Capturing Homogeneous Influence among Students: Hypergraph Cognitive Diagnosis for Intelligent Education Systems

Junhao Shen
shenjh@stu.ecnu.edu.cn
School of Computer Science and Technology, East China Normal University
Shanghai, China

Hong Qian*
hqian@cs.ecnu.edu.cn
School of Computer Science and Technology, and Shanghai Institute of AI Education, East China Normal University
Shanghai, China

Shuo Liu
shuoliu@stu.ecnu.edu.cn
School of Computer Science and Technology, East China Normal University
Shanghai, China

Wei Zhang
zhangwei.thu2011@gmail.com
School of Computer Science and Technology, and Shanghai Institute of AI Education, East China Normal University
Shanghai, China

Bo Jiang
bjiang@deit.ecnu.edu.cn
School of Computer Science and Technology, and Shanghai Institute of AI Education, East China Normal University
Shanghai, China

Aimin Zhou
amzhou@cs.ecnu.edu.cn
School of Computer Science and Technology, and Shanghai Institute of AI Education, East China Normal University
Shanghai, China

Abstract

Cognitive diagnosis is a vital upstream task in intelligent education systems. It models the student-exercise interaction, aiming to infer the students' proficiency levels on each knowledge concept. This paper observes that most existing methods can hardly effectively capture the homogeneous influence due to its inherent complexity. That is to say, although students exhibit similar performance on given exercises, their proficiency levels inferred by these methods vary significantly, resulting in shortcomings in interpretability and efficacy. Given the complexity of homogeneous influence, a hypergraph could be a choice due to its flexibility and capability of modeling high-order similarity which aligns with the nature of homogeneous influence. However, before incorporating hypergraph, one at first needs to address the challenges of distorted homogeneous influence, sparsity of response logs, and over-smoothing. To this end, this paper proposes a hypergraph cognitive diagnosis model (HyperCDM) to address these challenges and effectively capture the homogeneous influence. Specifically, to avoid distortion, HyperCDM employs a divide-and-conquer strategy to learn student, exercise and knowledge representations in their own hypergraphs respectively, and interconnects them via a feature-based interaction function. To construct hypergraphs based on sparse response logs, the auto-encoder is utilized to preprocess response logs and K-means is applied to cluster students. To mitigate over-smoothing, momentum hypergraph convolution networks are designed to partially keep previous representations during the message propagation. Extensive experiments on both offline and online real-world datasets show that HyperCDM achieves state-of-the-art performance in terms of interpretability and capturing homogeneous influence effectively, and is competitive in generalization. The ablation study verifies the efficacy of each component, and the case study explicitly showcases the homogeneous influence captured by HyperCDM.

CCS Concepts

• **Applied computing** → Education; • **Computing methodologies** → Machine learning.

Keywords

Student proficiency inference, Cognitive diagnosis, Homogeneous influence, Hypergraph

ACM Reference Format:
Junhao Shen, Hong Qian, Shuo Liu, Wei Zhang, Bo Jiang, and Aimin Zhou. 2024. Capturing Homogeneous Influence among Students: Hypergraph Cognitive Diagnosis for Intelligent Education Systems. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3637528.3672002

1 Introduction

Cognitive diagnosis [25, 29] is a fundamental upstream task in intelligent education systems [2], and extensive applications are found in areas such as exercise recommendation [18] and computerized adaptive testing [46]. An illustrative example is depicted in Figure 1. Generally, students are tasked with completing some exercises (e.g., e_1, e_2, e_3, e_4), and educators can collect response logs. The main object of cognitive diagnosis is to discover the students' proficiency levels on specific knowledge concepts (e.g., linear processing) and exercises' features (e.g., difficulty) via the analysis of these response logs, which can be used in various downstream tasks.

*Hong Qian is the corresponding author.

KDD '24, August 25–29, 2024, Barcelona, Spain

ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3672002

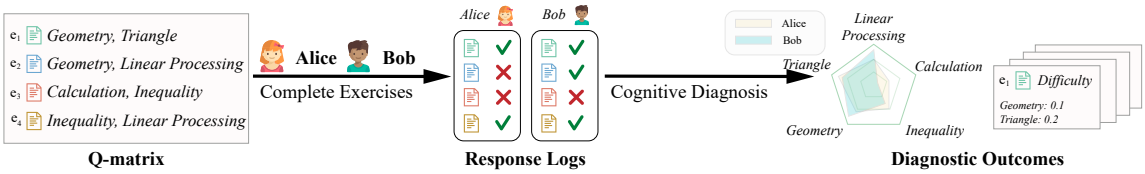


Figure 1: An illustrative example of cognitive diagnosis.

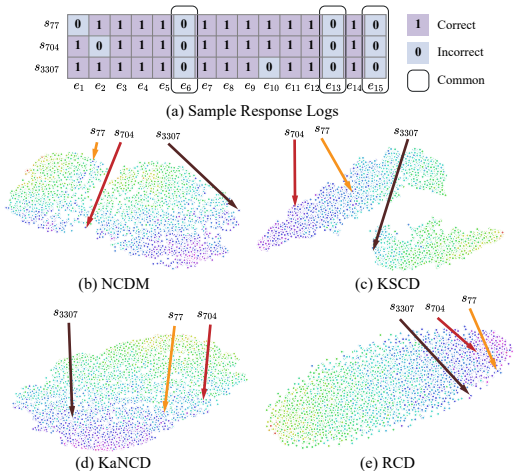


Figure 2: The motivation example and preliminary study results. The sampled response logs containing three similar students whose IDs are s_{77}, s_{704} and s_{3307}. Their inferred proficiency levels obtained by different CDMs on the real-world dataset Math1 are visualized by t-SNE. Each point represents a student and is assigned color ranging from red to purple based on the correctness rates of their answers. Students with similar correctness rates have similar colors, while with similar inferred proficiency levels are positioned close.

In the past decades, considerable efforts have been invested in th e development of cognitive diagnosis models (CDMs), which model the student-exercise interaction as a function, i.e., interaction function. For example, traditional CDMs [7, 30, 36] incorporate manually-designed interaction functions based on expert experience. In contrast, neural network-based CDMs [27, 39, 40] leverage the multi-layer perceptron to enhance the generalization and interpretability of diagnostic outcomes. Another approach, symbolic CDM [34], introduces symbolic tree as an explicit representation of non-linear intricate student-exercise interactions. Besides, some CDMs utilizes large language model [23] to diagnose student based on ample prior knowledge. Recently, a promising direction has emerged that utilizes graph neural networks (GNNs) to learn the structure of student-exercise-knowledge relation graphs, providing valuable insights for cognitive diagnosis. For example, relation map driven cognitive diagnosis model (RCD) [11] establishes relation graphs to fully utilize the rich information.

Although aforementioned CDMs have made progress in cognitive diagnosis, ***we have observed an important issue that is often ignored by most existing CDMs: the homogeneous influence among students***. In educational psychology, the homogeneous influence among students is frequently explained by social learning theory [3], suggesting that a student's learning is influenced by peers who exhibit similar behaviors or characteristics. However, due to the multifaceted nature of learning and intricate non-linear interaction among students, the backbones of existing CDMs may not be flexible and complex enough, making them hard to capture the homogeneous influence among students. Consequently, this could lead to shortcoming in interpretability and efficacy of diagnosis, thereby impacting downstream tasks and the practical use of teachers and students in real-world scenarios.

In order to understand this issue intuitively and support the motivation of this paper, we conduct a preliminary study of cognitive diagnosis on the real-world dataset Math1 [26] to analyze four representative CDMs (all tuned to optimum and data split identically). The students' proficiency levels obtained by them are visualized by t-SNE [37] respectively. As shown in Figure 2, although their response logs are similar and correctness rate is the same in (a), their proficiency levels inferred by these methods separate significantly, implying an inability to cluster similar students and suboptimal performance in cognitive diagnosis. The quantitative experimental results based on our proposed metrics in Section 5.2 also validate the deficiencies when homogeneous influence is not considered.

Since the backbones of existing CDMs, such as the multi-layer perceptron or pair-wise attention networks, may be inadequate in effectively capturing the homogeneous influence among students, a crucial question is then raised: Is there any more flexible and effective way to model the high-order and non-linear homogeneous influence among students? Fortunately, the answer is YES: A hypergraph could offer an effective solution since it possesses significant advantages in modeling high-order correlations [12] and is more complex than pair-wise relations adopted by RCD, and has found extensive applications in fields such as computer vision [19] and recommendation systems [43]. However, directly applying the hypergraph to cognitive diagnosis is not feasible due to the following challenges: **i) Distorted homogeneous influence.** The use of heterogeneous hypergraphs such as relation maps utilized by existing CDMs may lead to homogeneous influence among students being distorted by interference from exercises and knowledge concepts, finally reducing the interpretability and efficacy of diagnosis. **ii) Sparsity in response logs.** Some work in other domains has provided construction methods for hypergraphs [12]. However, because students may not complete every exercise, there is a high degree of sparsity in response logs of some online datasets. Constructing a graph directly based on it could lead to a graph with very few and sparse usable edges, hindering the effective utilization of homogeneous influence. **iii) Over-smoothing problem.** The

GNNs-based CDMs with embedding propagation layers may difficultly provide distinguishable students' proficiency levels, which adversely affects the representation quality of diagnostic outcomes.

To this end, this paper proposes a hypergraph cognitive diagnosis model (HyperCDM) to address these challenges and effectively capture the homogeneous influence. In particular, to avoid distortion, HyperCDM employs a divide-and-conquer strategy to learn student, exercise and knowledge concept representations in their own hypergraphs respectively, and interconnects them via a feature-based interaction function. To construct hypergraphs based on sparse response logs, the auto-encoder is utilized to preprocess response logs and K-means is applied to cluster students. To mitigate over-smoothing, momentum hypergraph convolution networks (MHGCN) are designed to partially keep previous representations during the message propagation on the networks. In experimental analysis, we also propose homogeneity index (HI) and consistency index (CI) as metrics of the capability of capturing homogeneous influence. Extensive experiments on four real-world datasets show that HyperCDM achieves state-of-the-art performance in interpretability and capturing the homogeneous influence, and competitively strong performance in generalization. The ablation study verifies the efficacy of each part in HyperCDM, hyperparameter analysis provides some insights, and the case study explicitly showcases the homogeneous influence captured by HyperCDM.

In the subsequent sections, we respectively recap the related work, introduce the preliminaries, present the proposed HyperCDM, analyze the experimental results and finally conclude the paper.

2 Related Work

Cognitive Diagnosis. Cognitive diagnosis is a vital field in educational psychology. Unlike knowledge tracing [1, 6, 35], it focuses on the diagnosis of static cognitive states. In recent decades, considerable efforts have been invested in the development of cognitive diagnosis models (CDMs), such as item response theory (IRT) [30], multidimensional IRT (MIRT) [36], deterministic inputs, noisy and gate model (DINA) [7], neural cognitive diagnosis model (NCDM) [39], knowledge-association neural cognitive diagnosis model (KaNCD) [40], Q-augmented causal cognitive diagnosis model (QCCDM) [27], symbolic cognitive diagnosis model (SCDM) [34] and foundation model enhanced cognitive diagnosis model (FineCD) [23]. These methods model the student-exercise interaction as the interaction function. Specifically, IRT, MIRT and DINA utilize interaction functions annotated by experts. NCDM, KaNCD and QCCDM leverage neural networks to capture intricate non-linear interactions, SCDM utilizes symbolic regression to explicitly capture non-linear interaction, and FineCD utilizes LLMs. Recently, some research has focused on graph-based CDMs since graphs can model more complex relationships than previous ones. For example, relation map driven cognitive diagnosis model (RCD) [11] and supervised graph learning cognitive diagnosis model (SCD) [41] establishes relation graphs to integrate structural information, and inductive cognitive diagnosis model (ICDM) [28] designs student-centered graph to aggregate neighbors' embeddings for fast diagnosis. As aforementioned, however, we observe that these models may not be flexible enough to effectively capture the homogeneous influence among students, sometimes leading to deficiencies in diagnosis.

Graph Representation Learning. In the past decades, graphs have found widespread applications in data mining and machine learning, whose representation learning, especially based on deep learning techniques, has become a popular topic. The message-passing graph neural networks (GNN) is the primary framework of GNN, including graph convolutional networks (GCN) [21], graph attention networks (GAT) [38], GraphSAGE [14] and others. Recently, some work has focused on hypergraphs, where the traditional pairwise edges are expanded into hyperedges capable of connecting an arbitrary number of nodes. To name a few, hypergraph convolution networks (HGCN) [10] and hypergraph transformer [4, 8]. As aforementioned, directly adopting existing architecture is not feasible due to the challenges of distorted homogeneous influence, sparsity of response logs and over-smoothing. Note that while transformer-based GNNs outperform traditional methods in alleviating over-smoothing and over-squashing [24] issues, they require a larger volume of training data to achieve the optimal performance [45] and lack of interpretability, which is not suitable for this task.

3 Preliminaries

3.1 Cognitive Diagnosis

Let $S = \{s_1, \ldots, s_{|S|}\}$, $E = \{e_1, \ldots, e_{|E|}\}$, and $K = \{k_1, \ldots, k_{|K|}\}$ respectively denote the sets of students, exercises and knowledge concepts, where $|\cdot|$ calculates the size of set. Each student is required to complete some exercises, and the corresponding response logs are denoted as a set of triplets $R = \{s_i, e_j, r_{ij} | s_i \in S, e_j \in E, r_{ij} \in \{0, 1\}\}$, where r_{ij} represents the score (i.e., $r_{ij} = 1$ means right while $r_{ij} = 0$ wrong) obtained by student s_i on exercise e_j. Besides, the Q-matrix, typically annotated by experts, is denoted as $\boldsymbol{Q} = \{q_{i,j}\}_{|E|\times|K|}$, which includes the relationship between exercises and knowledge concepts. Here, $q_{i,j} = 1$ if exercise e_i is associated with knowledge concept k_j, and $q_{i,j} = 0$ otherwise.

Task Definition. Given the observed triplet logs of students R and the labeled Q-matrix $\boldsymbol{Q}$, the goal of the task is to infer the students' proficiency levels on knowledge concepts.

To ensure the interpretability, recent work also incorporates the monotonicity assumption [32] and it is defined as Assumption 1. We assess diagnostic outcomes based on it in Section 5.2.

Assumption 1 (Monotonicity Assumption). *The probability of providing a correct response to the exercise consistently rises with the student's proficiency level on relevant knowledge concepts.*

Besides, as aforementioned, the homogeneous influence among students is expected to play a crucial role in cognitive diagnosis [3], but little attention has been devoted to this issue in previous studies. We define this assumption as Assumption 2 based on social learning theory [3] and assess diagnostic outcomes based on it in Section 5.2. The similarity can be measured via distances like cosine.

Assumption 2 (Homogeneity Assumption). *If students have similar correctness rates and perform similarly in answering exercises, their proficiency levels on relevant knowledge concepts are similar.*

3.2 Hypergraph

An edge of the simple graph only connects two vertices, but a hyperedge of the hypergraph establishes connections among two or more vertices. A hypergraph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, including

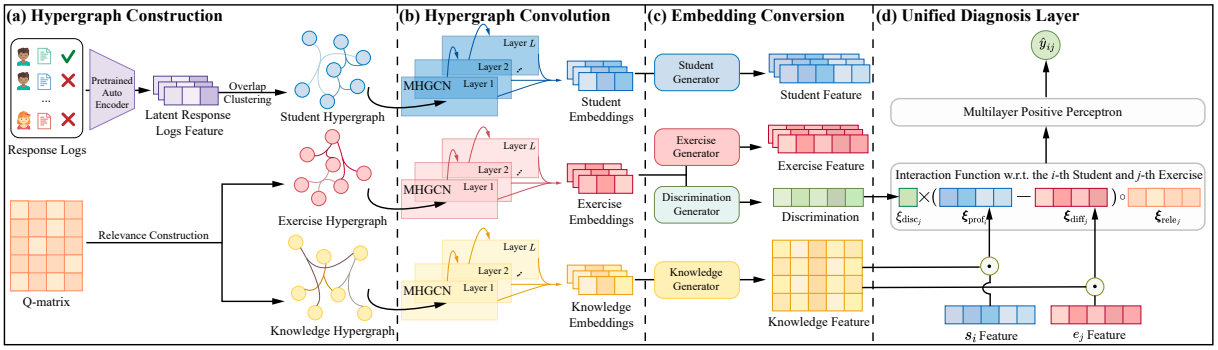


Figure 3: An overview of hypergraph cognitive diagnosis model (HyperCDM).

a vertex set $\mathcal{V}$ and a hyperedge set $\mathcal{E}$. The hypergraph $\mathcal{G}$ can be denoted by $|\mathcal{V}| \times |\mathcal{E}|$ incidence matrix $\boldsymbol{H}$, with entries defined as

$$h(v, \varepsilon) = \begin{cases} 1, & \text{if } v \in \varepsilon; \\ 0, & \text{if } v \notin \varepsilon. \end{cases} \tag{1}$$

The degree of vertex $v \in \mathcal{V}$ is defined as $\sum_{\varepsilon \in \mathcal{E}} h(v, \varepsilon)$, and the degree of an edge $\varepsilon \in \mathcal{E}$ is $\sum_{v \in \mathcal{V}} h(v, \varepsilon)$. Besides, $\boldsymbol{D}_\varepsilon$ and $\boldsymbol{D}_v$ denote the diagonal matrices of the edge degrees and the vertex degrees, respectively. Compared with the simple graph, a hypergraph inherently exhibits the capability to represent complex connections, enabling the capture of complex homogeneous influence.

4 Hypergraph Cognitive Diagnosis

An overview of the hypergraph cognitive diagnosis model (HyperCDM) is shown in Figure 3. Sequentially, student, exercise and knowledge hypergraphs are constructed in different strategies, and their representations are learned by the momentum hypergraph convolution networks (MHGCN). These embeddings are converted to diagnostic features, and these features are unified via an interaction function. HyperCDM infers students' proficiency levels by predicting the response of students on some exercises.

4.1 Hypergraph Construction

Firstly, we consider the construction of the exercise hypergraph and knowledge hypergraph. Unlike students' response logs, exercises and knowledge concepts inherently exhibit a low-noise relationship matrix, i.e., Q-matrix $\boldsymbol{Q}$ (similar to previous work, we affirm the expert-annotated $\boldsymbol{Q}$ is highly credible and needs no adjustment). And the construction is shown as Eq. (2).

$$\boldsymbol{H}_E = \boldsymbol{Q}, \quad \boldsymbol{H}_K = \boldsymbol{Q}^\top, \tag{2}$$

where the $\boldsymbol{H}_E$ is the incidence matrix of the exercise hypergraph and $\boldsymbol{H}_K$ is that of the knowledge hypergraph, as shown in Figure 3 (a). Next, we tackle constructing the student hypergraph. As mentioned before, students may not complete all exercises. Directly constructing a graph from their response logs could create sparse edges, leading to inefficient use of homogeneous influence. Therefore, inspired by [44], the construction method is depicted in Figure 4. We convert triplet response logs into a more manageable response matrix $\boldsymbol{R}$ as Eq. (3), where the $\hat{\boldsymbol{r}}_i$ is the i-th column of response matrix $\boldsymbol{R}$ (if not specified, all mentioned vectors are column vectors), and also the response logs of student s_i.

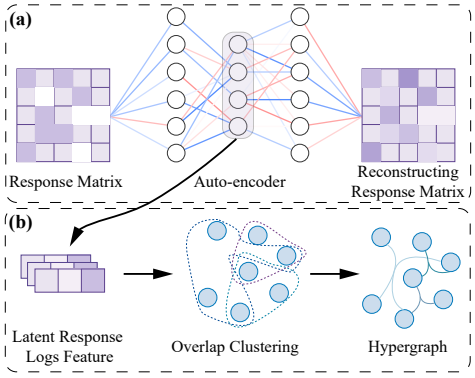


Figure 4: Details of the student hypergraph construction. (a) Utilizing auto-encoder to process response matrix. (b) After gaining the latent response logs feature, the overlap clustering is applied to construct the hypergraph.

$$\boldsymbol{R} = \{\hat{r}_{ji}\}^{|E| \times |S|}, \ \hat{r}_{ji} = \begin{cases} 1, & \text{if } (s_i, e_j, r_{ij}) \in R, r_{ji} = 1 \\ -1, & \text{if } (s_i, e_j, r_{ij}) \in R, r_{ji} = 0 \\ 0, & \text{otherwise} \end{cases}. \tag{3}$$

As shown in Figure 4 (a), we obtain clustering-friendly latent representations via auto-encoder, i.e.,

$$\tilde{\boldsymbol{r}}_i = \mathcal{A}(\hat{\boldsymbol{r}}_i; \Theta_{\mathcal{A}}), \quad \mathcal{A}(\cdot; \Theta_{\mathcal{A}}) : \mathbb{R}^{|E|} \mapsto \mathbb{R}^{|\tilde{E}|}, \tag{4}$$

where $\tilde{\boldsymbol{r}}_i$ is the latent response logs feature of student s_i, $\mathcal{A}$ is the non-linear encoder mapping, $|\tilde{E}|$ is the dimension of latent feature, and $\Theta_{\mathcal{A}}$ is a set of parameters of encoder. To improve the performance of subsequent clustering, we pre-train the auto-encoder via reconstruction loss $\mathcal{L}_{\text{rec}}$, i.e.,

$$\mathcal{L}_{\text{rec}} = \sum_{i=1}^{|S|} ||\mathcal{A}^{-1}(\mathcal{A}(\hat{\boldsymbol{r}}_i; \Theta_{\mathcal{A}}); \Theta_{\mathcal{A}^{-1}}) - \hat{\boldsymbol{r}}_i||_2, \tag{5}$$

where $\mathcal{A}^{-1}(\cdot; \Theta_{\mathcal{A}^{-1}}) : \mathbb{R}^{|\tilde{E}|} \mapsto \mathbb{R}^{|E|}$ is the decoder, $\Theta_{\mathcal{A}^{-1}}$ is its set of parameters, and $||\cdot||_2$ is l_2 norm.

After pre-training, we obtain the hypergraph using the overlap K-means. Initially, following the classical K-means [15], the optimization objective is defined as Eq. (6), where each column of

Algorithm 1 Alternating Clustering

Hyperparameter: Maximum number of epoch T.

1: **for** $t = 1, \ldots, T$ **do**
2: Update network parameters $\mathcal{A}, \mathcal{A}^{-1}$ via $\nabla_{(\mathcal{A},\mathcal{A}^{-1})}\mathcal{L}_{\text{cons}}$ and stochastic gradient descent optimizer (e.g., Adam).
3: **for** $i = 1, \ldots, |S|$ **do**
4: Update cluster assignment as follows:

$$o_{j,i} \leftarrow \begin{cases} 1, & \text{if } j = \underset{k=1,\cdots,\mathcal{K}}{\arg\min} ||\mathcal{A}(\hat{\boldsymbol{r}}_i; \Theta_{\mathcal{A}}) - \boldsymbol{c}_k||_2 ; \\ 0, & \text{otherwise} . \end{cases}$$

5: Update all centroids as follows:

$$\boldsymbol{c}_k \leftarrow \boldsymbol{c}_k - \delta_k^i (\boldsymbol{c}_k - \mathcal{A}(\hat{\boldsymbol{r}}_i; \Theta_{\mathcal{A}})) o_{k,i} .$$

6: **end for**
7: **end for**

centroid matrix $\boldsymbol{C} = [\boldsymbol{c}_1, \cdots, \boldsymbol{c}_{\mathcal{K}}] \in \mathbb{R}^{|\tilde{E}| \times \mathcal{K}}$ is the feature representation of $\mathcal{K}$ centroids, and $\boldsymbol{o_i}$ is the one-hot vector to signify which clustering the i-th student's latent feature belongs to.

$$\begin{aligned} &\min_{\boldsymbol{C}, \{\boldsymbol{o_i}\}} \mathcal{J}_{\text{clt}} = \sum_{i=1}^{|S|} ||\mathcal{A}(\hat{\boldsymbol{r}}_i; \Theta_{\mathcal{A}}) - \boldsymbol{C} \cdot \boldsymbol{o_i}||_2 \\ &\text{s.t. } \boldsymbol{o_i} \in \{0,1\}^{|\tilde{E}|}, \mathbf{1}^\top \boldsymbol{o_i} = 1, \forall i . \end{aligned} \tag{6}$$

And we give the loss function of hypergraph construction,

$$\mathcal{L}_{\text{cons}} = \mathcal{L}_{\text{rec}} + \mathcal{J}_{\text{clt}} . \tag{7}$$

However, optimizing Eq. (7) is challenging since both the loss function and the constraints are non-convex. Besides, scalability issues should be taken into consideration. Therefore, after the pre-training process, the clustering algorithm is shown in Algorithm 1. Line 2 shows further optimization based on the pre-trained auto-encoder using a stochastic gradient descent optimizer (e.g., Adam [20]). Line 4 entails reassigning the cluster centroids for each student based on the Euclidean distance. In line 5, an adaptive approach [33] is employed to update the cluster centers, where δ_k^i is the inverse of the frequency that the algorithm assigns students to cluster k before processing the incoming student s_i.

Finally, we construct a student hypergraph based on the Euclidean distance between each student's latent feature and centroids. Initially, we compute the first quartile $\gamma_{0.25_k}$ of the distances from students to each clustering k, and fill incidence matrix of student hypergraph $\boldsymbol{H}_S \in \mathbb{R}^{|S| \times \mathcal{K}}$ as follows in Figure 4 (b):

$$\boldsymbol{H}_S = \{h_{S_{i,k}}\},\ h_{S_{i,k}} = \begin{cases} 1, & \text{if } ||\mathcal{A}(\hat{\boldsymbol{r}}_i; \Theta_{\mathcal{A}}) - \boldsymbol{c}_k||_2 \le \gamma_{0.25_k} ; \\ 0, & \text{otherwise} . \end{cases} \tag{8}$$

4.2 Hypergraph Convolution

Upon constructing hypergraphs, the next step is to learn the graph representation. Given a hypergraph derived from datasets, the incident matrix $\boldsymbol{H}$ and vertex embeddings in l-th layer $\boldsymbol{X}^{(l)}$ are inputted into hypergraph convolution networks [10, 12] defined as:

$$\boldsymbol{X}^{(l+1)} = \sigma\left(\boldsymbol{D}_v^{-\frac{1}{2}} \boldsymbol{H} \boldsymbol{D}_\varepsilon^{-1} \boldsymbol{H}^\top \boldsymbol{D}_v^{-\frac{1}{2}} \boldsymbol{X}^{(l)} \boldsymbol{\Theta}^{(l)}\right), \tag{9}$$

where $\boldsymbol{\Theta}^{(l)}$ is a trainable parameter matrix at the l-th layer, and $\sigma(\cdot)$ denotes an non-linear activation function (e.g., Tanh). However, He et al. [16] utter that since each node has no concrete semantics features, applying multi-layer non-linear feature transformation (e.g., activation function) will increase training difficulty, impede speed and impair performance. Besides, the over-smoothing problem is prevalent in message-passing GNNs. To expedite training, enhance performance, and alleviate over-smoothing, we propose the momentum hypergraph convolution networks (MHGCN) and sum up the embeddings of each layer to combine them:

$$\begin{aligned} \boldsymbol{X}^{(l+1)} &= \boldsymbol{D}_v^{-\frac{1}{2}} \boldsymbol{H} \boldsymbol{D}_\varepsilon^{-1} \boldsymbol{H}^\top \boldsymbol{D}_v^{-\frac{1}{2}} \boldsymbol{X}^{(l)} + \eta \cdot \boldsymbol{X}^{(l)} , \\ \boldsymbol{X} &= \frac{1}{1+L} \sum_{l=0}^{L} \boldsymbol{X}^{(l)} , \end{aligned} \tag{10}$$

where η is momentum parameter and L is the number of layers of MHGCN. The omission of trainable parameter $\Theta^{(l)}$ and non-linear transformations σ accelerates training speed and improves performance. To mitigate over-smoothing, we integrate embeddings from previous layers partially during the update. This structural modification's efficacy is confirmed through ablation study in Section 5.2.

Based on MHGCN, as shown in Figure 3 (b), for each hypergraph, the convolution is defined as Eq. (11).

$$\begin{cases} \boldsymbol{X}_S^{(l+1)} = \boldsymbol{D}_{S_v}^{-\frac{1}{2}} \boldsymbol{H}_S \boldsymbol{D}_{S_\varepsilon}^{-1} \boldsymbol{H}_S^\top \boldsymbol{D}_{S_v}^{-\frac{1}{2}} \boldsymbol{X}_S^{(l)} + \eta \cdot \boldsymbol{X}_S^{(l)} \\ \boldsymbol{X}_E^{(l+1)} = \boldsymbol{D}_{E_v}^{-\frac{1}{2}} \boldsymbol{H}_E \boldsymbol{D}_{E_\varepsilon}^{-1} \boldsymbol{H}_E^\top \boldsymbol{D}_{E_v}^{-\frac{1}{2}} \boldsymbol{X}_E^{(l)} + \eta \cdot \boldsymbol{X}_E^{(l)} \\ \boldsymbol{X}_K^{(l+1)} = \boldsymbol{D}_{K_v}^{-\frac{1}{2}} \boldsymbol{H}_K \boldsymbol{D}_{K_\varepsilon}^{-1} \boldsymbol{H}_K^\top \boldsymbol{D}_{K_v}^{-\frac{1}{2}} \boldsymbol{X}_K^{(l)} + \eta \cdot \boldsymbol{X}_K^{(l)} \end{cases} , \tag{11}$$

where subscripts S, E, K mean components in the student, exercise and knowledge hypergraph respectively, and $\boldsymbol{X}_S^{(l)}, \boldsymbol{X}_S^{(l+1)} \in \mathbb{R}^{|S| \times d_{\text{emb}}}, \boldsymbol{X}_E^{(l)}, \boldsymbol{X}_E^{(l+1)} \in \mathbb{R}^{|E| \times d_{\text{emb}}}, \boldsymbol{X}_K^{(l)}, \boldsymbol{X}_K^{(l+1)} \in \mathbb{R}^{|K| \times d_{\text{emb}}}$. The layer combination is defined as Eq. (12).

$$\begin{cases} \boldsymbol{X}_S = \frac{1}{1+L} \sum_{l=0}^{L} \boldsymbol{X}_S^{(l)} \\ \boldsymbol{X}_E = \frac{1}{1+L} \sum_{l=0}^{L} \boldsymbol{X}_E^{(l)} \\ \boldsymbol{X}_K = \frac{1}{1+L} \sum_{l=0}^{L} \boldsymbol{X}_K^{(l)} \end{cases} , \tag{12}$$

where $\boldsymbol{X}_S \in \mathbb{R}^{|S| \times d_{\text{emb}}}$, $\boldsymbol{X}_E \in \mathbb{R}^{|E| \times d_{\text{emb}}}$, and $\boldsymbol{X}_K \in \mathbb{R}^{|K| \times d_{\text{emb}}}$.

4.3 Embedding Conversion

Some work [7, 39] usually sets d_{emb} as the number of knowledge concepts $|K|$, and inputs them into the interaction functions. However, forcing the dimension to $|K|$ may reduce the performance. This problem is commonly faced in some graph downstream tasks like recommendation systems [16]. To address it, we first embed students, exercises and knowledge concepts into a low-dimensional latent space. After convolution, we apply a linear transformation to increase dimension of embeddings, which is expressed as follows:

$$\begin{cases} \tilde{\boldsymbol{X}}_S = \sigma(\text{Linear}_S(\boldsymbol{X}_S; \Theta_S)) \\ \tilde{\boldsymbol{X}}_E = \sigma(\text{Linear}_E(\boldsymbol{X}_E; \Theta_E)) \\ \tilde{\boldsymbol{X}}_K = \sigma(\text{Linear}_K(\boldsymbol{X}_S; \Theta_K)) \end{cases} , \tag{13}$$

where $\Theta_S, \Theta_E, \Theta_K$ are parameters in each linear transformation, $\text{Linear}_S(\cdot; \Theta_S)$ is the mapping $\mathbb{R}^{|S| \times d_{\text{emb}}} \mapsto \mathbb{R}^{|S| \times d_{\text{feat}}}$, $\text{Linear}_E(\cdot; \Theta_E)$ is $\mathbb{R}^{|E| \times d_{\text{emb}}} \mapsto \mathbb{R}^{|E| \times d_{\text{feat}}}$, $\text{Linear}_K(\cdot; \Theta_K)$ is $\mathbb{R}^{|K| \times d_{\text{emb}}} \mapsto \mathbb{R}^{|K| \times d_{\text{feat}}}$,

and σ is the activation function (here is LeakyReLU). It is noteworthy that d_{feat} may not always equal $|K|$, and typically it satisfies $d_{\text{feat}} \geq |K| > d_{\text{emb}}$. This is because more parameters can result in a more effective fit [17], which is also validated in hyperparameter analysis of Section 5.2. At the same time, discrimination holds equal importance as another aspect of exercise features (i.e., difficulty) [25, 29]. We obtain the discrimination $\boldsymbol{\xi}_{\text{disc}}$ based on the exercise embeddings, and employ a method similar to Eq. (13), as shown below. The $\text{Linear}_{\text{disc}}(\cdot;\Theta_{\text{disc}})$ is the mapping $\mathbb{R}^{|E|\times d_{\text{emb}}} \mapsto \mathbb{R}^{|E|\times 1}$, and Θ_{disc} is a set of parameters in the linear transformation.

$$\boldsymbol{\xi}_{\text{disc}} = \sigma(\text{Linear}_{\text{disc}}(\boldsymbol{X}_E;\Theta_{\text{disc}}))\,. \tag{14}$$

4.4 Unified Diagnosis Layer

According to the principle of divide-and-conquer, we first model students, exercises and knowledge concepts respectively using homogeneous hypergraphs, aiming to prevent interference from different types of nodes in capturing homogeneous influence; then, we employ a unified interaction function to combine these three components to simultaneously train these features. As illustrated in Figure 3 (d), for student s_i and exercise e_j, the corresponding proficiency level, exercise difficulty, discrimination and relevance features are obtained as follows:

$$\begin{aligned}
&\boldsymbol{\xi}_{\text{prof}_i} = \tilde{\boldsymbol{X}}_K \tilde{\boldsymbol{X}}_S^\top \boldsymbol{u}_i\,,\\
&\text{s.t. } \boldsymbol{u}_i \in \{0,1\}^{|S|}, \mathbf{1}^\top \boldsymbol{u}_i = 1, u_{i,i} = 1\,;\\
&\boldsymbol{\xi}_{\text{diff}_j} = \tilde{\boldsymbol{X}}_K \tilde{\boldsymbol{X}}_E^\top \boldsymbol{z}_j,\ \xi_{\text{disc}_j} = \boldsymbol{\xi}_{\text{disc}}^\top \boldsymbol{z}_j,\ \boldsymbol{\xi}_{\text{rele}_j} = \boldsymbol{Q}^\top \boldsymbol{z}_j\,,\\
&\text{s.t. } \boldsymbol{z}_j \in \{0,1\}^{|E|}, \mathbf{1}^\top \boldsymbol{z}_j = 1, z_{j,j} = 1\,,
\end{aligned} \tag{15}$$

where $\boldsymbol{\xi}_{\text{prof}_i}, \boldsymbol{\xi}_{\text{diff}_j}, \boldsymbol{\xi}_{\text{rele}_j} \in \mathbb{R}^{|K|}$, $\xi_{\text{disc}_j} \in \mathbb{R}$, and $\boldsymbol{u}_i, \boldsymbol{z}_j$ are one-hot vectors. In Eq. (15), it is noteworthy that proficiency level and difficulty are not directly obtained from feature matrices $\tilde{\boldsymbol{X}}_S, \tilde{\boldsymbol{X}}_E$. Instead, they are derived by multiplying $\tilde{\boldsymbol{X}}_S, \tilde{\boldsymbol{X}}_E$ with the knowledge feature matrix $\tilde{\boldsymbol{X}}_K$ and extracting via one-hot vectors. This is because: **i)** Performing a linear transformation via $\tilde{\boldsymbol{X}}_K$ to ensure their dimensions correspond to $|K|$, thereby ensuring interpretability (distinguishing it from previous work like MIRT [36]). **ii)** In the knowledge hypergraph, the implicit associations between knowledge concepts can be learned and kept in $\tilde{\boldsymbol{X}}_K$. By incorporating these associations into $\tilde{\boldsymbol{X}}_S, \tilde{\boldsymbol{X}}_E$, it is possible to predict proficiency level even when a student has minimal exposure or zero-shot on a particular knowledge concept (especially in online datasets where students are not required to interact with all concepts) and enhances interpretability and capturing the homogeneous influence.

After obtaining these diagnostic factors, we input them into the unified diagnostic layers defined as:

$$y_{ij} = \text{Sigmoid}(\text{MLP}(\xi_{\text{disc}_j} \times (\boldsymbol{\xi}_{\text{prof}_i} - \boldsymbol{\xi}_{\text{diff}_j}) \circ \boldsymbol{\xi}_{\text{rele}_j}; \Theta_{\text{inter}}))\,, \tag{16}$$

where y_{ij} is the prediction of the i-th student's response on the j-th exercise, $\text{Sigmoid}(x)$ is the activation function $\frac{1}{1+e^{-x}}$, $\text{MLP}(\cdot): \mathbb{R}^{|K|} \mapsto \mathbb{R}$ is the multi-layer perceptron, Θ_{inter} are parameters whose weights are non-negative to adhere to Assumption 1, $\times$ is the scalar multiplication, and $\circ$ is the Hadamard product.

To train HyperCDM via prediction, we adopt cross entropy loss $\mathcal{L}_{\text{bce}}$ between output y_{ij} and the true label r_{ij}, and incorporate the l_2 regularization term to avoid over-fitting. The entire loss $\mathcal{L}$ is defined as Eq. (17), where Θ including all parameters.

Table 1: Time complexity and space complexity analysis of each component in HyperCDM. The symbol "—" means that this entry is not applicable to the component.

Component		Time Complexity	Space Complexity
Hypergraph Construction	Auto-encoder	$O(T\|E\|\|S\|(\|\Theta_{\mathcal{A}}\| + \|\Theta_{\mathcal{A}^{-1}}\|))$	$O(\|\Theta_{\mathcal{A}}\| + \|\Theta_{\mathcal{A}^{-1}}\|)$
	Overlap K-means	$O(T\mathcal{K}\|S\|\|\tilde{E}\|)$	$O(\|S\|\|\tilde{E}\|)$
Representation Learning	Incidence Matrix	–	$O(2\|E\|\|K\| + \|S\|\mathcal{K})$
	Convolution	$O(L(\text{trace}(\boldsymbol{D}_v)))$	$O(d_{\text{emb}}(\|S\| + \|E\| + \|K\|))$
	Conversion	$O(d_{\text{feat}}\|S\|\|\Theta_s\|)$	$O(\|\Theta_S\| + \|\Theta_E\| + \|\Theta_K\| + \|\Theta_{\text{disc}}\|)$
	BCE Loss	$O(2\|R\|)$	$O(2\|R\|)$

$$\begin{aligned}
&\mathcal{L}_{\text{bce}} = -\sum_{r_{ij}\in R} (r_{ij}\log y_{ij} + (1-r_{ij})\log(1-y_{ij}))\,,\\
&\mathcal{L} = \mathcal{L}_{\text{bce}} + \lambda||\Theta||_2\,.
\end{aligned} \tag{17}$$

4.5 Discussion

We would like to emphasize some points related to HyperCDM.

Complexity. We consider both time complexity and space complexity from two aspects: graph construction and graph learning. Detailed analysis is shown in Table 1, where the number of students $|S|$ is significantly larger than $|E|$, $|K|$, and we have omitted the terms in the time complexity that only involve $|E|$ and $|K|$.

Flexibility. In Figure 4, if necessary, we can also input additional side-information to assist in the hypergraph construction, such as students' background information or the problem context. And in Figure 3 (d), other types of interaction functions (e.g., MIRT and DINA) are also applicable, and adjustments to diagnostic factors can be made based on the specific requirements.

Training. The hypergraph construction, in relation to subsequent graph representation learning, serves as a "pre-training" phase. Specifically, once hypergraphs are constructed, the structure remains unchanged thereafter, which ensures the hypergraph construction does not become a bottleneck of the efficiency of subsequent representation learning. In essence, hypergraph construction and graph learning are two mutually independent processes.

5 Experiments

This section conducts experiments on real-world datasets to answer the following research questions.[1]

- **Q1:** Does the proposed HyperCDM successfully capture homogeneous influence among students?
- **Q2:** How does HyperCDM perform in interpretability?
- **Q3:** How does HyperCDM perform in generalization?
- **Q4:** Does each component contribute to the performance?
- **Q5:** How do hyperparameters influence HyperCDM?
- **Q6:** How efficient is HyperCDM compared to existing methods?
- **Q7:** Why and how does the diagnostic outcome of HyperCDM work in real-world educational scenarios?

5.1 Experiment Setup

We explain the datasets, comparison methods, metrics and settings.

[1]The source code are available at https://github.com/shinkungoo/HyperCDM.

Table 2: Details of the real-world datasets for experiments.

Dataset	NeurIPS20	EdNet-1	Math1	Math2
#Student	3000	1827	4209	3911
#Exercise	6000	11996	15	16
#Knowledge Concepts	268	189	11	16
#Response Logs	215323	556770	63135	62576
Density	0.012	0.025	1.000	1.000

Dataset Description. The experiments are conducted on four real-world datasets from both online (sparse) scenarios (NeurIPS20 [42], EdNet-1 [5]) and offline (dense) scenarios (Math1, Math2 [26]). These datasets cover common learning scenarios and are representative, which can show our method's versatility in various educational scenarios. The density is computed as $|R|/(|S| \times |E|)$. A density value nearing 1 indicates that students have completed nearly all exercises in the datasets, which is often seen in offline datasets. Conversely, a density value nearing 0 suggests that students may only complete a subset of the exercises in the datasets, which is often seen in online datasets. Details of these datasets are shown in Table 2, and their sources are explained in Appendix A.1.

Baselines and State-of-the-Art Methods. In recent decades, various CDMs have been developing, some of which are representative methods and selected for comparison: MIRT [36], DINA [7], NCDM [39], KaNCD [40], KSCD [31] and RCD [11]. Since graph-based CDMs are less explored, we choose the strong model RCD as a representative graph-based model for comparison. Note that these methods are all open source and adjusted to optimum according to their recommended settings in the paper. Details of these methods and the source are shown in Appendix A.2.

Generalization Metrics. Assessing the performance of CDM proves challenging due to the inherent difficulty in accurately observing students' proficiency levels. To address this challenge, a widely accepted strategy is to evaluate them by predicting students' test scores. Similar to previous methods [11, 39], we evaluate how close the model predicts whether a student solves an exercise to the ground truth in the test set with common classification metrics, i.e., accuracy (Acc.), area under curve (AUC) and F1-score (F1).

Interpretability Metric. Generalization metrics are only one facet of assessing the performance of CDMs. In previous work, degree of agreement (DOA) [39, 40] is usually applied to quantitatively assess the interpretability of diagnostic outcomes. As Assumption 1, if student s_a shows higher accuracy in responding to exercises related to k_i than student s_b, it implies the proficiency level of s_a on knowledge concept k_i ($\xi_{\text{prof}_{a,i}}$) may surpass that of student s_b ($\xi_{\text{prof}_{b,i}}$). The DOA of k_i is defined as Eq. (18).

$$DOA(i) = \frac{1}{Z_1} \sum_{a=1}^{|S|} \sum_{b=1}^{|S|} \Lambda(\xi_{\text{prof}_{a,i}}, \xi_{\text{prof}_{b,i}}) \sum_{j=1}^{|E|} q_{j,i} \frac{I(j,a,b) \cdot \Lambda(r_{aj}, r_{bj})}{I(j,a,b)}, \quad (18)$$

where $Z_1 = \sum_{a=1}^{|S|} \sum_{b=1}^{|S|} \Lambda(\xi_{\text{prof}_{a,i}}, \xi_{\text{prof}_{b,i}})$, and $\xi_{\text{prof}_{a,i}}$ is the proficiency level of student s_a on knowledge concept k_i. $\Lambda(x, y) = 1$ if $x > y$ and otherwise $\Lambda(x, y) = 0$. $q_{ji} = 1$ if exercise e_j contains knowledge concept k_i and otherwise $q_{ji} = 0$. $I(j, a, b) = 1$ if both student s_a and s_b complete exercise e_j and otherwise $I(j, a, b) = 0$. For DOA, a greater value indicates stronger interpretability and adherence to Assumption 1. To evaluate each knowledge concept, we average $DOA(i)$ across all knowledge concepts for offline datasets (i.e., DOA) and the top-10 frequent ones for online datasets (i.e., DOA@10), which is consistent with previous work [22, 27].

Homogeneity Metrics. To assess whether the model successfully captures the homogeneous influence among students, we propose two metrics: homogeneity index (HI) and consistency index (CI). As Assumption 2, if the correctness rate of students s_a and s_b are equal, it implies that their proficiency levels should be similar and comparable. Based on it, the HI is defined as Eq. (19).

$$HI = \frac{1}{Z_2} \sum_{a=1}^{|S|} \sum_{b=1}^{|S|} \left\| \xi_{\text{prof}_a} - \xi_{\text{prof}_b} \right\|_2 \times J(a, b), \quad (19)$$

where $Z_2 = \sum_{a=1}^{|S|} \sum_{b=1}^{|S|} J(a, b)$. The function $J(a, b) = 1$ if both students s_a and s_b ($a \neq b$) have an identical correctness rate and otherwise $J(a, b) = 0$. The correctness rate is computed as the ratio of correctly answered exercises to the total number of attempted exercises. Besides, Assumption 2 also considers that students with similar response logs should have similar proficiency levels. Therefore, we present the formula for calculating CI as Eq. (20).

$$CI = \frac{1}{Z_3} \sum_{a=1}^{|S|} \sum_{b=1}^{|S|} \left\| \xi_{\text{prof}_a} - \xi_{\text{prof}_b} \right\|_2 \cdot \left(\frac{\hat{\boldsymbol{r}}_a^\top \cdot \hat{\boldsymbol{r}}_b}{||\hat{\boldsymbol{r}}_a||_2 \times ||\hat{\boldsymbol{r}}_b||_2} \right), \quad (20)$$

where $Z_3 = |S|(|S| - 1)$ and $\cdot$ denotes the inner product. $\hat{\boldsymbol{r}}_a$ is the a-th column of the response matrix given in Section 4.1, and also the student s_a's response logs. For both HI and CI, a smaller value indicates stronger homogeneity and adherence to Assumption 2. To deeply understand these two metrics, please refer to Appendix A.4.

However, DOA, HI and CI are only applicable for NCDM, KaNCD, KSCD and RCD since the diagnostic outcomes of MIRT only exhibit vague correspondence between latent traits and knowledge concepts, and DINA is confined to discrete representation.

Experiment Settings. During training, we initialize parameters in all networks with Xavier normal initialization [13] and use Adam optimizer [20] with a learning rate being 0.0001. The batch size is 64 for all datasets and the l_2-regularization parameter $\alpha = 0.0005$. To evaluate performance, we perform an 80%/20% train/test split of response logs, and the hypergraph construction is only based on the train set. All experiments are repeated independently with 10 seeds. More Implementation details are shown in Appendix A.3.

5.2 Experimental Results

We conduct comprehensive experiments to analyze the experimental results and answer the aforementioned research questions.

Homogeneous Influence (Q1). The performance on homogeneity of all methods is shown in Table 3, and the variance of each metric is less than 0.05. From the table: **i)** By t-test, HyperCDM outperforms all baselines and state-of-the-art methods with significant level $\alpha = 5\%$ in terms of homogeneity metrics HI and CI, indicating that our model can effectively capture the homogeneous influence among students. **ii)** Based on the HI and CI values, we sort all comparative methods from low to high and observe that RCD performs best, while KSCD worst. These findings align with the preliminary results depicted in Figure 2, that is to say, in Figure 2 (c) (i.e., KSCD), the distance between similar students is farthest, whereas in (e) (i.e., RCD), the distance is relatively closer. **iii)** In each

Table 3: Performance of baselines, state-of-the-art methods, and HyperCDM in interpretability and capture of the homogeneous influence (i.e., homogeneity). DOA@10 and DOA are expressed in percentage while HI and CI in decimal form. "↑" means larger values are better while "↓" means smaller values are better. In each column, an entry is marked in bold if its mean value is the best. By t-test, a bold one is significantly better than others on the corresponding metrics with significant level $\alpha = 5\%$.

	NeurIPS20			EdNet-1			Math1			Math2		
Methods	DOA@10 (%)↑	HI↓	CI↓	DOA@10 (%)↑	HI↓	CI↓	DOA (%)↑	HI↓	CI↓	DOA (%)↑	HI↓	CI↓
NCDM	71.36	1.4992	1.5963	59.99	1.4564	2.2783	54.16	0.2206	0.2647	54.89	0.5299	0.5732
KaNCD	73.37	3.3682	3.4992	63.72	2.5189	2.3983	58.50	0.0710	0.1366	62.30	0.9455	0.9666
KSCD	57.43	9.9884	9.1289	57.03	8.7416	9.0598	51.69	0.8367	1.2651	48.36	1.9788	2.3911
RCD	58.95	0.1506	0.1584	55.12	0.1599	0.1775	57.93	0.0280	0.0338	61.77	0.0339	0.0431
HyperCDM	**74.29**	**0.1350**	**0.1486**	**66.02**	**0.1103**	**0.1602**	**60.42**	**0.0099**	**0.0107**	**64.34**	**0.0159**	**0.0278**

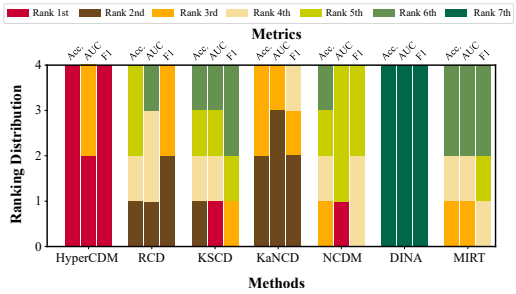


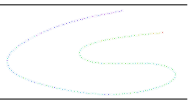
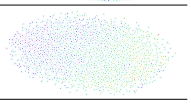
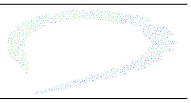
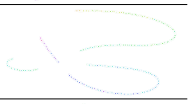
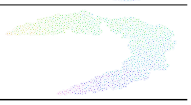
Figure 5: Performance rank on generalization. The height of rectangles of different colors signifies the number of corresponding rank of the given metric (upper x-axis) across all datasets for the given method (lower x-axis). Detailed statistics are shown in Appendix B.1 and Table 6.

dataset, graph-based methods (HyperCDM and RCD) show similar magnitudes for HI and CI values, indicating graph structures improve capturing the homogeneous influence among students. However, simple pair-wise graphs fall short in effectively capturing complex relationships, leading to HyperCDM outperforming RCD.

Interpretability (Q2) In Table 3, by t-test, HyperCDM reaches state-of-the-art performance in terms of interpretability metric DOA, and its variance is less than 0.05. The relative improvements (%) over the best baselines are 1.25%, 3.61%, 3.28% and 3.27% for NeurIPS20, EdNet-1, Math1 and Math2 respectively, and those over the RCD are 26.0%, 19.8%, 4.30% and 4.16%. It demonstrates that: **i)** HyperCDM not only effectively captures homogeneous influence but also enhances the interpretability via hypergraph structure. **ii)** Compared to the graph-based method RCD, HyperCDM significantly outperforms it, especially in online datasets NeurIPS20 and EnNet-1. This indicates the hypergraph is more flexible and effective than pair-wise graphs, and validates the effectiveness of our hypergraph construction strategy in handling sparse online interaction. Specifically, while RCD directly constructs a relation map based on sparse response logs [11], we construct hypergraphs via their latent features to avoid this problem.

Generalization (Q3). The performance of baseline methods and HyperCDM on generalization are illustrated in Figure 5, which shows the ranking distribution of a model across the four datasets for a specific metric. The HyperCDM secures all top positions in both Acc. and F1, while also achieving commendable performance

Table 4: Ablation study of HyperCDM. By t-test, a bold one is significantly better than others on the corresponding metrics with significant level $\alpha = 5\%$. To intuitively understand the function of each component, we visualize the corresponding proficiency level by t-SNE (similar to Figure 2).

Methods	Datasets	Metrics				t-SNE Figure on Math1
		Acc.(%)↑	AUC(%)↑	HI↓	DOA(%)↑	
HyperCDM w.o. Mon	NeurIPS20	71.76	77.32	0.1571	68.90	
	EdNet-1	71.70	74.41	0.1385	65.87	
	Math1	68.32	74.30	0.0108	59.97	
	Math2	69.35	77.76	0.0264	63.46	
HyperCDM w.o. Con	NeurIPS20	71.14	76.99	1.5126	46.90	
	EdNet-1	71.45	74.40	1.4977	45.72	
	Math1	67.60	73.84	0.2718	49.42	
	Math2	69.91	77.21	0.5764	49.95	
HyperCDM w.o. G	NeurIPS20	71.66	76.72	0.9649	72.80	
	EdNet-1	71.08	73.13	0.8756	62.84	
	Math1	68.04	74.21	0.1406	51.81	
	Math2	69.40	77.05	0.3675	52.53	
HyperCDM w.o. L	NeurIPS20	71.48	76.84	0.3547	73.05	
	EdNet-1	71.50	74.03	0.4152	63.31	
	Math1	67.76	74.01	0.0514	59.73	
	Math2	69.99	77.67	0.0739	63.56	
HyperCDM	NeurIPS20	**71.88**	**77.48**	**0.1350**	**74.29**	
	EdNet-1	**71.98**	**74.45**	**0.1103**	**66.02**	
	Math1	**68.47**	**74.76**	**0.0099**	**60.42**	
	Math2	**70.35**	**78.27**	**0.0159**	**64.34**	

on AUC. This suggests that HyperCDM not only excels in capturing homogeneous influence and interpretability but also shows strong competitiveness in terms of generalization. Detailed statistics are shown in Appendix B.1 and Table 6.

Ablation Study (Q4). We conduct an ablation study to evaluate several variants including: (a) **HyperCDM w.o. Mon** that does not use momentum strategy in MHGCN (i.e., let $\eta = 0$); (b) **HyperCDM w.o. Con** that does not convert low-dimensional embeddings to high-dimensional features; (c) **HyperCDM w.o. G** that replaces constructed hypergraphs with random graphs; (d) **HyperCDM w.o. L** that directly construct hypergraphs based on sparse response logs. As shown in Table 4, HyperCDM outperforms all variants. Besides, we observe that utilizing latent features for hypergraph construction and momentum in MHGCN enhances the distinctiveness and establishes a more evident sequential relationship among students in t-SNE figure. We also find that converting embeddings into features makes the arrangement of points more compact and orderly,

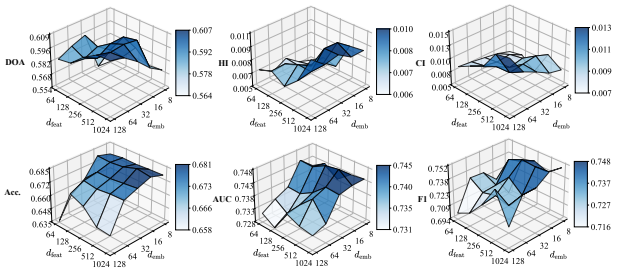


Figure 6: Hyperparameter analysis of d_{emb} and d_{feat}.

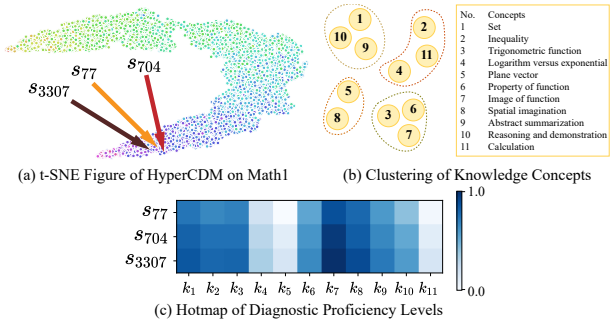
Table 5: Comparison of running times between baselines and HyperCDM, measured in seconds. The measurement interval spans from the graph construction (if any) to achieving optimum performance on the validation set.

Methods	NeurIPS20	EdNet-1	Math1	Math2
MIRT	327	704	295	249
DINA	344	512	366	305
NCDM	683	482	136	117
KaNCD	651	467	121	114
KSCD	1678	1403	322	264
RCD	9144	38524	1531	1318
HyperCDM	3067	3744	511	535

aligning with Assumption 2. Compared with random graphs, students' proficiency levels obtained by HyperCDM better match their correctness rate. More details can be found in Appendix B.2.

Hyperparameter Analysis (Q5). We conduct a hyperparameter experiment and the results of $d_{\text{emb}}, d_{\text{feat}}$ on Math1 are shown in Figure 6. Without loss of generality, we find that: **i)** Optimal interpretability is often achieved with high dimension of both feature space and embeddings because more dimension enriches the representation. **ii)** Optimal homogeneity is usually achieved with low dimension of both feature space and embeddings. **iii)** Optimal generalization is typically achieved with a high-dimensional feature space and a low-dimensional embedding. Thus, they mutually constrain each other. To optimize performance, we can choose high-dimensional features, and improve the homogeneity and generalization by choosing low-dimensional embeddings (as we mentioned in Section 4.3). Comprehensive analysis can be found in Appendix B.3.

Scalability Analysis (Q6). Table 5 presents the runtime of baseline methods and proposed HyperCDM on the same hardware devices (details of the devices are shown in Section A.3). Graphs offer significant benefits in modeling complex relationships but at the cost of increased time requirements. Compared with traditional and neural network-based cognitive diagnosis models, graph-based models are slower. To address this, our model uses a lightweight network to reduce time disadvantages. Specifically, our model operates significantly faster compared to RCD. And even on large-scale datasets (i.e., NeurIPS20 and EdNet-1), HyperCDM can still complete diagnosis within the timeframe that static cognitive diagnosis can tolerate. Therefore, HyperCDM can be applied to large-scale datasets under the current task setting.



(a) t-SNE Figure of HyperCDM on Math1 (b) Clustering of Knowledge Concepts

(c) Hotmap of Diagnostic Proficiency Levels

Figure 7: Case study: t-SNE figure of HyperCDM, knowledge concepts clustering and proficiency levels of $s_{77}, s_{704}, s_{3307}$ (response logs are mentioned in Figure 2).

Case Study (Q7). Considering an educator conducts cognitive diagnosis with HyperCDM, and the results are shown in Figure 7. In (a), compared with other methods of Figure 2, three similar students are positioned closer, and they all exhibit deficiencies in nearly identical knowledge concepts (k_4, k_5, k_{11}), as shown in (c). This indicates that HyperCDM can effectively capture homogeneous influence among students. Moreover, HyperCDM can learn the implicit associations among concepts from the knowledge hypergraph. In (b), we apply K-means on knowledge concepts features ($K = 4$) and observe that semantically related concepts are clustered together. For example, k_3, k_6, k_7 are all related to function, and k_5, k_8 are geometry. In summary, HyperCDM effectively helps teachers identify similar students for tailored instruction and discover connections between knowledge points, enhancing teaching methods.

6 Conclusion

This paper first observes a key issue that is often neglected by previous work: the homogeneous influence among students. To capture the homogeneous influence, we propose a hypergraph cognitive diagnosis model (HyperCDM) to model flexible and high-order relationships among students. To avoid distortion, HyperCDM employs a divide-and-conquer strategy to learn different kinds of representations and unifies them via a feature-based interaction function. To construct the hypergraph based on sparse response logs, the auto-encoder is used to learn latent features to cluster students. To alleviate over-smoothing, momentum hypergraph convolution networks are designed. Experiment results show that HyperCDM excels in capturing the homogeneous influence. The future work of HyperCDM includes further enhancing its generalization and landing it on more real-world intelligent education scenarios.

Acknowledgments

We would like to thank the anonymous reviewers for their comprehensive and constructive reviews. The first author would like to extend special thanks to Xinyu Shi, whose unwavering support has enriched his brightest days during the research. This work is supported by the Natural Science Foundation of Shanghai (No. 21ZR1420300, 23ZR1418500), and Science and Technology Commission of Shanghai Municipality Grant (No. 22511105901).

References

[1] Ghodai Abdelrahman, Qing Wang, and Bernardo Pereira Nunes. 2023. Knowledge Tracing: A Survey. *ACM Computing Survey* 55, 11 (2023), 224:1–224:37.

[2] Ashton Anderson, Daniel P. Huttenlocher, Jon M. Kleinberg, and Jure Leskovec. 2014. Engaging with massive online courses. In *Proceedings of the ACM on Web Conference 2014*. Seoul, Korea.

[3] Albert Bandura and Richard H Walters. 1977. *Social Learning Theory*. Vol. 1. Englewood Cliffs Prentice Hall.

[4] Dexiong Chen, Leslie O'Bray, and Karsten M. Borgwardt. 2022. Structure-Aware Transformer for Graph Representation Learning. In *Proceedings of the 39th International Conference on Machine Learning*, Vol. 162. Baltimore, MD, 3469–3489.

[5] Youngduck Choi, Youngnam Lee, Dongmin Shin, Junghyun Cho, Seoyon Park, Seewoo Lee, Jineon Baek, Chan Bae, Byungsoo Kim, and Jaewe Heo. 2020. EdNet: A Large-Scale Hierarchical Dataset in Education. In *Proceedings of the 21st International Conference on Artificial Intelligence in Education*. Ifrane, Morocco, 69–73.

[6] Jiajun Cui, Zeyuan Chen, Aimin Zhou, Jianyong Wang, and Wei Zhang. 2023. Fine-Grained Interaction Modeling with Multi-Relational Transformer for Knowledge Tracing. *ACM Transactions on Information Systems* 41, 4 (2023), 104:1–104:26.

[7] Jimmy De La Torre. 2008. An Empirically Based Method of Q-matrix Validation for the DINA Model: Development and applications. *Journal of Educational Measurement* 45, 4 (2008), 343–362.

[8] Kaize Ding, Albert Jiongqian Liang, Bryan Perozzi, Ting Chen, Ruoxi Wang, Lichan Hong, Ed H. Chi, Huan Liu, and Derek Zhiyuan Cheng. 2023. HyperFormer: Learning Expressive Sparse Feature Representations via Hypergraph Transformer. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Taipei, Taiwan, 2062–2066.

[9] Paszke et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*. British Columbia, Canada, 8024–8035.

[10] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. Hypergraph Neural Networks. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*. Honolulu, HI, 3558–3565.

[11] Weibo Gao, Qi Liu, Zhenya Huang, Yu Yin, Haoyang Bi, Mu-Chun Wang, Jianhui Ma, Shijin Wang, and Yu Su. 2021. RCD: Relation Map Driven Cognitive Diagnosis for Intelligent Education Systems. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Virtual, 501–510.

[12] Yue Gao, Zizhao Zhang, Haojie Lin, Xibin Zhao, Shaoyi Du, and Changqing Zou. 2022. Hypergraph Learning: Methods and Practices. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 5 (2022), 2548–2566.

[13] Xavier Glorot and Yoshua Bengio. 2010. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In *Proceedings of the 13rd International Conference on Artificial Intelligence and Statistics*. Sardinia, Italy, 249–256.

[14] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems 30*. Long Beach, CA, 1024–1034.

[15] John A Hartigan and Manchek A Wong. 1979. Algorithm AS 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)* 28, 1 (1979), 100–108.

[16] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. virtual, 639–648.

[17] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer Feedforward Networks are Universal Approximators. *Neural Networks* 2, 5 (1989), 359–366.

[18] Lu Jiang, Kunpeng Liu, Yibin Wang, Dongjie Wang, Pengyang Wang, Yanjie Fu, and Minghao Yin. 2023. Reinforced Explainable Knowledge Concept Recommendation in MOOCs. *ACM Transactions on Intelligent Systems and Technology* 14, 3 (2023), 43:1–43:20.

[19] Aisha Urooj Khan, Hilde Kuehne, Bo Wu, Kim Chheu, Walid Bousselham, Chuang Gan, Niels da Vitoria Lobo, and Mubarak Shah. 2023. Learning Situation HyperGraphs for Video Question Answering. In *Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Vancouver, Canada.

[20] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations*. San Diego, CA.

[21] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*. Toulon, France.

[22] Jiatong Li, Fei Wang, Qi Liu, Mengxiao Zhu, Wei Huang, Zhenya Huang, Enhong Chen, Yu Su, and Shijin Wang. 2022. HierCDF: A Bayesian Network-based Hierarchical Cognitive Diagnosis Framework. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Washington, DC, 904–913.

[23] Mingjia Li, Hong Qian, Jinglan Lv, Mengliang He, Wei Zhang, and Aimin Zhou. 2024. Foundation Model Enhanced Derivative-Free Cognitive Diagnosis. *Frontiers of Computer Science* (2024).

[24] Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S. Yu, and Chuan Shi. 2023. Towards Graph Foundation Models: A Survey and Beyond. *CoRR* abs/2310.11829 (2023). arXiv:2310.11829

[25] Qi Liu. 2021. Towards a New Generation of Cognitive Diagnosis. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*. Montreal, Canada, 4961–4964.

[26] Qi Liu, Runze Wu, Enhong Chen, Guandong Xu, Yu Su, Zhigang Chen, and Guoping Hu. 2018. Fuzzy Cognitive Diagnosis for Modelling Examinee Performance. *ACM Transactions on Intelligent Systems and Technology* 9, 4 (2018), 48:1–48:26.

[27] Shuo Liu, Hong Qian, Mingjia Li, and Aimin Zhou. 2023. QCCDM: A Q-Augmented Causal Cognitive Diagnosis Model for Student Learning. In *Proceedings of the 26th European Conference on Artificial Intelligence*. Kraków, Poland, 1536–1543.

[28] Shuo Liu, Junhao Shen, Hong Qian, and Aimin Zhou. 2024. Inductive Cognitive Diagnosis for Fast Student Learning in Web-Based Intelligent Education Systems. In *Proceedings of the ACM on Web Conference 2024*. Singapore.

[29] Yingjie Liu, Tiancheng Zhang, Xuecen Wang, Ge Yu, and Tao Li. 2023. New development of cognitive diagnosis models. *Frontiers of Computer Science* 17, 1 (2023), 171604.

[30] Frederic Lord. 1952. A Theory of Test Scores. *Psychometric Monographs* (1952).

[31] Haiping Ma, Manwei Li, Le Wu, Haifeng Zhang, Yunbo Cao, Xingyi Zhang, and Xuemin Zhao. 2022. Knowledge-Sensed Cognitive Diagnosis for Intelligent Education Platforms. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*. Atlanta, GA, 1451–1460.

[32] Mark D. Reckase. 2009. *Multidimensional Item Response Theory Models*. Springer. 79–112 pages.

[33] D. Sculley. 2010. Web-scale k-means clustering. In *Proceedings of the ACM on Web Conference 2010*. Raleigh, NC, 1177–1178.

[34] Junhao Shen, Hong Qian, Wei Zhang, and Aimin Zhou. 2024. Symbolic Cognitive Diagnosis via Hybrid Optimization for Intelligent Education Systems. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*. Vancouver, Canada.

[35] Jianwen Sun, Fenghua Yu, Sannyuya Liu, Yawei Luo, Ruxia Liang, and Xiaoxuan Shen. 2023. Adversarial Bootstrapped Question Representation Learning for Knowledge Tracing. In *Proceedings of the 31st ACM International Conference on Multimedia*. Ottawa, Canada, 8016–8025.

[36] J. B. Sympson. 1978. A Model for Testing with Multidimensional Items. In *Proceedings of the 1977 Computerized Adaptive Testing Conference*. Minneapolis, MN, 82–98.

[37] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, 11 (2008).

[38] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph Attention Networks. *CoRR* abs/1710.10903 (2017). arXiv:1710.10903

[39] Fei Wang, Qi Liu, Enhong Chen, Zhenya Huang, Yuying Chen, Yu Yin, Zai Huang, and Shijin Wang. 2020. Neural Cognitive Diagnosis for Intelligent Education Systems. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*. New York, NY, 6153–6161.

[40] Fei Wang, Qi Liu, Enhong Chen, Zhenya Huang, Yu Yin, Shijin Wang, and Yu Su. 2022. NeuralCD: A General Framework for Cognitive Diagnosis. *IEEE Transactions on Knowledge and Data Engineering* 35, 8 (2022), 8312–8327.

[41] Shanshan Wang, Zhen Zeng, Xun Yang, and Xingyi Zhang. 2023. Self-Supervised Graph Learning for Long-Tailed Cognitive Diagnosis. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*. Washington, DC, 110–118.

[42] Zichao Wang, Angus Lamb, Evgeny Saveliev, Pashmina Cameron, Yordan Zaykov, José Miguel Hernández-Lobato, Richard E Turner, Richard G Baraniuk, Craig Barton, Simon Peyton Jones, Simon Woodhead, and Cheng Zhang. 2020. Diagnostic Questions: The NeurIPS 2020 Education Challenge. *arXiv preprint arXiv:2007.12061* (2020).

[43] Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy Huang. 2022. Hypergraph Contrastive Collaborative Filtering. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY.

[44] Bo Yang, Xiao Fu, Nicholas D. Sidiropoulos, and Mingyi Hong. 2017. Towards K-means-friendly Spaces: Simultaneous Deep Learning and Clustering. In *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70. Sydney, Australia, 3861–3870.

[45] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do Transformers Really Perform Badly for Graph Representation?. In *Advances in Neural Information Processing Systems 34*. virtual, 28877–28888.

[46] Yan Zhuang, Qi Liu, GuanHao Zhao, Zhenya Huang, Weizhe Huang, Zachary Pardos, Enhong Chen, Jinze Wu, and Xin Li. 2024. A Bounded Ability Estimation for Computerized Adaptive Testing. *Advances in Neural Information Processing Systems 36*.

Appendix

The appendix is organized as follows:

• Appendix A presents the details of datasets, methods, implementation, and explanation of proposed metrics HI and CI.

• Appendix B presents detailed statistics of performance on generalization, ablation study and other hyperparameter analysis results.

• Appendix C presents the limitations of our method.

A Experiment Details

A.1 Datasets Sources

We conduct experiments on four real-world datasets. Note that these datasets are open-accessible and do not involve ethical issues such as data collection. Their sources are introduced below.

Math1 and Math2. The datasets are collected from high school students who participate in their final exams during their first and second senior years, respectively. They include both objective and subjective problems. However, to align with previous work, we only consider the objective problems.

NeurIPS20. This dataset is used in the NeurIPS 2020 Education Challenge. It encompasses four semesters (September 2018 to May 2020) of students' answers to mathematics exercises from Eedi, a leading educational platform which students interact with daily from all around the world. We only use utilize the dataset offered for cognitive diagnosis in this competition.

EdNet-1. This dataset contains all student-system interactions collected over 2 years by Santa, a multi-platform AI tutoring service in Korea. The 1 means the first section of the huge dataset.

In the main paper, we state, "These datasets cover common learning scenarios and are representative," because they include both online and offline environments, as well as middle school and high school levels, and also cover students from different locations.

A.2 Baselines and State-of-the-Art Methods

We compare with six baseline and state-of-the-art methods including transitional CDMs, neural CDMs and the graph-based CDM. All of them are open-source and available at https://github.com/bigdata-ustc/EduCDM.

• **MIRT** [36] extends the IRT by adopting multidimensional vectors to model latent traits of students and exercises. The simple form of the interaction function is defined as

$$P(r_{ij}=1)=\frac{1}{1+e^{-1.7(\boldsymbol{a}_j^\top\boldsymbol{\theta}_i-\boldsymbol{\beta}_j)}},$$

where $P(r_{ij}=1)$ is the probability that student s_i answers the exercise e_j correctly, $\boldsymbol{a}_j\in\mathbb{R}^d$ is the discrimination of exercise j, $\boldsymbol{\theta}_i\in\mathbb{R}^d$ is the latent traits of students, and $\boldsymbol{\beta}_j\in\mathbb{R}^d$ is that of exercises. Note that $d\neq|K|$, i.e., each dimension of traits does not correspond to the specific knowledge concepts.

• **DINA** [7] is a conjunctive assumption-based model, where proficiency levels are denoted by discrete binary values. The interaction function is defined as

$$P(r_{ij}=1)=g_j^{1-\eta_{ij}}(1-s_j)^{\eta_{ij}},\quad \eta_{ij}=\prod_k\theta_{ij}^{\beta_{ij}},$$

where g_j is the probability of correctly guessing exercise e_j, s_j is the probability of making a careless mistake on exercise e_j, $\theta_{ij}\in\{0,1\}$ denotes whether student i masters knowledge concept j, and $\beta_{ij}\in 0,1$ donates whether exercise i includes concept j.

• **NCDM** [39] adopts neural networks in replacement of manually designed interaction functions and outperforms traditional CDMs on most datasets in generalization and interpretability. The interaction function are multi-layer position perceptron.

• **KaNCD** [40] builds upon the advancements of NCDM, considers implicit associations between knowledge concepts, and achieves state-of-the-art results across most datasets.

• **KSCD** [31] explores implicit relationships between knowledge concepts and exercises, employing a novel interaction function.

• **RCD** [11] explores intricate relationships among students, exercises, and knowledge attributes, employing a graph attention network to model these connections.

A.3 Implementation Details

In hypergraph construction, we set the number of clusters $\mathcal{K}=0.02|S|$, the dimension of latent response logs feature $\tilde{E}=64$; the network dimension of auto-encoder is $[512,256,128]$, the pre-training epoch is 100, and the K-means clustering epoch T is 50. In diagnostic layers, the network dimension of MLP is $[512,256,128]$. In MHGCN, the number of layers $L=4$, the momentum $\eta=0.8$, and the dimension of embeddings $d_{\text{emb}}=16$. In embedding conversion, the dimension of feature $d_{\text{feat}}=512$.

When selecting hyperparameters, we adopt the cross-validation method, which involves a portion of the training set as a validation set for parameter selection, and utilizing grid search to explore the parameters. Through the pilot study, we determined the hyperparameters for HyperCDM.

All models are implemented by Pytorch [9] framework and performed on CUDA 11.1. All experiments are conducted on a Linux server with Intel(R) Xeon(R) Gold 6130 CPU and NVIDIA V100-32GB GPU.

A.4 Metrics Explanation

To deeply understand HI and CI, we give an example to show how to evaluate models' performance on homogeneity via our proposed metrics. Assuming that we have two cognitive diagnostic models, A and B, and a response matrix formed by students s_1,s_2,s_3 and exercises e_1,e_2,e_3 as follows.

$$\boldsymbol{R}=\begin{bmatrix}1&-1&-1\\-1&1&1\\1&-1&-1\end{bmatrix},$$

where each column signifies the response logs of the corresponding student. The diagnostic outcomes after employing the cognitive diagnostic models would be as follows, and each column signifies the proficiency levels of the corresponding student.

$$\boldsymbol{\xi}_{\text{prof}_A}=\begin{bmatrix}0.4&0.7&0.7\\0.6&0.2&0.3\\0.9&0.2&0.1\end{bmatrix},\ \boldsymbol{\xi}_{\text{prof}_B}=\begin{bmatrix}0.5&0.8&0.6\\0.6&0.3&0.1\\0.9&0.4&0.2\end{bmatrix}.$$

We find that A significantly outperforms B. Given the similarity in students' interaction records, their proficiency levels should be equivalent.

Then we try to evaluate with proposed metrics. To calculate HI, we first compute the correctness rate of each student, i.e.,

Table 6: Performance of baselines, state-of-the-art methods, and HyperCDM in generalization. Acc., AUC and F1 are expressed in percentage. "↑" means larger values are better. In each column, an entry is marked in bold if its mean value is the best. By t-test, a bold one is significantly better than others on the corresponding metrics with significant level $\alpha = 5\%$.

Datasets	NeurIPS2020			EdNet-1			Math1			Math2		
Methods	Acc. (%)↑	AUC (%)↑	F1 (%)↑	Acc. (%)↑	AUC (%)↑	F1 (%)↑	Acc. (%)↑	AUC (%)↑	F1 (%)↑	Acc. (%)↑	AUC (%)↑	F1 (%)↑
MIRT	66.18	69.19	73.76	68.45	68.98	77.60	67.99	74.42	72.06	69.88	77.08	69.38
DINA	39.16	62.16	12.87	42.95	54.73	41.99	47.18	67.51	24.50	50.54	68.90	18.66
NCDM	71.79	**77.73**	78.59	70.50	72.70	79.83	67.59	74.25	72.03	69.36	76.87	69.99
KaNCD	71.86	77.53	78.75	71.76	74.51	80.35	68.31	75.11	74.24	70.28	78.10	70.32
KSCD	71.51	76.79	78.32	71.38	74.43	80.42	68.34	**75.26**	71.69	67.75	76.48	67.06
RCD	71.62	77.11	78.64	71.87	74.57	80.60	67.84	73.93	74.37	69.20	77.08	70.21
HyperCDM	**71.88**	77.32	**78.95**	**71.98**	**74.75**	**80.81**	**68.47**	74.76	**75.09**	**70.35**	**78.27**	**70.93**

Table 7: Ablation study of HyperCDM on F1 and CI. F1 is expressed in percentage. "↑" means larger values are better, while "↓" means smaller values are better. In each column, an entry is marked in bold if its mean value is the best. By t-test, a bold one is significantly better than others on the corresponding metrics with significant level $\alpha = 5\%$.

Datasets	NeurIPS20		EdNet-1		Math1		Math2	
Methods	F1(%)↑	CI↓	F1(%)↑	CI↓	F1(%)↑	CI↓	F1(%)↑	CI↓
HyperCDM w.o. Mon	78.07	0.1864	80.69	0.2034	73.01	0.0258	65.07	0.0579
HyperCDM w.o. Con	79.32	1.4363	80.09	2.1768	73.92	0.2541	68.32	0.5032
HyperCDM w.o. G	78.62	0.9631	80.81	1.7631	74.06	0.1325	67.86	0.2655
HyperCDM w.o. L	78.58	0.5064	80.90	0.8152	72.94	0.0832	69.20	0.2315
HyperCDM	**78.95**	**0.1486**	**80.99**	**0.1602**	**75.09**	**0.0107**	**70.93**	**0.0278**

Table 8: Hyperparameter analysis on two offline datasets. When investigating a particular hyperparameter, we maintain the settings of other hyperparameters as specified in Section A.3.

Hyperparameters		Math1						Math2					
		Acc.	AUC	F1	DOA	HI	CI	Acc.	AUC	F1	DOA	HI	CI
d_{emb}	8	67.84	74.34	73.84	56.99	0.0089	0.0097	70.62	78.86	70.96	64.26	0.0140	0.0244
	16	68.47	74.76	75.09	60.42	0.0099	0.0107	70.35	78.27	70.93	64.34	0.0159	0.0278
	32	67.70	74.33	72.74	60.56	0.0096	0.0104	69.85	77.29	70.86	64.48	0.0174	0.0301
	64	67.34	74.13	72.35	60.41	0.0082	0.0101	69.80	77.26	70.19	64.86	0.0150	0.0261
	128	64.91	73.36	74.80	60.83	0.0080	0.0149	69.62	76.95	69.98	65.02	0.0165	0.0289
d_{feat}	64	67.99	74.44	71.69	55.51	0.0057	0.0074	69.46	77.37	70.11	63.93	0.0127	0.0254
	128	67.93	74.41	74.59	58.02	0.0061	0.0085	69.74	77.80	70.51	64.07	0.0134	0.0265
	256	68.06	74.68	75.19	60.04	0.0070	0.0089	69.80	77.98	70.87	64.16	0.0142	0.0268
	512	68.47	74.76	75.09	60.42	0.0099	0.0107	70.35	78.27	70.93	64.34	0.0159	0.0278
	1024	68.22	74.39	75.23	58.13	0.0091	0.0086	70.46	78.34	71.01	64.59	0.0172	0.0288
L	1	67.09	73.62	72.52	59.14	0.0053	0.0077	69.68	77.21	69.49	63.80	0.0115	0.0199
	2	67.46	73.65	73.79	59.22	0.0051	0.0082	69.92	77.87	70.40	64.16	0.0126	0.0209
	3	68.02	74.36	74.66	59.14	0.0064	0.0096	70.12	78.12	70.35	64.18	0.0149	0.0213
	4	68.47	74.76	75.09	60.42	0.0099	0.0107	70.35	78.27	70.93	64.34	0.0159	0.0248
	5	67.37	73.73	72.34	58.22	0.0145	0.0279	69.52	77.02	69.68	65.11	0.0166	0.0265
η	0.0	68.32	74.30	73.01	59.97	0.0108	0.0258	69.35	77.76	65.07	63.46	0.0264	0.0107
	0.2	68.38	74.32	74.99	60.90	0.0125	0.0205	70.33	78.24	70.03	64.50	0.0187	0.0302
	0.4	68.03	74.16	74.91	60.12	0.0101	0.0164	70.15	78.22	70.49	64.17	0.0178	0.0290
	0.6	68.24	74.35	75.08	60.21	0.0092	0.0122	70.07	78.16	70.26	64.26	0.0151	0.0267
	0.8	68.47	74.76	75.09	60.42	0.0099	0.0107	70.35	78.27	70.93	64.34	0.0159	0.0278
	1	67.53	74.28	75.01	61.17	0.0074	0.0084	70.06	78.15	70.47	64.75	0.0148	0.0277

0.67, 0.33, 0.33 respectively. Then we have

$$HI_A \approx 0.1414\,,\ HI_B \approx 0.3464\,,$$

and $HI_A < HI_B$, validating that A outperforms B.

On the other hand, we try to calculate CI. At first we give the cosine similarity matrix

$$C = \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{bmatrix}.$$

And then we have

$$CI_A \approx -0.3610\,,\ CI_B \approx -0.2612\,,$$

where $CI_A < CI_B$, also validating that A outperforms B.

B Experiment Statistics

B.1 Generalization

The performance of baselines, state-of-the-art methods, and HyperCDM on generalization is shown in Table 6, and the variance of each entry is less than 0.05. By t-test, HyperCDM significantly outperforms other methods on Acc. and F1 with significant level $\alpha = 5\%$, and performs competitively with other methods on AUC.

B.2 Ablation Study

In addition to the metrics presented in Table 4 (i.e., Acc., AUC, HI, and DOA), we also conducted experiments involving F1 and CI, as depicted in Table 7. The results still indicate that HyperCDM outperforms all other HyperCDM variants.

B.3 Hyperparameter Analysis

We conducted a hyperparameter experiment on $d_{\text{feat}}, d_{\text{emb}}$ in Math1, as well as experiments on the number of layers L and momentum η in MHGCN across two offline datasets (without loss of generality, and similar to NeurIPS20 and EdNet-1), as depicted in Table 8. The $d_{\text{feat}} \in \{64, 128, 256, 512, 1024\}$, $d_{\text{emb}} \in \{8, 16, 32, 64, 128\}$, $L \in \{1, 2, 3, 4, 5\}$, and $\eta \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. We find the trend in hyperparameter analysis of Section 5.2 persists across all datasets: $d_{\text{emb}}, d_{\text{feat}}$ mutually constrain each other. Thus, to optimize performance, we can choose high-dimensional features, and improve the homogeneity and generalization by choosing low-dimensional embeddings (as we mentioned in Section 4.3). We also observe that a momentum value η around 0.8 is most suitable, since small values fail to mitigate over-smoothing, while overly large values result in the MHGCN consistently prioritizing representations of the previous layer. Additionally, $L = 4$ reaches optimal in most datasets, because shallow GNNs struggle to capture the structure of the graph, while overly deep ones may encounter over-smoothing.

C Limitations

Our model might occasionally struggle with scenarios involving millions of students because graph (including hypergraph) representation learning faces many challenges in the face of large-scale data. This paper is focused on how to effectively apply hypergraphs to capturing homogeneous influence among students, and we hope it can be addressed in the future work.