer; 2020; p. 299–315.
55. Tong H, Wang Z, Liu Q, Zhou Y, Han W. HGKT: Introducing Hierarchical Exercise Graph for Knowledge Tracing. *arXiv preprint arXiv:2006.16915*. 2020.
56. Song X, Li J, Tang Y, Zhao T, Chen Y, Guan Z. Jkt: a joint graph convolutional network based deep knowledge tracing. *Inf Sci*. 2021;580:510–23.
57. Wu Z, Huang L, Huang Q, Huang C, Tang Y. SGKT: session graph-based knowledge tracing for student performance prediction. *Expert Syst Appl*. 2022;206:117681.
58. Song X, Li J, Lei Q, Zhao W, Chen Y, Mian A. Bi-CLKT: Bi-graph contrastive learning based knowledge tracing. *Knowl-Based Syst*. 2022;241: 108274.
59. Wu T, Ling Q. Self-supervised heterogeneous hypergraph network for knowledge tracing. *Inf Sci*. 2023;624:200–16.
60. Wu T, Ling Q. Fusing hybrid attentive network with self-supervised dual-channel heterogeneous graph for knowledge tracing. *Expert Syst Appl*. 2023;225:120212.
61. Tan H, Wang C, Duan Q, Lu Y, Zhang H, Li R. Automatic short answer grading by encoding student responses via a graph convolutional network. *Interact Learn Environ*. 2023;31(3):1636–50.
62. Ying R, He R, Chen K, Eksombatchai P, Hamilton WL, Leskovec J. Graph convolutional neural networks for web-scale recommender systems. In: *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*; 2018; p. 974–983.
63. Wang H, Zhao M, Xie X, Li W, Guo M. Knowledge graph convolutional networks for recommender systems. In: *The world wide web conference*; 2019; p. 3307–3313.
64. Tan H, Wang C, Duan Q, Lu Y, Zhang H, Li R. Automatic short answer grading by encoding student responses via a graph convolutional network. *Interact Learn Environ*. 2020;31:1–15.
65. Agarwal R, Khurana V, Grover K, Mohania M, Goyal V. Multi-relational graph transformer for automatic short answer grading. In: *Proceedings of the 2022 Conference of the North American chapter of the association for computational linguistics: human language technologies*; 2022; p. 2001–2012.
66. Li H, Wei H, Wang Y, Song Y, Qu H. Peer-inspired student performance prediction in interactive online question pools with graph neural network. In: *Proceedings of the 29th ACM international conference on information & knowledge management*; 2020; p. 2589–2596.
67. Zhou Y, Zheng H, Huang X, Hao S, Li D, Zhao J. Graph neural networks: taxonomy, advances, and trends. *ACM Trans Intell Syst Technol (TIST)*. 2022;13(1):1–54.
68. Bo D, Wang X, Shi C, Shen H. Beyond low-frequency information in graph convolutional networks. In: *Proceedings of the AAAI conference on artificial intelligence*. 2021; vol. 35. p. 3950–3957.
69. Michira MK, Rimiru RM, Mwangi WR. Improved multilayer perceptron neural networks weights and biases based on the grasshopper optimization algorithm to predict student performance on ambient learning. In: *Proceedings of the 2023 7th international conference on machine learning and soft computing*; 2023; p. 61–68.

70. Wang X, Zhao Y, Li C, Ren P. ProbSAP: a comprehensive and high-performance system for student academic performance prediction. *Pattern Recognit.* 2023;137: 109309.
71. Talebi K, Torabi Z, Daneshpour N. Ensemble models based on CNN and LSTM for dropout prediction in MOOC. *Expert Syst Appl.* 2024;235:121187.
72. Kukkar A, Mohana R, Sharma A, Nayyar A. A novel methodology using RNN + LSTM + ML for predicting student's academic performance. *Education and Information Technologies.* 2024;p. 1–37.
73. Rossi E, Chamberlain B, Frasca F, Eynard D, Monti F, Bronstein M. Temporal graph networks for deep learning on dynamic graphs. arXiv preprint [arXiv:2006.10637](https://arxiv.org/abs/2006.10637). 2020.
74. Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph Attention Networks. In: International conference on learning representations; 2018; p. 1–12.
75. Wang S, Li BZ, Khabsa M, Fang H, Ma H. Linformer: Self-attention with linear complexity. arXiv preprint [arXiv:2006.04768](https://arxiv.org/abs/2006.04768). 2020.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.