Preprint – IJCAI 2025: This is the accepted version made available for conference attendees.
Do not cite. The final version will appear in the IJCAI 2025 proceedings.

Metapath and Hypergraph Structure-based Multi-Channel Graph Contrastive Learning for Student Performance Prediction

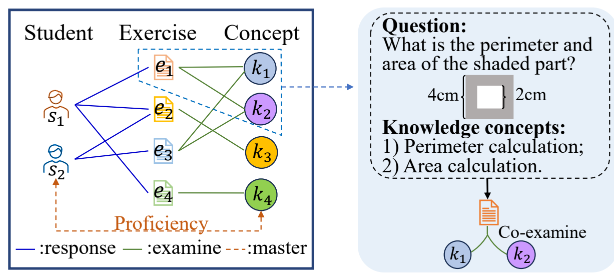
Lingyun Song[1,2], **Xiaofan Sun**[2], **Xinbiao Gan**[3*], **Yudai Pan**[2*], **Xiaolin Han**[2], **Jie Ma**[4], **Jun Liu**[4] and **Xuequn Shang**[2*]
[1]Shenzhen Research Institute of Northwestern Polytechnical University
[2]Key Laboratory of Big Data Storage and Management, Northwestern Polytechnical University, Ministry of Industry and Information Technology
[3]School of Computer Science, National University of Defense Technology
[4]School of Computer Science, Xi'an Jiaotong University
xinbiaogan@nudt.edu.cn, {panyuda, shang}@nwpu.edu.cn

Abstract

Considerable attention has been paid to predicting student performance on exercises. The performance of prior studies is determined by the quality of the trait features of students and exercises. Nevertheless, most of the prior study primarily examines simple pairwise interactions in learning trait features, like those between students and exercises or exercises and concepts, while disregarding the complex higher-order interactions that typically exist among these components, which in turn hinders the prediction results. In this paper, we using an innovative Multi-Channel Graph Contrastive Learning (MCGCL) framework that integrates various high-order interactions for predicting student performance. MCGCL characterizes graph structures reflecting various high-order relationships among students, exercises, and concepts through multiple channels, thereby enhancing the trait features of both students and exercises. Moreover, graph contrastive learning is employed to enhance the representation of trait features acquired from high-order graph structures in diverse views. Extensive experiments on real-world datasets show that MCGCL achieves state-of-the-art results on the task of predicting student performance. The code is available at https://github.com/sunlitsong/MCGCL.

1 Introduction

Student performance prediction has been the key to intelligent web education systems that provide personalized instructions for student learning. Existing cognitive diagnosis methods are mainly based on IRT [Lord, 2012] and Multidimensional IRT (MIRT) [Reckase, 2009], which model student performance as a result of interactions between student trait vectors and exercise trait vectors (e.g., exercise difficulty and discrimination). However, these methods usually rely on hand-crafted interaction functions, which often fail to capture the complex relationships between student traits and exercise traits. To address this, some works [Wang *et al.*, 2020] introduce complex student-exercise interactions into student performance prediction using Deep Neural Networks (DNN). The effectiveness of DNN-based interaction modeling is heavily dependent on the quality of the trait features of the students and the exercises. To improve trait feature learning, some researchers [Gao *et al.*, 2021] explore pairwise relationships within the three-level hierarchy of student-exercise-concept graphs. However, existing works focus only on pairwise relations, which are not enough to capture higher-order or composite interactions among students and exercises, and thus are not beneficial for improving trait learning.



(a) Heterogeneous graph of S-E-K (b) Hypergraph

Figure 1: Examples of complex composite interactions based on metapath and high-order interactions in student exercising process.

The higher-order relations pertain to interactions that include more than two entities at the same time, surpassing the conventional pairwise connections. For example, as shown in Fig. 1a, interactions between exercises and students or knowledge concepts are usually beyond just pairwise, e.g., multiple students do the same exercises or one exercise examines multiple concepts simultaneously. For example, in Fig. 1b, one exercise examines two knowledge concepts, representing a higher-order relationship that cannot be fully captured by pairwise edges. These relations encode composite semantics, reflecting inter-dependencies among multiple entities, whereas pairwise edges focus only on two-node interactions. These high-order relations are still underexplored,

---
*Corresponding author

leading to a significant loss of information and thus hindering the learning of trait vectors [Liu *et al.*, 2023]. Several hypergraph neural networks (HNNs) have been developed to model these high-order relations, where hypergraph convolutions are applied to graph structures of hyperedges connecting more than two nodes [Bai *et al.*, 2021]. However, the data sparsity in such networks hinders high-quality feature learning, making it challenging to directly apply HNNs to student-exercise-knowledge networks to improve trait learning.

To address the above issue, we propose a novel Multi-Channel Graph Contrastive Learning (MCGCL) framework for student performance prediction. Unlike existing methods that focus only on simple pairwise interactions between students and exercises, MCGCL improves trait learning for students and exercises by incorporating structure information of diverse high-order relations via a multi-channel graph contrastive learning framework. The technical contributions of our work can be summarized as follows.

1. MCGCL can effectively incorporate higher-order interactions among students, exercises, and concepts into performance prediction, which helps to enrich the trait features of students and exercises.
2. A new graph contrastive loss is introduced to constrain the trait feature learning for students and exercises, which helps to distinguish the differences of these trait features learned from different graph views.
3. We conduct extensive experiments to evaluate the performance of our MCGCL. The experimental results show that it outperforms all competing baseline methods on the task of predicting student performance.

2 Related Work

2.1 Student Performance Prediction

Previous works modeling student performance are mainly under the framework of cognitive diagnosis, which models the interactions between students' proficiency vectors (i.e., student trait vectors) and exercises' trait vectors by a logistic-like function. How to improve trait learning and interaction modeling is the key for these cognitive diagnosis models to improve their prediction results. In recent years, due to the powerful representation learning ability of DNN, researchers have developed a series of deep neural cognitive diagnosis methods. For example, [Wang *et al.*, 2022] proposed deep neural cognitive diagnosis frameworks to model the complex non-linear interactions by the MLP. To improve trait learning, [Gao *et al.*, 2021] designs a relation map-driven framework to learn relation-aware representations for students and exercises based on a three-layer relation map of the student-exercise-concept hierarchy. Subsequent research has focused on enhancing trait learning by examining the quantitative connections between exercises and concepts [Qi *et al.*, 2023] and the diverse interactions between students and exercises [Wu *et al.*, 2023]. And [Liu *et al.*, 2024] leverages student-centered graph-based relationships for fast new students' mastery levels inference, while [Ma *et al.*, 2024] models the higher-order connectivity in group-level cognitive diagnosis by constructing a Group-Student-Exercise graph based on binary relationships. Although these methods show promising results, they only focus on pairwise relations among students, exercises, and knowledge concepts. This ignores the high-order relations that inherently exist, which could reduce the ability of the learned trait vectors to represent high-level composite semantics.

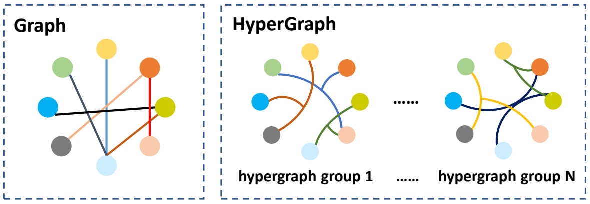


Figure 2: Examples of graphs with pairwise relations connecting two nodes and hypergraphs with high-order relations connecting multiple nodes.

2.2 Graph Structure Modeling

The Heterogeneous Information Network (HIN) consists of multiple types of entities and relationships, and its powerful ability to learn structural representations has led to extensive research in data mining and artificial intelligence tasks. Existing methods can be roughly divided into two groups: 1) traditional network embedding methods [Dong *et al.*, 2017]; 2) GNN based methods[Liu *et al.*, 2023]. For more complex relations, hypergraphs offer a flexible approach to modeling higher-order relationships. In Fig. 2, a hyperedge can connect multiple nodes, extending beyond simple pairwise interactions. To model hypergraph structures effectively, HNN usually implicitly transforms hypergraphs into simple graphs and then applies existing GNN algorithms. Most existing HNNs are mainly focused on homogeneous hypergraphs [Sun *et al.*, 2021]. HyperGAT [Ding *et al.*, 2020] utilizes the attention mechanism for heterogeneous hypergraph representation learning to achieve text classification.

Recent advancements in contrastive learning, particularly Graph Contrastive Learning (GCL), enable self-supervised training on graph data. The consistency between instances is usually measured by mutual information. GCL techniques [Kumar *et al.*, 2022] contrast congruent views (positive pairs) from the same graph instance with incongruent views (negative pairs) from different instances to learn rich graph/node representations. These methods usually learn one or more encoders to make the representations of positive view pairs consistent with each other, while those of negative pairs are dissimilar. Most of the existing GCL methods [You *et al.*, 2020; Xu *et al.*, 2021] focus only on local or global structures of pairwise relations, neglecting the effect of high-order relational structures on contrastive learning. To address this issue, some work [Xia *et al.*, 2021; Yu *et al.*, 2021] introduce contrastive learning into hypergraph learning by developing multi-channel hypergraph convolutional networks. These networks use multiple channels to learn hypergraph encodings, with each channel representing a different graph view. The different channels can acquire new information from each other to improve their ability in feature learning.

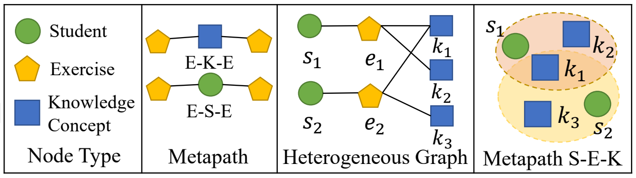


Figure 3: Examples of several metapaths that connect nodes of students, exercises, and knowledge concepts.

3 Preliminaries

Suppose that there are N students, J exercises, and K knowledge concepts. The performance of the n-th student on the j-th exercise is denoted by r_{nj}, where $r_{nj} = 1$ for a correct response and $r_{nj} = 0$ otherwise. The relations between exercises and knowledge concepts are captured by the binary Q matrix $(q_{jk})_{J\times K}$. If the j-th exercise assesses the k-th knowledge concept, $q_{jk} = 1$, otherwise $q_{jk} = 0$. The embeddings of students, exercises, and knowledge concepts are projected into the d-dimensional feature space as:

$$\boldsymbol{S} = \boldsymbol{X}_s^\top \boldsymbol{W}_s, \quad \boldsymbol{E} = \boldsymbol{X}_e^\top \boldsymbol{W}_e, \quad \boldsymbol{K} = \boldsymbol{X}_k^\top \boldsymbol{W}_k, \qquad (1)$$

where $\boldsymbol{X}_s \in \mathbb{R}^{N\times N}$, $\boldsymbol{X}_e \in \mathbb{R}^{J\times J}$ and $\boldsymbol{X}_k \in \mathbb{R}^{K\times K}$ are the one-hot embedding matrices. The trainable parameter matrices $\boldsymbol{W}_s \in \mathbb{R}^{N\times d}$, $\boldsymbol{W}_e \in \mathbb{R}^{J\times d}$ and $\boldsymbol{W}_k \in \mathbb{R}^{K\times d}$ initialize the embeddings for students, exercises, and concepts. The embedding vector for each student is also called student proficiency trait [Reckase, 2009], each dimension of which indicates the proficiency degree on each concept.

Definition 1 (Heterogeneous hypergraph)**.** *Heterogeneous hypergraphs $G_{hyper} = (V, E)$ contain multiple types of nodes or hyperedges, where V is the set of P unique nodes and E is the set of M hyperedges. Each hyperedge ϵ connects at least two nodes and is assigned a positive weight $W_{\epsilon\epsilon}$. The incidence matrix $\boldsymbol{H} \in \mathbb{R}^{P\times M}$ is defined such that $H_{i\epsilon} = 1$ if ϵ contains node v_i, otherwise 0.*

Definition 2 (Metapath)**.** *A metapath ϕ is a path of the form $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$, which describes the composite relation $R = R_1 \circ R_2 \circ \cdots \circ R_l$ between the node types A_1 and A_{l+1}, where $\circ$ denotes the composition operator on relations. Fig. 3 shows some examples of metapaths connecting nodes of students, exercises, and knowledge concepts.*

Definition 3 (Metapath-based hypergraph)**.** *Given a metapath ϕ, we construct a hypergraph G^{ϕ}_{hyper} by treating the node v_i of one type as a hyperedge ϵ_i with nodes connected to v_i through ϕ as the member nodes of ϵ_i. The degree of each node is $\boldsymbol{D}^{\phi}_{ii} = \sum_{\epsilon=1}^{M} \boldsymbol{W}_{\epsilon\epsilon}\boldsymbol{H}_{i\epsilon}$ and the degree of each hyperedge is $\boldsymbol{B}^{\phi}_{\epsilon\epsilon} = \sum_{i=1}^{P} \boldsymbol{H}_{i\epsilon}$, where both of them are diagonal matrices. And different metapaths correspond to different $\boldsymbol{W}^{\phi}_{\epsilon\epsilon}$.*

4 The Proposed Model

The proposed MCGCL framework, shown in Fig. 4, enhances trait learning for students and exercises through multi-channel contrastive learning with hypergraphs and metapaths. It consists of four main modules.

4.1 Module I: Hypergraph-Based Multi-channel Trait Learning (HMT)

In this module, we learn trait vectors for students and exercises through a two-channel hypergraph convolution, with each channel encoding one type of high-order relation derived from the *Exercise-Knowledge_concept (E-K)* and *Student-Knowledge_concept (S-K)* metapaths.

Channel E-K. Given the known Q-matrix, we construct a hypergraph using the $\phi_{E\text{-}K}$ to capture the high-order relations among exercises and knowledge concepts. This part involves learning exercise traits and concept feature vectors.

We denote the hypergraph constructed by $\phi_{E\text{-}K}$ as G^{EK}_{hyper}, which contains two types of nodes and thus has two incidence matrices: 1) $\boldsymbol{H}^{EK}_{v_K}$ treats exercises as hyperedges and knowledge concepts as nodes; 2) $\boldsymbol{H}^{EK}_{v_E}$ treats knowledge concepts as hyperedges and exercises as nodes. Using G^{EK}_{hyper}, we learn the feature representations for knowledge concepts by the spectral hypergraph convolution used in [Bai *et al.*, 2021]:

$$\boldsymbol{K}^{EK^{(l+1)}} = \boldsymbol{D}^{EK^{-1}} \boldsymbol{H}^{EK}_{v_K} \boldsymbol{W}^{EK}_{\epsilon\epsilon} \boldsymbol{B}^{EK^{-1}} \boldsymbol{H}^{EK^{\top}}_{v_K} \boldsymbol{X}^{EK^{(l)}}, \quad (2)$$

where $\boldsymbol{K}^{EK^{(l+1)}}$ denotes the learned feature vectors for K concepts at the $l+1$ layer, $\boldsymbol{D}^{EK}$ and $\boldsymbol{B}^{EK}$ are degree matrices of nodes and hyperedges. $\boldsymbol{X}^{EK^{(0)}}$ is the initial feature of concept $\boldsymbol{K}$. The convolution process can be seen as a two-step process of performing the node-hyperedge-node feature transformation. First, aggregating information from nodes to hyperedges using $\boldsymbol{H}^{EK^{\top}}_{v_K} \boldsymbol{X}^{EK^{(l)}}$; then aggregating from hyperedges to knowledge concept nodes by pre-multiplying with $\boldsymbol{H}^{EK}_{v_K}$. Similar to $\boldsymbol{K}^{EK}$, exercise trait vectors $\boldsymbol{E}^{EK} = \{\boldsymbol{e}^{EK}_j\}_{j=0}^{J}$ based on G^{EK}_{hyper} can be learned.

Channel S-K. Similar to Channel E-K, this part learns trait representations for students and concepts based on the hypergraph G^{SK}_{hyper} constructed from the $\phi_{S\text{-}K}$. This hypergraph captures high-order relations among students and knowledge concepts. Since the groundtruth for *S-K* is unavailable, we obtain it from the *Student-Exercise-Knowledge (S-E-K)* metapath, using the exercise as the intermediate. This is reasonable because a student's mastery of a concept can be inferred from their response to exercises that assess the concept.

The hypergraph G^{SK}_{hyper} contains two types of nodes: student and concept nodes, leading to two incidence matrices. First, $\boldsymbol{H}^{SK}_{v_K}$ treats student nodes as hyperedges and concept nodes as their members. Second, $\boldsymbol{H}^{SK}_{v_S}$ treats concept nodes as hyperedges and student nodes as their members. Specifically, we can learn the feature matrix $\boldsymbol{K}^{SK^{(l+1)}}$ by

$$\boldsymbol{K}^{SK^{(l+1)}} = \boldsymbol{D}^{SK^{-1}} \boldsymbol{H}^{SK}_{v_K} \boldsymbol{W}^{SK}_{\epsilon\epsilon} \boldsymbol{B}^{SK^{-1}} \boldsymbol{H}^{SK^{\top}}_{v_K} \boldsymbol{X}^{SK^{(l)}}, \quad (3)$$

where $\boldsymbol{D}^{SK}$ and $\boldsymbol{B}^{SK}$ denote the degree matrices of nodes and hyperedges in the hypergraph constructed by $\phi_{S\text{-}E\text{-}K}$. The trait vectors for students $\boldsymbol{S}^{SK^{(l+1)}}$ can also be learned by the hypergraph convolution based on G^{SK}_{hyper}.

Since the importance of the nodes in a hyperedge is usually different from each other, we impose an attention learning on the incidence matrix $\boldsymbol{H}_{v_z}$ to measure the connection

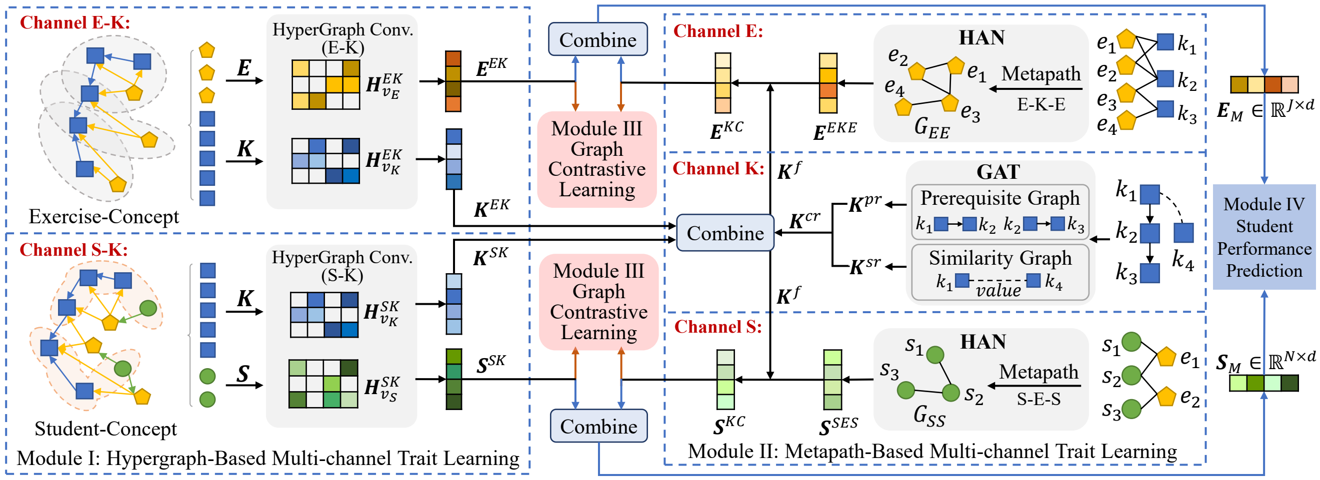


Figure 4: The framework of our proposed MCGCL.

degree (or importance) of one node to a hyperedge. Specifically, we denote the attention weights of v_i for hyperedge ϵ_j by $\boldsymbol{H}^{\phi}_{v_z}(i,j)$, which can be learned by

$$\boldsymbol{H}^{\phi}_{v_z}(i,j) = \frac{\exp(\sigma(\boldsymbol{b}^{\top}[\boldsymbol{x}_i\boldsymbol{W}_a||\boldsymbol{e}_j\boldsymbol{W}_b]))}{\sum_{k\in\mathcal{N}_i}\exp(\sigma(\boldsymbol{b}^{\top}[\boldsymbol{x}_i\boldsymbol{W}_a||\boldsymbol{e}_k\boldsymbol{W}_b]))}, \quad (4)$$

where $\boldsymbol{x}_i$ denotes the feature of node v_i and $\boldsymbol{e}_j$ is the feature of the hyperedge ϵ_j it connects to. $\mathcal{N}_i$ is the set of all neighbor nodes within ϵ_j connected to v_i. $||$ denotes the concatenation to retain both sets of information, and $\boldsymbol{b}$ is a learnable vector, allowing the model to flexibly capture intricate interactions.

Finally, we multiply $\boldsymbol{H}^{\phi}_{v_z}(i,j)$ and $\boldsymbol{H}^{EK}_{v_K}$ in Eq. (2), and with $\boldsymbol{H}^{SK}_{v_K}$ in Eq.(3) to obtain non-binary incidence matrices.

4.2 Module II: Metapath-Based Multi-channel Trait Learning (MMT)

This module integrates structural information from diverse composite relations into trait representation learning through three channels, each corresponding to a specific metapath. Channel E learns exercise trait vectors based on the *Exercise-Knowledge-Exercise (E-K-E)* metapath, Channel S learns student trait vectors based on the *Student-Exercise-Student (S-E-S)* metapath, and Channel K learns concept features.

Channel E. We first construct an exercise relation graph G_{EE} based on the $\phi_{\textit{E-K-E}}$. Then, we apply the heterogeneous graph attention network [Wang *et al.*, 2019] on G_{EE} to learn the exercise trait vectors, denoted by $\boldsymbol{E}^{EKE} = \{\boldsymbol{e}^{EKE}_j\}^{J}_{j=1}$. Specifically, for exercise node $\boldsymbol{e}_j$, we learn its trait vector $\boldsymbol{e_j}^{EKE}$ by aggregating the trait vectors of metapath-based neighbors of $\boldsymbol{e}_j$ with the corresponding attention coefficients

$$\boldsymbol{e}^{EKE^{(l+1)}}_j = \sigma[\sum_{i\in\mathcal{N}^{EKE}_j}\alpha^{EKE}_{ji}\boldsymbol{e}_i{}^{(l)}], \quad (5)$$

where $\boldsymbol{e}^{EKE}_j$ is the feature vector for exercise e_j. The $\mathcal{N}^{EKE}_j$ denotes the set of metapath-based neighbors of e_j, in which the nodes are connected to e_j via the $\phi_{\textit{E-K-E}}$. α^{EKE}_{ji} denotes the importance of node e_i to node e_j, representing the edge weights between them based on $\phi_{\textit{E-K-E}}$, which can be learned through the metapath-based node-level attention mechanism in [Wang *et al.*, 2019].

Formally, given a node pair (i,j) connected via metapath ϕ, the node-level attention u^{ϕ}_{ij} can be represented by

$$u^{\phi}_{ji} = att(\boldsymbol{x}_j,\boldsymbol{x}_i;\phi) = \sigma\left(\boldsymbol{z}^{\top}_{\phi}\left[\boldsymbol{x}_j||\boldsymbol{x}_i\right]\right), \quad (6)$$

where u^{ϕ}_{ji} denotes the importance weights of node i to node j based on matapath ϕ. The att denotes the neural network that performs attention learning. After obtaining u^{ϕ}_{ji} for all meta-path based neighbors of node j, we normalize them to get the weight coefficient α^{ϕ}_{ji} for node i via softmax function.

Channel S. Similar to Channel E, we first construct a student relation graph based on $\phi_{\textit{S-E-S}}$, denoted by G_{SS}. Then, we learn exercise trait features by using the same graph convolution operations [Wang *et al.*, 2019] on G_{SS}. The feature matrix for all N students is denoted by $\boldsymbol{S}^{SES} = \{\boldsymbol{s}^{SES}_n\}^{N}_{n=1}$, which can be learned as Eq. (5).

Channel K. This channel encodes the structure information of knowledge concept graphs. We consider two types of classical concept dependency relations–prerequisite and similarity relations–which have been widely used in concept graph learning [Zhang *et al.*, 2022]. Since these relations carry distinct semantics, learning features separately from each graph is beneficial. Specifically, we apply graph attention convolutions on prerequisite and similarity graphs to learn the concept features $\boldsymbol{K}^{pr}$ and $\boldsymbol{K}^{sr}$, and combine them to obtain the final knowledge concept features $\boldsymbol{K}^{cr} = \{\boldsymbol{k}^{cr}_c\}^{K}_{c=1}$.

Different knowledge concepts have varying importance in answering questions, and each dimension of these trait vectors reflects the proficiency of students and discrimination factors for each concept. In Fig. 5, we present the framework for updating the trait features under the guidance of concepts. We first learn comprehensive semantic features $\boldsymbol{K}^{f}$ for knowledge concepts by combining the concept features learned in different settings of graph structures. $\boldsymbol{K}^{EK}$ and

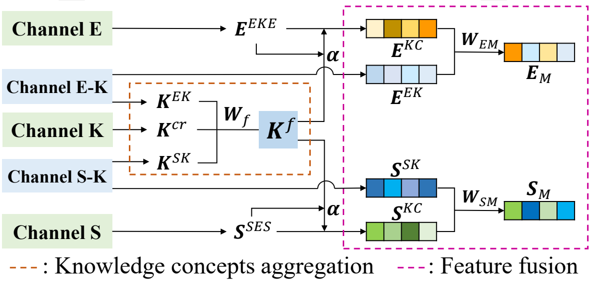


Figure 5: The illustrations for combining the concept features and the trait features learned from different channels.

$\boldsymbol{K}^{SK}$ capture the high-order relations based on $\phi_{E\text{-}K}$ and $\phi_{S\text{-}K}$, respectively. $\boldsymbol{K}^r$ learned in Module II captures the concept dependency relations. The $\boldsymbol{K}^f = \boldsymbol{k}_c^f{}_{c=1}^K$ can be obtained by

$$\boldsymbol{K}^f = LeakyRelu\left(\boldsymbol{W}_f\left(\boldsymbol{K}^{EK}||\boldsymbol{K}^{SK}||\boldsymbol{K}^{cr}\right)\right). \quad (7)$$

With $\boldsymbol{K}^f$, we learn the importance weight for each concept by estimating the attention of exercise e_j on each k_c, denoted by α_{jc}. The computation of α_{jc} can be formulated by

$$\boldsymbol{\alpha}_{jc} = softmax\left(\boldsymbol{e}_j^{EKE}\boldsymbol{W}_{eke}\left(q_{jc}\boldsymbol{k}_c^f\boldsymbol{W}_k\right)^\top\right), \quad (8)$$

where $q_{jc} \in \{0, 1\}$ denotes the examination relations between exercises and concepts. The feature vector for exercise $\boldsymbol{e}_j^{EKE}$, can be obtained by Eq. (5). Thus, the alignment between the features of exercises and concepts is measured via a dot product. This score is then converted into an importance weight. Considering the importance weight α_{jc} of concept k_c for exercise e_j, we update exercise features by

$$\boldsymbol{e}_j^{KC} = \boldsymbol{\alpha}_j \odot \boldsymbol{e}_j^{EKE}, \quad (9)$$

where $\boldsymbol{\alpha}_j = (\alpha_{j1}, \alpha_{j2}, \cdots, \alpha_{jK})$, $\boldsymbol{E}^{KC} = \{\boldsymbol{e}_j^{KC}\}_{j=1}^J$ and $\boldsymbol{e}_j^{KC} \in \mathbb{R}^K$ denotes the exercise feature after update. The symbol $\odot$ denotes element-wise product between vectors.

Similarly, by taking into account the importance of each concept, we can also update student trait features $\boldsymbol{S}^{KC}$ by the way described in Eq. (8) and (9).

4.3 Module III: Graph Contrastive Learning

Inspired by GCL works [Tian, 2022; Huang *et al.*, 2022], it is beneficial to improve node feature learning by distinguishing the differences among different graph views to obtain useful structure information of each graph. Besides, GCL also helps to alleviate the sparsity problem of the interactions among entities by encouraging consistent features across graph views.

We use contrastive learning to pull together the different views of the same node and push apart those of different nodes. The loss for optimizing the features of the student nodes and the exercise nodes is denoted by L_{ss}^S and L_{ss}^E:

$$L_{ss}^S = -\log\sigma\left(f_D\left(\boldsymbol{s}_n^{KC}, \boldsymbol{s}_n^{SK}\right)\right) - \log\sigma\left(1 - f_D\left(\tilde{\boldsymbol{s}}_i^{KC}, \boldsymbol{s}_n^{SK}\right)\right), \quad (10)$$

$$L_{ss}^E = -\log\sigma\left(f_D\left(\boldsymbol{e}_j^{KC}, \boldsymbol{e}_j^{EK}\right)\right) - \log\sigma\left(1 - f_D\left(\tilde{\boldsymbol{e}}_i^{KC}, \boldsymbol{e}_j^{EK}\right)\right). \quad (11)$$

The $\tilde{\boldsymbol{s}}_i^K$ and $\tilde{\boldsymbol{e}}_i^K$ denote the features of the negative sample nodes of the student node s_n and the exercise node e_j. For example, the negative samples of s_i are the student nodes other than s_n. The $f_D(\cdot)$ denotes the distinguisher used to compute the consistency scores of the two vectors, which is implemented as an inner product operation between vectors.

4.4 Module IV: Student Performance Prediction

This part predicts the student performance based on interactions between students and exercises, where the ground truth is binary: 1 for correct and 0 for incorrect responses. We first fuse the features learned by Module I and II to obtain comprehensive trait features for both students and exercises.

Student feature fusion. We learn comprehensive student trait features by combining the trait features learned from different channels, which can be formulated by

$$\boldsymbol{S}_M = \sigma\left(\left(\boldsymbol{S}^{SK}||\boldsymbol{S}^{KC}\right)\cdot \boldsymbol{W}_{SM}^\top\right). \quad (12)$$

The $\boldsymbol{S}^{SK}$ and $\boldsymbol{S}^{KC}$ encode different high-order student relations, which are derived from the corresponding hypergraph structure in Module I and the metapath-based graph structure in Module II, respectively. $\boldsymbol{S}_M \in \mathbb{R}^{N\times d}$ denotes the obtained trait feature matrix for students, and $\boldsymbol{W}_{SM} \in \mathbb{R}$ denotes a trainable parameter matrix.

Exercise feature fusion. Similarly, we learn comprehensive exercise trait features $\boldsymbol{E}_M \in \mathbb{R}^{J\times d}$ by combining the trait features learned from different channels, i.e., $\boldsymbol{E}^{EK}$ and $\boldsymbol{E}^{KC}$. Specifically, the learning of $\boldsymbol{E}_M$ can be formulated by

$$\boldsymbol{E}_M = \sigma\left(\left(\boldsymbol{E}^{EK}||\boldsymbol{E}^{KC}\right)\cdot \boldsymbol{W}_{EM}^\top\right). \quad (13)$$

Finally, we input the learned $\boldsymbol{S}_M$ and $\boldsymbol{E}_M$ into a multi-layer perceptron to predict the probability of students giving correct answers by

$$\mathbf{Y} = MLP\left(\boldsymbol{S}_M\boldsymbol{W}_p\boldsymbol{E}_M^\top\right), \quad (14)$$

where $\boldsymbol{W}_p \in \mathbb{R}^{K\times K}$ is the trainable matrix and $\mathbf{Y} = \{y_{nj}\} \in \mathbb{R}^{N\times J}$ are the predicted probability scores for all N students on J exercises. During the training phase, a cross-entropy is used to estimate the loss between predicted results and the groundtruth, which can be formulated by

$$L_{tr} = -\sum_{n=1}^N\sum_{j=1}^J \tilde{y}_{nj}\log(y_{nj}) + (1-\tilde{y}_{nj})\log(1-y_{nj}), \quad (15)$$

where $\tilde{y}_i$ denotes the groundtruth performance for the n-th student on the j-th exercise.

Overall, we use both the above cross-entropy loss and the contrastive loss for self-supervised feature learning . Specifically, the joint learning loss objective can be formulated by

$$L = L_{tr} + \gamma L_{ss}^S + \beta L_{ss}^E, \quad (16)$$

where γ and β are used to control the importance of the self-supervised task in the overall task.

5 Experiments

5.1 Datasets

We conduct experiments on two benchmarks: Junyi[1] [Chang *et al.*, 2015] and ASSIST[2] [Razzaq *et al.*, 2005], both of

[1] https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=1198
[2] https://sites.google.com/site/assistmentsdata/home/2009-2010-assistment-data

Dataset	Junyi	ASSIST
#Students	10,000	2,493
#Exercises	835	17,746
#Knowledge concepts	835	123
#Response records	353,835	267,415
#Knowledge concepts per exercise	1.0	1.2
#Response records per student	35.38	107.27
#Prerequisite relations	988	1,164
#Similarity relations	1,040	1,256

Table 1: The statistics of datasets.

which provide students' exercise performance records and a Q-matrix indicating the relations between exercises and knowledge concepts. To ensure sufficient data for diagnosis, we exclude students with fewer than 15 response records in each dataset. The statistics of both datasets are shown in Table 1. Junyi dataset provides the concept dependency relations annotated by experts, which contain prerequisite and similarity relations. Specifically, the prerequisite relations between concept pairs (k_i, k_j) indicate the former is the prerequisite knowledge of mastering the latter. The similarity relation is annotated by the triplet $(k_i, k_j, value)$, where $value$ denotes the strength of similarity between two knowledge concepts. As ASSIST does not explicitly provide dependency relations between knowledge concepts, we follow the baselines [Gao *et al.*, 2021] to construct concept graphs with prerequisite and similarity relations based on certain statistics.

5.2 Experimental Setup

During training, the batch size for dataset is set to 9000 with a node feature dimension of 200. The l_2 regularization parameter is 10^{-5}, and the learning rate is 0.0001. In our experiments, we set $\gamma = \beta = 1$ to give equal importance to the overall loss function. All competing methods are implemented with PyTorch, and experiments are conducted on an NVIDIA RTX 4090 GPU. To evaluate the effectiveness of MCGCL, we compare it against state-of-the-art performance prediction methods, which can be divided into two groups. One group of baselines is Cognitive Diagnosis Models (CDMs). Another group focuses on graph-based approaches, which utilize structural and relational information embedded in graphs to enhance performance.

5.3 Performance Comparison

In Table 2, we present the performance results of all competing methods. Several important insights can be drawn from these results. First, MCGCL consistently outperforms both previous CDMs and GNN-based baselines on all metrics. It indicates that incorporating diverse high-order interactions among students, exercises, and concepts into performance prediction is beneficial for achieving the best results. This demonstrates the effectiveness of our MCGCL in improving trait learning for students and exercises by modeling the diverse high-order interactions under the framework of multi-channel GCL. In contrast, existing methods usually treat each student and each exercise independently when learning their corresponding trait features. This weakens the ability of the learned trait features to capture the high-order interactions between students and exercises and thus impedes performance improvement. Second, almost all graph learning-based methods perform better than CDMs that do not consider complex interactions. This is because these CDMs mostly only take into account pairwise student-exercise interactions, overlooking the diverse high-order interactions (e.g., student-exercise-student and exercise-concept-exercise interactions) that carry complex semantics.

5.4 Ablation Study

Effectiveness Analysis of MCGCL Modules

To explore the effectiveness of each module in our MCGCL, we test the performance of our MCGCL with different configurations. MCGCL-P, MCGCL-H, and MCGCL-GC are obtained by removing module I, II and III, respectively.

The experimental results of the variants are also reported in Table 2. The performance of variant *MCGCL-P* shows the largest performance degeneration, e.g., 1.29% on *AUC* for ASSIST. This demonstrates the necessity to improve trait learning by encoding high-order student interactions and high-order exercise interactions through the hypergraph structures in Module *HMT*. Without encoding graph structures of composite relations in Module *MMT*, variant *MCGCL-H* witnesses a performance drop, which demonstrates the effectiveness of encoding composite relations between students and between exercises. We also observe that the performance of variant *MCGCL-GC* degenerates on all datasets, which indicates the effectiveness of GCL in improving trait learning. This is because contrastive learning between different channels helps to distinguish node feature differences across various graph views, which enhances node feature learning via useful structure information of each graph.

Comparative Analysis of Different Prediction Modules

In our MCGCL, we predict student performance directly by the inner product of the trait features of students and exercises, which is different from the IRT-like scoring functions used in previous cognitive diagnosis methods. Therefore, we also conduct some experiments to investigate the effectiveness of our scoring functions based on the inner product of the trait features in predicting student performance. Specifically, we test the performance of MCGCL with the following configurations: *MCGCL-IRT* is obtained by replacing the scoring function in Eq. (14) with the IRT function. *MCGCL-MIRT* is obtained by replacing our scoring function with the MIRT function. *MCGCL-NCD* is obtained by replacing our scoring function with the diagnosis layer of NCD.

From Fig. 6, we can see that MCGCL outperforms all variant models, achieving at least 0.97% improvement in AUC for Junyi and 1.86% improvement in AUC for ASSIST. This is because the trait features of students and exercises learned by our MCGCL can effectively capture rich high-order and composite interactions between students and exercises. In this case, MCGCL does not require manually designed IRT-like linear functions to model student-exercise interactions, which may degenerate the performance prediction results by imposing ill-fitting linear interaction constraints.

Preprint – IJCAI 2025: This is the accepted version made available for conference attendees.
Do not cite. The final version will appear in the IJCAI 2025 proceedings.

Methods	Junyi			ASSIST		
	ACC	AUC	RMSE	ACC	AUC	RMSE
IRT [Lord, 2012]	67.60	77.50	42.68	64.26	69.83	46.59
MIRT [Yao and Schwarz, 2006]	75.13	79.89	41.17	71.70	74.94	45.17
PMF [Mnih and Salakhutdinov, 2007]	68.34	76.44	43.73	67.12	76.45	44.51
NCD [Wang *et al.*, 2020]	74.43	79.09	41.72	73.14	75.94	43.08
KSCD [Ma *et al.*, 2022]	77.83	82.23	39.12	73.51	76.36	41.40
SCD [Shen *et al.*, 2024]	74.86	76.69	43.91	-	-	-
Metapath2vec [Dong *et al.*, 2017]	75.40	80.52	40.87	72.13	73.30	42.66
GTN [Yun *et al.*, 2019]	74.63	79.30	41.43	73.01	76.58	42.43
HAN [Wang *et al.*, 2019]	76.29	81.53	40.31	73.13	76.46	42.43
RCD [Gao *et al.*, 2021]	77.16	82.62	39.63	73.55	77.21	42.13
KaNCD [Wang *et al.*, 2022]	75.40	77.50	42.20	73.20	76.40	42.40
ICD [Qi *et al.*, 2023]	-	-	-	73.83	77.27	41.90
MCGCL-P	76.69	82.28	39.74	72.91	76.59	41.13
MCGCL-H	77.03	82.44	38.95	72.98	77.32	41.02
MCGCL-GC	77.86	83.06	**38.83**	73.62	77.51	40.79
MCGCL	**78.02**	**83.29**	39.41	**74.02**	**77.88**	**40.35**

Table 2: Experimental results on student performance prediction (%). The best results are denoted in bold.

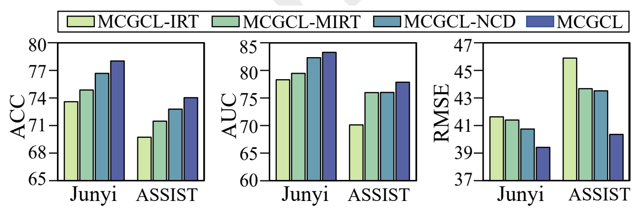
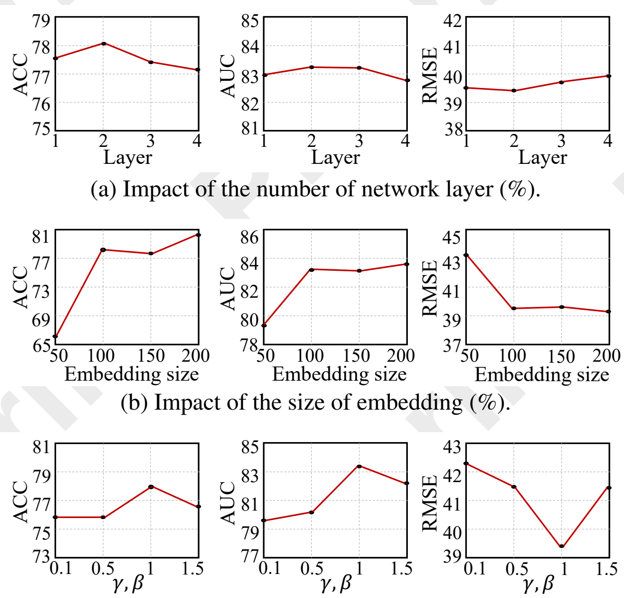


Figure 6: Ablation experiment results of the prediction module(%).

5.5 Hyper-Parameters Sensitivity Analysis

Key hyperparameters in MCGCL encompass the number of graph convolution layers in Module II, the sizes of the student and exercise embeddings as determined by Eq.(12) and Eq.(13), along with the parameters γ and β, which serve as the weighting factors for the loss outlined in Eq.(16). We investigates how these hyperparameters affect MCGCL's performance and evaluates its robustness using the Junyi dataset.

As shown in Fig. 7a shows that altering the number of layers in the neural network limitedly affects the model's performance on both datasets. Optimal results are achieved with a two-layer structure, whereas more complex networks often suffer from overfitting, leading to a decline of performance. Fig. 7b, enlarging the embedding size enhances prediction performance, suggesting that a larger embedding size aids in encoding complex knowledge information effectively. Nonetheless, this enlargement in embedding size also leads to a substantial rise in the model's number of parameters, thereby requiring more computational power. Consequently, selecting the embedding size involves balancing enhanced performance against the need for resources. As illustrated in Fig. 7c, setting γ and β to values less than 1 leads to inadequate utilization of graph contrastive learning by the model, causing underfitting. Nevertheless, when γ and β take on large values, the model is excessively oriented towards contrastive learning, resulting in bias and overfitting issues. Adjusting these parameters adequately is essential for achieving an optimized balance.



(a) Impact of the number of network layer (%).

(b) Impact of the size of embedding (%).

(c) Impact of γ and β (%).

Figure 7: Experiment results on hyper-parameter sensitivity.

6 Conclusion

In this paper, we present an innovative MCGCL framework that integrates a variety of higher-order interactions between students and exercises to improve student performance prediction. MCGCL employs diverse channels to encode numerous high-order or composite relationships from different graph views, while graph contrastive learning enriches the trait features acquired from these views. Comprehensive tests on actual benchmark datasets reveal that MCGCL attains leading-edge performance with exceptional efficiency. In future studies, efforts could could focus on enhancing computational efficiency and integrating temporal dynamics to better reflect shifts in student learning over time.

Acknowledgments

The research was supported in part by Guangdong Basic and Applied Basic Research Foundation (2024A1515011715, 2025A1515012995), Key Program of National Natural Science Foundation of China (62433016), and CCF-Zhipu Large Model Innovation Fund (CCF-Zhipu202413).

References

[Bai *et al.*, 2021] Song Bai, Feihu Zhang, and Philip HS Torr. Hypergraph convolution and hypergraph attention. *Pattern Recognition*, 110:107637, 2021.

[Chang *et al.*, 2015] Haw-Shiuan Chang, Hwai-Jung Hsu, and Kuan-Ta Chen. Modeling exercise relationships in e-learning: A unified approach. In *EDM*, pages 532–535, 2015.

[Ding *et al.*, 2020] Kaize Ding, Jianling Wang, Jundong Li, Dingcheng Li, and Huan Liu. Be more with less: Hypergraph attention networks for inductive text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4927–4936, 2020.

[Dong *et al.*, 2017] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD '17*, pages 135–144. ACM, 2017.

[Gao *et al.*, 2021] Weibo Gao, Qi Liu, Zhenya Huang, Yu Yin, Haoyang Bi, Mu-Chun Wang, Jianhui Ma, Shijin Wang, and Yu Su. 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*, pages 501–510, 2021.

[Huang *et al.*, 2022] Mingyuan Huang, Pengpeng Zhao, Xuefeng Xian, Jianfeng Qu, Guanfeng Liu, Yanchi Liu, and Victor S Sheng. Help from meta-path: Node and meta-path contrastive learning for recommender systems. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 01–08, 2022.

[Kumar *et al.*, 2022] Pranjal Kumar, Piyush Rawat, and Siddhartha Chauhan. Contrastive self-supervised learning: review, progress, challenges and future research directions. *International Journal of Multimedia Information Retrieval*, 11(4), 2022.

[Liu *et al.*, 2023] Jie Liu, Lingyun Song, Guangtao Wang, and Xuequn Shang. Meta-hgt: Metapath-aware hypergraph transformer for heterogeneous information network embedding. *Neural Networks*, 157:65–76, 2023.

[Liu *et al.*, 2024] Shuo Liu, Junhao Shen, Hong Qian, and Aimin Zhou. Inductive cognitive diagnosis for fast student learning in web-based online intelligent education systems. In *Proceedings of the ACM Web Conference 2024*, Singapore, 2024.

[Lord, 2012] Frederic M Lord. *Applications of item response theory to practical testing problems*. Routledge, 2012.

[Ma *et al.*, 2022] Haiping Ma, Manwei Li, Le Wu, Haifeng Zhang, Yunbo Cao, Xingyi Zhang, and Xuemin Zhao. Knowledge-sensed cognitive diagnosis for intelligent education platforms. In Mohammad Al Hasan and Li Xiong, editors, *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pages 1451–1460. ACM, 2022.

[Ma *et al.*, 2024] Haiping Ma, Siyu Song, chuan Qin, Xiaoshan Yu, Limiao Zhang, Xingyi Zhang, and Hengshu Zhu. Dgcd: An adaptive denoising gnn for group-level cognitive diagnosis. In *Proceedings of the 33th International Joint Conference on Artificial Intelligence*, 2024.

[Mnih and Salakhutdinov, 2007] Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. *Advances in neural information processing systems*, 20, 2007.

[Qi *et al.*, 2023] Tianlong Qi, Meirui Ren, Longjiang Guo, Xiaokun Li, Jin Li, and Lichen Zhang. Icd: A new interpretable cognitive diagnosis model for intelligent tutor systems. *Expert Systems with Applications*, 215:119309, 2023.

[Razzaq *et al.*, 2005] Leena Razzaq, Mingyu Feng, Goss Nuzzo-Jones, Neil T Heffernan, KR Koedinger, Brian Junker, Steven Ritter, Andrea Knight, C Aniszczyk, S Choksey, et al. The assistment project: Blending assessment and assisting. In *Proceedings of the 12th annual conference on artificial intelligence in education*, pages 555–562, 2005.

[Reckase, 2009] Mark D Reckase. Multidimensional item response theory models. *Multidimensional item response theory*, pages 79–112, 2009.

[Shen *et al.*, 2024] Junhao Shen, Hong Qian, Wei Zhang, and Aimin Zhou. Symbolic cognitive diagnosis via hybrid optimization for intelligent education systems. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, Vancouver, Canada, 2024.

[Sun *et al.*, 2021] Xiangguo Sun, Hongzhi Yin, Bo Liu, Hongxu Chen, Jiuxin Cao, Yingxia Shao, and Nguyen Quoc Viet Hung. Heterogeneous hypergraph embedding for graph classification. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 725–733, 2021.

[Tian, 2022] Quanhong Tian. Multi-channel contrastive learning for sequential recommendation. In *Proceedings of the 8th International Conference on Systems and Informatics*, pages 1–6, 2022.

[Wang *et al.*, 2019] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *Proceedings of the World Wide Web Conference*, pages 2022–2032, 2019.

[Wang *et al.*, 2020] Fei Wang, Qi Liu, Enhong Chen, Zhenya Huang, Yuying Chen, Yu Yin, Zai Huang, and Shijin Wang. Neural cognitive diagnosis for intelligent education systems. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 6153–6161, 2020.

[Wang *et al.*, 2022] Fei Wang, Qi Liu, Enhong Chen, Zhenya Huang, Yu Yin, Shijin Wang, and Yu Su. Neuralcd: A general framework for cognitive diagnosis. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[Wu *et al.*, 2023] Kaifang Wu, Yonghui Yang, Kun Zhang, Le Wu, Jing Liu, and Xin Li. Multi-relational cognitive diagnosis for intelligent education. In *The proceedings of the CAAI International Conference on Artificial Intelligence*, pages 425–437, 2023.

[Xia *et al.*, 2021] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. Self-supervised hypergraph convolutional networks for session-based recommendation. In *AAAI Conference on Artificial Intelligence*, pages 4503–4511, 2021.

[Xu *et al.*, 2021] Minghao Xu, Hang Wang, Bingbing Ni, Hongyu Guo, and Jian Tang. Self-supervised graph-level representation learning with local and global structure. In *International Conference on Machine Learning*, pages 11548–11558, 2021.

[Yao and Schwarz, 2006] Lihua Yao and Richard D Schwarz. A multidimensional partial credit model with associated item and test statistics: An application to mixed-format tests. *Applied psychological measurement*, 30(6):469–492, 2006.

[You *et al.*, 2020] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *Conference on Neural Information Processing Systems*, pages 5812–5823, 2020.

[Yu *et al.*, 2021] Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Nguyen Quoc Viet Hung, and Xiangliang Zhang. Self-supervised multi-channel hypergraph convolutional network for social recommendation. In *The Web Conference*, pages 413–424, 2021.

[Yun *et al.*, 2019] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in neural information processing systems*, 32, 2019.

[Zhang *et al.*, 2022] Juntao Zhang, Nanzhou Lin, Xuelong Zhang, Wei Song, Xiandi Yang, and Zhiyong Peng. Learning concept prerequisite relations from educational data via multi-head attention variational graph auto-encoders. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 1377–1385, 2022.