Relation-Aware Graph Learning with Mixture-of-Experts Prediction for Cognitive Diagnosis

Jingwei Qu[1*], **Mingze Zhang**[1], **Pingshun Zhang**[1], **Li Tao**[1], **Ying Wang**[1], **Zhaofang Yang**[1] and **Haibin Ling**[2]

[1]College of Computer and Information Science, Southwest University, Chongqing, China
[2]Intelligent Computing and Application Lab, Westlake University, Zhejiang, China
qujingwei@swu.edu.cn

Abstract

Cognitive diagnosis aims to infer students' concept-level mastery from exercise response logs and exercise-concept associations. Fully leveraging heterogeneous relations and modeling large mastery-difficulty variations remain challenging, especially with a single predictor. To address these challenges, we propose RMCD, a unified cognitive diagnosis model that integrates relation-aware graph learning with Mixture-of-Experts (MoE) prediction. RMCD constructs a heterogeneous relational graph over students, exercises, and concepts with multiple relation types, and learns node and edge representations simultaneously. It derives relation-strength vectors from student-concept and exercise-concept edges to distinguish relation effects and refine node representations. RMCD further introduces an MoE-based prediction head that adaptively combines multiple expert predictors to capture diverse mastery-difficulty discrepancies. Experiments on benchmark datasets demonstrate that RMCD consistently outperforms state-of-the-art cognitive diagnosis methods. Our algorithm is available at https://github.com/swu-qjw-lab/code/tree/main/RMCD.

1 Introduction

Cognitive diagnosis in intelligent education assesses students' mastery of knowledge concepts from exercise response logs [Anderson *et al.*, 2014; Burns *et al.*, 2014]. As shown in Fig. 1, a cognitive diagnosis model estimates each student's concept-level mastery from students' responses and exercise-concept associations, supporting applications such as learning path planning [Liu *et al.*, 2019] and resource recommendation [Kuh *et al.*, 2011]. As online education platforms and large-scale response data rapidly grow, accurately characterizing students' cognitive states from complex response patterns has become a fundamental challenge.

Early cognitive diagnosis studies are largely grounded in psychometric theories, including DINA [De La Torre, 2009], MIRT [Reckase, 2009], and IRT [Embretson and Reise, 2013]. These models are concise and interpretable, yet their limited expressiveness makes it difficult to accommodate diverse response patterns. As deep learning developed, some methods leverage neural networks to model student-exercise interactions [Wang *et al.*, 2020]. Recently, graph-based cognitive diagnosis methods have gained attention by leveraging graph structure to represent students, exercises, concepts, and their relations [Gao *et al.*, 2021; Wang *et al.*, 2023b]. Despite this progress, these methods still face challenges in both relation modeling and prediction mechanisms.

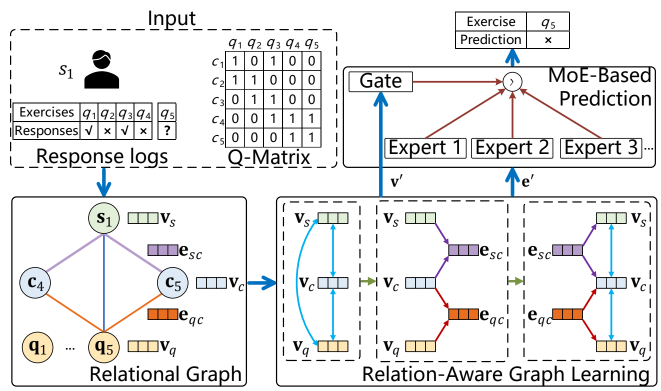

Figure 1: Conceptual illustration of RMCD: integrating relation-aware graph learning with MoE-based prediction.

First, holistic heterogeneous relations are still underexplored. Existing graph-based methods usually model only partial relations, such as student-exercise interactions or exercise-concept links, making it hard to distinguish different relation effects. Second, a single predictor has limited capacity to characterize students' response performance. Correctness depends on student mastery and exercise difficulty, which vary widely in real-world educational data. Most methods model these factors with one predictor for all students and exercises, making it difficult to capture such variations and limiting prediction accuracy.

To address these challenges, we propose RMCD (*Relation-Aware Graph Learning with Mixture-of-Experts Prediction*), which integrates relation-aware graph learning with a Mixture-of-Experts (MoE) prediction head for cognitive diagnosis. First, RMCD constructs a heterogeneous relational graph that jointly represents students, exercises, and concepts

*Corresponding author.

with student-exercise, exercise-concept, and student-concept relations. On this graph, a relation-aware graph encoder jointly learns node and edge representations. In particular, it learns dedicated representations for student-concept and exercise-concept edges and maps them to relation-strength vectors, which distinguish relation effects and are fed back to refine node representations. Second, RMCD introduces an MoE-based prediction head to overcome the limitation of a single predictor. Conditioned on the representations of students, exercises, and concepts, it adaptively combines multiple expert predictors to estimate students' response performance. The experts capture diverse concept-level mastery-difficulty discrepancies, yielding a flexible prediction mechanism for varied student mastery and exercise difficulty.

Our main contributions include: 1) We build a relational graph over students, exercises, and concepts with student-exercise, exercise-concept, and student-concept relations. 2) We propose a relation-aware graph encoder that jointly learns node and edge representations and derives relation-strength vectors to differentiate relation effects and refine nodes. 3) We design an MoE-based prediction head that conditions on student, exercise, and concept representations and adaptively combines experts to model diverse mastery-difficulty discrepancies. To the best of our knowledge, this is the first work that introduces MoE into cognitive diagnosis.

2 Related Work

2.1 Cognitive Diagnosis

Cognitive diagnosis aims to infer students' mastery of knowledge concepts from response records. Early psychometric models include IRT [Embretson and Reise, 2013], MIRT [Reckase, 2009], and DINA [De La Torre, 2009], where DINA models student-concept mastery with guessing and slipping for interpretability. NCD [Wang *et al.*, 2020] uses neural networks to capture nonlinear student-exercise interactions. Recent graph-based methods jointly model students, exercises, and concepts to exploit higher-order relations, *e.g.*, RCD [Gao *et al.*, 2021] and SCD [Wang *et al.*, 2023b], with ISGCD [Shao *et al.*, 2025] furtherconsidering edge heterogeneity and uncertainty. Beyond structural relations, ACD [Wang *et al.*, 2024] incorporates students' affective states. Inspired by graph-based cognitive diagnosis, RMCD constructs the relational graph with student-exercise, exercise-concept, and student-concept relations, and learns node and edge representations simultaneously. Relation-strength vectors derived from student-concept and exercise-concept edges further differentiate relation effects.

2.2 Mixture of Experts

MoE is a conditional computation framework that uses a gating function to route inputs to different experts [Jacobs *et al.*, 1991; Masoudnia and Ebrahimpour, 2014]. MoE has dense or sparse variants. Dense MoE evaluates all experts and aggregates their outputs with mixture weights [Eigen *et al.*, 2013; Masoudnia and Ebrahimpour, 2014], while sparse MoE activates only a subset of experts for controlled computation [Shazeer *et al.*, 2017; Fedus *et al.*, 2022]. Both variants have been explored in large models and graph learning [Lin *et al.*, 2026; Wang *et al.*, 2023a; Zeng *et al.*, 2023; Zhou *et al.*, 2025]. In RMCD, we adopt a dense MoE prediction head tailored to cognitive diagnosis. Experts operate on concept-level mastery-difficulty discrepancies, and a gating function conditioned on student, exercise, and concept representations adaptively combines expert outputs. This design provides a flexible prediction mechanism under substantial variations in mastery and difficulty.

2.3 Graph Neural Networks

GNNs learn graph representations by aggregating neighborhood information [Wu *et al.*, 2020; Zhou *et al.*, 2020]. GCN [Kipf, 2016] and GAT [Veličković *et al.*, 2018] are representative homogeneous GNNs, while heterogeneous GNNs such as R-GCN [Schlichtkrull *et al.*, 2018] and HAN [Wang *et al.*, 2019] perform type-aware message passing over multi-typed nodes and relations. Recent studies further investigate meta-path modeling and analysis in heterogeneous graphs [Li *et al.*, 2024; Li *et al.*, 2025]. Different from these methods, RMCD parameterizes key educational relations with learnable edge representations and relation-strength vectors, which are fed back to refine node representations and capture relation effects critical for cognitive diagnosis.

3 Problem Formulation

3.1 Problem Definition

Given a set of students, exercises, and knowledge concepts, we define the cognitive diagnosis problem as follows:
Input: A student set $\{s_i\}_{i=1}^{n_s}$, an exercise set $\{q_j\}_{j=1}^{n_q}$, and a knowledge concept set $\{c_k\}_{k=1}^{n_c}$. In addition, a Q-matrix $\mathbf{Q} \in \{0,1\}^{n_q \times n_c}$ that describes the exercise-concept association is provided, where each element $\mathbf{Q}_{jk} = 1$ indicates that exercise q_j involves concept c_k and $\mathbf{Q}_{jk} = 0$ otherwise. An observed historical response log is denoted by $\mathbb{L} = \{(s_i, q_j, r_{ij})\}$, where r_{ij} denotes the response of student s_i on exercise q_j, *i.e.*, $r_{ij} = 1$ if correct and 0 otherwise.
Output: A diagnosis model $\mathcal{D}_\theta$ estimates the correctness probability of a student on an exercise, thereby implicitly characterizing students' concept mastery from $\mathbb{L}$ and $\mathbf{Q}$. For any unobserved student-exercise pair (s_i, q_j), the model outputs the correctness probability:

$$\hat{r}_{ij} = \mathcal{D}_\theta(s_i, q_j; \mathbf{Q}, \mathbb{L}) \in [0,1] \quad (s_i, q_j, \cdot) \notin \mathbb{L}, \tag{1}$$

where θ denotes the model parameters.

3.2 Graph Formulation

We construct a heterogeneous relational graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ to model the heterogeneous relations among students, exercises, and concepts in cognitive diagnosis. Each entity (student, exercise, or concept) is associated with a node. The node set is defined as $\mathbb{V} = \mathbb{V}_s \cup \mathbb{V}_q \cup \mathbb{V}_c$, where $\mathbb{V}_s = \{\mathbf{s}_i\}_{i=1}^{n_s}$, $\mathbb{V}_q = \{\mathbf{q}_j\}_{j=1}^{n_q}$, and $\mathbb{V}_c = \{\mathbf{c}_k\}_{k=1}^{n_c}$ denote the three types of nodes corresponding to students, exercises, and concepts, respectively. Each node is assigned a one-hot vector $\mathbf{s}_i \in \{0,1\}^{n_s}$, $\mathbf{q}_j \in \{0,1\}^{n_q}$, or $\mathbf{c}_k \in \{0,1\}^{n_c}$.

The edge set is defined as $\mathbb{E} = \mathbb{E}_{sq} \cup \mathbb{E}_{sc} \cup \mathbb{E}_{qc}$. Here, $\mathbb{E}_{sq} \subseteq \mathbb{V}_s \times \mathbb{V}_q$ represents student-exercise interactions

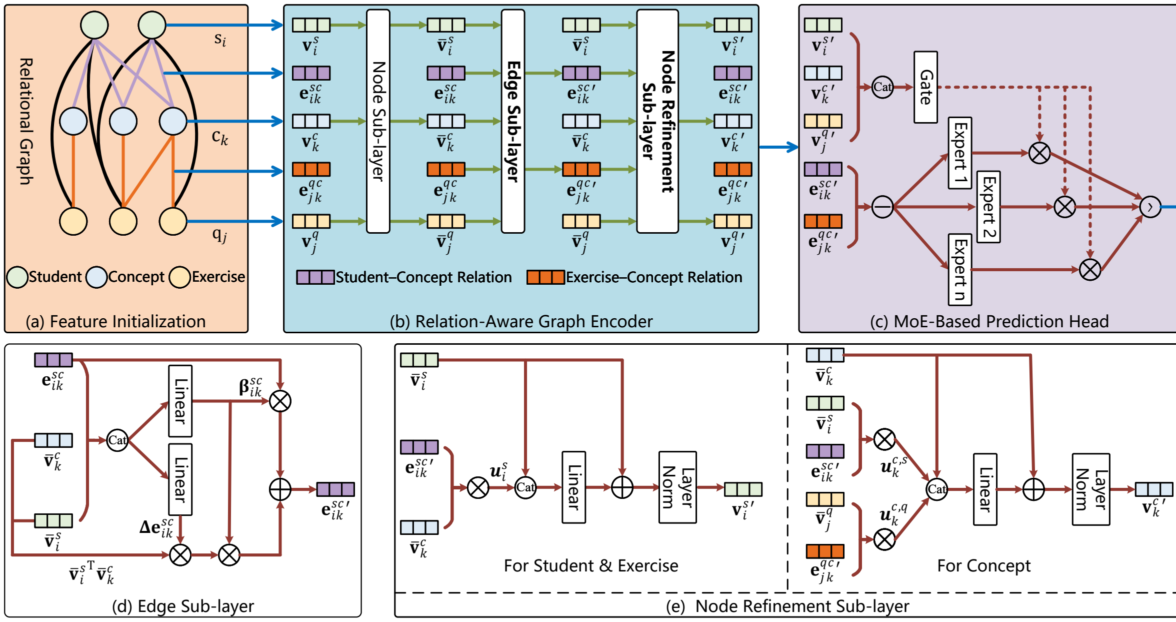

Figure 2: Overview of RMCD. RMCD first initializes student, exercise, and concept node features with learnable student-concept and exercise-concept edges. Next, the relation-aware graph encoder updates nodes and edges to derive concept-level mastery and difficulty strengths. Finally, the MoE-based prediction head with adaptive gating predicts correctness probabilities. Some details are omitted for clarity.

recorded in $\mathbb{L}$, where an edge $(\mathbf{s}_i, \mathbf{q}_j) \in \mathbb{E}_{sq}$ is included if $(s_i, q_j, r_{ij}) \in \mathbb{L}$. Moreover, $\mathbb{E}_{qc} \subseteq \mathbb{V}_q \times \mathbb{V}_c$ encodes exercise-concept associations induced by $\mathbf{Q}$: for each $\mathbf{Q}_{jk} = 1$, we include an edge $(\mathbf{q}_j, \mathbf{c}_k) \in \mathbb{E}_{qc}$. To explicitly model student-concept relations, we define $\mathbb{E}_{sc} \subseteq \mathbb{V}_s \times \mathbb{V}_c$ by linking a student to the concepts involved in the exercises the student has interacted with, *i.e.*,

$$\mathbb{E}_{sc} = \{(\mathbf{s}_i, \mathbf{c}_k) \mid \exists\, j \text{ s.t. } (s_i, q_j, r_{ij}) \in \mathbb{L} \wedge \mathbf{Q}_{jk} = 1\}\,. \quad (2)$$

4 Methodology

As illustrated in Fig. 2, RMCD contains three main stages:

- **Feature Initialization.** We project one-hot student, exercise, and concept nodes to dense representations, and initialize learnable features on student-concept and exercise-concept edges as inputs for graph encoding.
- **Relation-Aware Graph Encoder.** The encoder learns node and edge representations through three sub-layers: node updating, edge updating, and node refinement. It further map edge representations to relation-strength vectors, which support edge updating, node refinement, and concept-level mastery-difficulty modeling.
- **MoE-Based Prediction Head.** The prediction head uses multiple experts to model diverse mastery-difficulty discrepancies, with a gating function adaptively combines expert outputs for final prediction.

4.1 Feature Initialization

Given the one-hot node vectors defined in Sec. 3.2, we map students, exercises, and concepts to initial node representations $\mathbf{v}_i^s, \mathbf{v}_j^q, \mathbf{v}_k^c \in \mathbb{R}^{d_v}$ via linear projections:

$$\mathbf{v}_i^s = \mathbf{W}^{s\top} \mathbf{s}_i, \quad \mathbf{v}_j^q = \mathbf{W}^{q\top} \mathbf{q}_j, \quad \mathbf{v}_k^c = \mathbf{W}^{c\top} \mathbf{c}_k, \quad (3)$$

where $\mathbf{W}^s \in \mathbb{R}^{n_s \times d_v}$, $\mathbf{W}^q \in \mathbb{R}^{n_q \times d_v}$ and $\mathbf{W}^c \in \mathbb{R}^{n_c \times d_v}$ are learnable projection matrices, and d_v denotes the hidden dimension of the node representations.

We further introduce the edge representations for student-concept and exercise-concept relations. For each edge $(\mathbf{s}_i, \mathbf{c}_k) \in \mathbb{E}_{sc}$ and $(\mathbf{q}_j, \mathbf{c}_k) \in \mathbb{E}_{qc}$, we associate a trainable vector $\mathbf{e}_{ik}^{sc} \in \mathbb{R}^{d_e}$ and $\mathbf{e}_{jk}^{qc} \in \mathbb{R}^{d_e}$, respectively, where d_e denotes the hidden dimension of edge representations. These representations are randomly initialized as:

$$\mathbf{e}_{ik}^{sc} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \mathbf{I}\right), \quad \mathbf{e}_{jk}^{qc} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \mathbf{I}\right), \quad (4)$$

where σ is a hyperparameter controlling the initialization standard deviation. Student-exercise edges in $\mathbb{E}_{sq}$ only encode observed interactions from the response log $\mathbb{L}$ and do not carry additional learnable edge features in this stage.

4.2 Relation-Aware Graph Encoder

The relation-aware graph encoder learns representations on $\mathbb{G}$ by stacking n_l GNN layers (Fig. 2(b)). In each layer, we update node and edge representations through three sub-layers: a node sub-layer, an edge sub-layer, and a node refinement sub-layer. In particular, node representations are jointly updated by the node sub-layer and the node refinement sub-layer. For notation, we use unprimed and primed symbols to denote variables before and after an update (*e.g.*, $\mathbf{v}, \mathbf{v}'$).

Node Sub-layer. For each node, we obtain its intermediate representation via an attention-based aggregation function with residual connections. Specifically, we adopt an off-the-shelf heterogeneous GNN operator $\Phi(\cdot)$ to mean-aggregate neighbor representations along the student-exercise edges $\mathbb{E}_{sq}$ and the exercise-concept edges $\mathbb{E}_{qc}$, thereby producing residual updates:

$$\left(\Delta\mathbf{v}_i^s, \Delta\mathbf{v}_j^q, \Delta\mathbf{v}_k^c\right) = \Phi\left(\mathbf{v}_i^s, \mathbf{v}_j^q, \mathbf{v}_k^c; \mathbb{E}_{sq}, \mathbb{E}_{qc}\right). \tag{5}$$

Then the intermediate node representations are computed with residual connections:

$$\bar{\mathbf{v}}_i^s = \mathbf{v}_i^s + \Delta\mathbf{v}_i^s, \quad \bar{\mathbf{v}}_j^q = \mathbf{v}_j^q + \Delta\mathbf{v}_j^q, \quad \bar{\mathbf{v}}_k^c = \mathbf{v}_k^c + \Delta\mathbf{v}_k^c. \tag{6}$$

The updated node representations will be obtained in the subsequent node refinement sub-layer.

Edge Sub-layer. We update edge representations on $\mathbb{E}_{sc}$ and $\mathbb{E}_{qc}$ using the current edge state and the intermediate representations of the two incident nodes (Fig. 2(d)). For each student-concept edge $(\mathbf{s}_i, \mathbf{c}_k) \in \mathbb{E}_{sc}$, we first compute a relation-strength vector

$$\mathbf{p}_{ik}^{sc} = \Psi(\mathbf{e}_{ik}^{sc}), \tag{7}$$

where $\Psi(\cdot)$ denotes an element-wise squashing function that maps vectors to $(0, 1)$. Given the relation-strength $\mathbf{p}_{ik}^{sc}$ and the incident node representations $(\bar{\mathbf{v}}_i^s, \bar{\mathbf{v}}_k^c)$, we then compute (i) an edge update vector $\Delta\mathbf{e}_{ik}^{sc}$, (ii) a similarity-based scaling factor α_{ik}^{sc} from the node inner product, and (iii) a balancing vector $\boldsymbol{\beta}_{ik}^{sc}$ to trade off the current edge representation and the scaled update.

$$\begin{aligned}
\Delta\mathbf{e}_{ik}^{sc} &= \mathcal{F}_{sc}([\mathbf{p}_{ik}^{sc}; \bar{\mathbf{v}}_i^s; \bar{\mathbf{v}}_k^c]), \\
\alpha_{ik}^{sc} &= \Psi\left(\bar{\mathbf{v}}_i^{s\top} \bar{\mathbf{v}}_k^c\right), \\
\boldsymbol{\beta}_{ik}^{sc} &= \Psi(\mathcal{H}_{sc}([\mathbf{p}_{ik}^{sc}; \bar{\mathbf{v}}_i^s; \bar{\mathbf{v}}_k^c])), \\
\mathbf{e}_{ik}^{sc\prime} &= \boldsymbol{\beta}_{ik}^{sc} \odot \mathbf{e}_{ik}^{sc} + (1 - \boldsymbol{\beta}_{ik}^{sc}) \odot (\alpha_{ik}^{sc}\, \Delta\mathbf{e}_{ik}^{sc}),
\end{aligned} \tag{8}$$

where $[\cdot]$ denotes concatenation, $\odot$ is the element-wise product, $\mathcal{F}_{sc}(\cdot)$ and $\mathcal{H}_{sc}(\cdot)$ are learnable functions.

Similarly, for each exercise-concept edge $(\mathbf{q}_j, \mathbf{c}_k) \in \mathbb{E}_{qc}$, we update the representation $\mathbf{e}_{jk}^{qc}$ analogously following Eqs. (7)-(8), using its context $[\mathbf{p}_{jk}^{qc}; \bar{\mathbf{v}}_j^q; \bar{\mathbf{v}}_k^c]$ and the functions $\mathcal{F}_{qc}(\cdot)$ and $\mathcal{H}_{qc}(\cdot)$, producing $\mathbf{e}_{jk}^{qc\prime}$.

Node Refinement Sub-layer. Given the updated edge representations $\mathbf{e}_{ik}^{sc\prime}$ and $\mathbf{e}_{jk}^{qc\prime}$, we first compute the updated relation-strength vectors $\mathbf{p}_{ik}^{sc\prime}$ and $\mathbf{p}_{jk}^{qc\prime}$ by Eq. (7). We then refine node representations by aggregating neighbor information weighted by these strengths, followed by residual connections and layer normalization.

As shown in Fig. 2(e), for each student node, we aggregate neighbor concept representations weighted by the student-concept strengths $\mathbf{p}_{ik}^{sc\prime}$ and update the node representation as:

$$\begin{aligned}
\mathbf{u}_i^s &= \sum_{k:(\mathbf{s}_i, \mathbf{c}_k) \in \mathbb{E}_{sc}} \mathbf{p}_{ik}^{sc\prime} \odot \bar{\mathbf{v}}_k^c, \\
\mathbf{v}_i^{s\prime} &= \mathrm{LN}(\bar{\mathbf{v}}_i^s + \mathcal{F}_s([\bar{\mathbf{v}}_i^s; \mathbf{u}_i^s])).
\end{aligned} \tag{9}$$

Similarly, for each exercise node, the representation is refined based on the exercise-concept strengths $\mathbf{p}_{jk}^{qc\prime}$:

$$\begin{aligned}
\mathbf{u}_j^q &= \sum_{k:(\mathbf{q}_j, \mathbf{c}_k) \in \mathbb{E}_{qc}} \mathbf{p}_{jk}^{qc\prime} \odot \bar{\mathbf{v}}_k^c, \\
\mathbf{v}_j^{q\prime} &= \mathrm{LN}\left(\bar{\mathbf{v}}_j^q + \mathcal{F}_q\left([\bar{\mathbf{v}}_j^q; \mathbf{u}_j^q]\right)\right).
\end{aligned} \tag{10}$$

For each concept node, we fuse neighbor representations from both the student nodes and the exercise nodes:

$$\begin{aligned}
\mathbf{u}_k^{c,s} &= \sum_{i:(\mathbf{s}_i, \mathbf{c}_k) \in \mathbb{E}_{sc}} \mathbf{p}_{ik}^{sc\prime} \odot \bar{\mathbf{v}}_i^s, \\
\mathbf{u}_k^{c,q} &= \sum_{j:(\mathbf{q}_j, \mathbf{c}_k) \in \mathbb{E}_{qc}} \mathbf{p}_{jk}^{qc\prime} \odot \bar{\mathbf{v}}_j^q, \\
\mathbf{v}_k^{c\prime} &= \mathrm{LN}(\bar{\mathbf{v}}_k^c + \mathcal{F}_c([\bar{\mathbf{v}}_k^c; \mathbf{u}_k^{c,s}; \mathbf{u}_k^{c,q}])),
\end{aligned} \tag{11}$$

where $\mathcal{F}_s(\cdot)$, $\mathcal{F}_q(\cdot)$ and $\mathcal{F}_c(\cdot)$ are learnable functions.

Batch-wise Relation-Strength. After n_l layers of the graph encoder, we obtain the final edge representations on $\mathbb{E}_{sc}$ and $\mathbb{E}_{qc}$. To support mini-batch training, we sample a batch of B student-exercise interactions from the response log $\mathbb{L}$, yielding index pairs $\{(i_b, j_b)\}_{b=1}^{B}$ such that $(s_{i_b}, q_{j_b}, r_{i_b j_b}) \in \mathbb{L}$. For each sample index b and each concept index k, we collect the corresponding student-concept and exercise-concept relation-strength vectors to form two 3D tensors $\mathbf{M} \in \mathbb{R}^{B \times n_c \times d_e}$ and $\mathbf{D} \in \mathbb{R}^{B \times n_c \times d_e}$. For a required concept without a corresponding student-concept edge, which may occur when predicting an unobserved student-exercise pair, we use a hybrid estimate $\tilde{\mathbf{p}}_{i_b k}^{sc} = \frac{1}{2}(\bar{\mathbf{p}}_{i_b}^s + \bar{\mathbf{p}}_k^c)$, where $\bar{\mathbf{p}}_{i_b}^s$ and $\bar{\mathbf{p}}_k^c$ denote the average strengths over the existing student-concept edges incident to student s_{i_b} and concept c_k, respectively:

$$\begin{aligned}
\mathbf{M}_{b,k,:} &= \begin{cases} \mathbf{p}_{i_b k}^{sc}, & (\mathbf{s}_{i_b}, \mathbf{c}_k) \in \mathbb{E}_{sc}, \\ \tilde{\mathbf{p}}_{i_b k}^{sc}, & (\mathbf{s}_{i_b}, \mathbf{c}_k) \notin \mathbb{E}_{sc},\ \mathbf{Q}_{j_b k} = 1, \\ \mathbf{0}, & \mathbf{Q}_{j_b k} = 0, \end{cases} \\
\mathbf{D}_{b,k,:} &= \begin{cases} \mathbf{p}_{j_b k}^{qc}, & (\mathbf{q}_{j_b}, \mathbf{c}_k) \in \mathbb{E}_{qc}, \\ \mathbf{0}, & \text{otherwise}. \end{cases}
\end{aligned} \tag{12}$$

Here, $\mathbf{p}_{i_b k}^{sc}$ and $\mathbf{p}_{j_b k}^{qc}$ are the final relation-strength vectors obtained by applying Eq. (7) to the final edge representations.

The relation-strength vectors on student-concept and exercise-concept edges provide concept-level diagnostic factors for prediction. The student-concept strength reflects the mastery state of a student over a concept, while the exercise-concept strength captures the difficulty level of an exercise with respect to that concept. The prediction head jointly models these two factors to estimate the correctness probability.

4.3 MoE-based Prediction Head

The MoE-based prediction head estimates the correctness probability for each sampled student-exercise pair (Fig. 2(c)). It uses multiple experts to capture varied mastery-difficulty discrepancies and a gating function to adaptively combine expert outputs. Given the batch-wise tensors $\mathbf{M}$ and $\mathbf{D}$, we construct the discrepancy tensor $\mathbf{X} \in \mathbb{R}^{B \times n_c \times d_e}$:

$$\mathbf{X}_{b,k,:} = \mathbf{M}_{b,k,:} - \mathbf{D}_{b,k,:}, \tag{13}$$

where $\mathbf{X}_{b,k,:}$ represents the discrepancy between the mastery of student s_{i_b} on concept c_k and the difficulty of exercise q_{j_b} with respect to c_k. This discrepancy serves as the core input to the expert predictors.

Gating Function. We instantiate n_e expert predictors $\{\mathcal{E}_m\}_{m=1}^{n_e}$ and compute their mixture expert weights using a gating function $\mathcal{G}(\cdot)$. For each interaction (s_{i_b}, q_{j_b}) and concept c_k, $\mathcal{G}(\cdot)$ outputs the mixture weights $\boldsymbol{\pi}_{b,k} \in (0,1)^{n_e}$ conditioned on the corresponding student, exercise, and concept representations:

$$\mathbf{g}_{b,k} = \begin{cases} \left[\mathbf{v}^s_{i_b};\, \mathbf{v}^q_{j_b};\, \mathbf{v}^c_k\right], & \mathbf{Q}_{j_b k} = 1, \\ \mathbf{0}, & \mathbf{Q}_{j_b k} = 0, \end{cases}$$
$$\boldsymbol{\pi}_{b,k} = \mathrm{softmax}\left(\frac{\mathcal{G}(\mathbf{g}_{b,k})}{\tau}\right) \in \mathbb{R}^{n_e}. \tag{14}$$

Here, τ is a temperature parameter that controls the smoothness of expert selection.

Expert Mixture Prediction. Given the discrepancy vector $\mathbf{X}_{b,k,:}$, each expert $\mathcal{E}_m$ produces a scalar logit, and the concept-level prediction is obtained by aggregating expert outputs with the mixture weights:

$$\hat{r}_{b,k} = \Psi\left(\sum_{m=1}^{n_e} \pi_{b,k}^{(m)}\, \mathcal{E}_m(\mathbf{X}_{b,k,:})\right). \tag{15}$$

Finally, predictions are aggregated over the concepts involved in the exercise q_{j_b} to obtain the predicted correctness probability $\hat{r}_{i_b j_b}$ for the student-exercise pair (s_{i_b}, q_{j_b}):

$$\hat{r}_{i_b j_b} = \frac{\sum_{k=1}^{n_c} \mathbf{Q}_{j_b k}\, \hat{r}_{b,k}}{\sum_{k=1}^{n_c} \mathbf{Q}_{j_b k}}. \tag{16}$$

4.4 Training Objective

We train RMCD by minimizing a prediction loss with an MoE load-balancing regularizer. For the batch $\{(i_b, j_b)\}_{b=1}^{B}$ sampled from the response log $\mathbb{L}$, we optimize the correctness probability $\hat{r}_{i_b j_b}$ against the ground-truth response $r_{i_b j_b}$. The prediction loss is instantiated as the binary cross-entropy:

$$\mathcal{L}_p = -\sum_{b=1}^{B} r_{i_b j_b} \log \hat{r}_{i_b j_b} + (1 - r_{i_b j_b}) \log(1 - \hat{r}_{i_b j_b}). \tag{17}$$

To prevent the gating function from collapsing to a few experts, we introduce a load-balancing regularizer to encourage balanced expert utilization during training. For each expert $\mathcal{E}_m$, we compute its average mixture weight p_m over the concept positions with $\mathbf{Q}_{j_b k} = 1$ in the batch, and penalize the deviation of p_m from the uniform usage $\frac{1}{n_e}$:

$$p_m = \frac{\sum_{b,k} \mathbf{Q}_{j_b k}\, \pi_{b,k}^{(m)}}{\sum_{b,k} \mathbf{Q}_{j_b k}}, \quad \mathcal{L}_r = \sum_{m=1}^{n_e} \left(p_m - \frac{1}{n_e}\right)^2. \tag{18}$$

Finally, the overall objective is:

$$\mathcal{L} = \mathcal{L}_p + \lambda\, \mathcal{L}_r, \tag{19}$$

where λ controls the strength of load balancing.

5 Experiments

We evaluate the performance of RMCD by comparing it with various methods for cognitive diagnosis. Additionally, we conduct ablation studies for further evaluation and analysis.

Statistics	ASSIST17	ASSIST09	Junyi
#Students	1,709	2,493	10,000
#Exercises	3,162	17,746	835
#Knowledge concepts	102	123	835
#Response logs	390281	267,415	353,835
#Response logs per student	228.37	107.27	35.38

Table 1: Statistics of the ASSIST17, ASSIST09, and Junyi datasets.

5.1 Experimental Setup

Evaluation Benchmarks. We conduct experiments on three real-world cognitive diagnosis datasets: ASSIST17 [Feng *et al.*, 2009], ASSIST09 [Feng *et al.*, 2009], and Junyi [Chang *et al.*, 2015]. Following common practice, we remove students with fewer than 15 response logs to ensure sufficient observations for reliable modeling. Dataset statistics are summarized in Tab. 1, and each dataset is split into 80% training and 20% testing.

Implementation Details. RMCD is implemented in PyTorch [Paszke *et al.*, 2019] and optimized with Adam [Kingma, 2014]. We set the hidden dimensions to $d_v = d_e = 256$ and use a mini-batch size of $B = 256$. The graph encoder stacks $n_l = 1$ relation-aware GNN layer, where Eq. (5) is instantiated with GAT [Veličković *et al.*, 2018]. The function $\Psi(\cdot)$ is implemented as sigmoid, and all $\mathcal{F}(\cdot)$ and $\mathcal{H}(\cdot)$ in the encoder are fully-connected (FC) layers. The gate $\mathcal{G}(\cdot)$ is a two-layer MLP with ReLU, and each expert is an FC layer. On ASSIST17/ASSIST09/Junyi, we set $(n_e, \eta, \lambda) = (2, 5{\times}10^{-5}, 0.4)$, $(16, 5{\times}10^{-4}, 0.01)$, and $(2, 5{\times}10^{-5}, 0.3)$, respectively, where η denotes the learning rate. All experiments are conducted on a single NVIDIA GeForce RTX 3090 GPU with an Intel Core i9-12900K CPU.

5.2 Evaluation Results

We compare RMCD with eight cognitive diagnosis models: IRT [Embretson and Reise, 2013], MIRT [Reckase, 2009], DINA [De La Torre, 2009], NCD [Wang *et al.*, 2020], RCD [Gao *et al.*, 2021], SCD [Wang *et al.*, 2023b], ACD [Wang *et al.*, 2024], and ISGCD [Shao *et al.*, 2025]. All models are evaluated using ACC, RMSE, and AUC. For baselines, we adopt the reported results on ASSIST17 and Junyi from ACD, and those on ASSIST09 from RCD, except for DINA, SCD, and ISGCD as described below. Since the available results of DINA and SCD on ASSIST09 and those of ISGCD on all three datasets are not fully aligned with our evaluation protocol, we re-train these models under the same data split and original hyperparameter settings. For RMCD and re-trained baselines, we report mean and standard deviation over five runs.

Table 2 shows that RMCD consistently outperforms all compared models on all three datasets across ACC, RMSE, and AUC, with the largest gains on ASSIST17. This is consistent with the denser relations in ASSIST17 given its smaller student set and richer exercise and concept structures (Tab. 1). Moreover, RMCD outperforms ACD on ASSIST17 across all metrics without using affect-related information, where we report the best ACD variant, *i.e.*, SCD augmented with affective states. Since ASSIST17 provides affect labels while

Model	ASSIST17			ASSIST09			Junyi		
	ACC	RMSE	AUC	ACC	RMSE	AUC	ACC	RMSE	AUC
DINA	64.84 ± 0.09	46.06 ± 0.02	69.64 ± 0.06	69.36 ± 0.10	46.37 ± 0.02	72.44 ± 0.09	74.22	41.76	78.72
IRT	65.96 ± 0.10	46.53 ± 0.03	72.37 ± 0.07	64.26	46.59	69.83	67.60	42.68	77.50
MIRT	68.17 ± 0.02	46.48 ± 0.03	74.13 ± 0.00	71.70	45.17	74.94	75.13	41.17	79.89
NCD	69.21 ± 0.87	45.13 ± 0.60	75.34 ± 1.01	73.14	43.08	75.94	74.43	41.72	79.09
RCD	71.55 ± 0.15	43.39 ± 0.10	78.10 ± 0.03	73.55	42.13	77.21	77.16	39.63	82.62
SCD	71.59 ± 0.10	43.35 ± 0.05	78.19 ± 0.01	73.45 ± 0.13	42.26 ± 0.05	77.00 ± 0.10	77.30	39.61	82.77
SCD(ACD)	72.69 ± 0.05	42.79 ± 0.02	79.40 ± 0.07	–	–	–	77.45	39.59	82.90
ISGCD	71.68 ± 0.18	43.24 ± 0.07	78.61 ± 0.03	74.15 ± 0.11	41.80 ± 0.07	77.94 ± 0.06	77.33 ± 0.14	39.58 ± 0.06	82.84 ± 0.01
RMCD	**72.89** ± **0.02**	**42.43** ± **0.01**	**80.23** ± **0.01**	**74.18** ± **0.04**	**41.66** ± **0.02**	**77.99** ± **0.04**	**77.78** ± **0.03**	**39.29** ± **0.01**	**83.25** ± **0.01**

Table 2: Comparison of cognitive diagnosis performance on the ASSIST17, ASSIST09, and Junyi datasets. All metrics are reported in %; lower RMSE and higher ACC/AUC are better. Numbers in bold indicate the best performance.

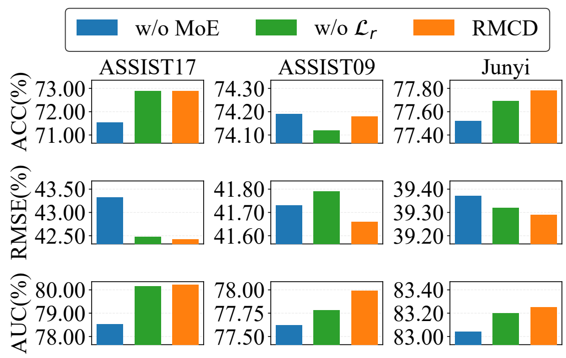

Figure 3: Ablation study of the MoE head and the regularizer $\mathcal{L}_r$.

Model	ACC	RMSE	AUC
n_l=1	71.53 ± 0.08	43.33 ± 0.00	78.52 ± 0.02
n_l=2	71.67 ± 0.06	43.26 ± 0.05	78.51 ± 0.11
n_l=3	**71.81** ± **0.05**	**43.19** ± **0.03**	**78.62** ± **0.07**

Table 3: Ablation study of the graph encoder depth on ASSIST17.

Sub-layer	ACC	RMSE	AUC
N	70.95 ± 0.06	43.77 ± 0.02	77.41 ± 0.06
R	70.91 ± 0.09	43.80 ± 0.02	77.37 ± 0.04
E	71.79 ± 0.04	43.18 ± 0.01	78.69 ± 0.02
N + R	70.87 ± 0.05	43.77 ± 0.02	77.41 ± 0.04
N + E	72.01 ± 0.08	43.06 ± 0.08	79.21 ± 0.12
E + R	72.63 ± 0.03	42.59 ± 0.01	79.94 ± 0.03
N + E + R	**72.89** ± **0.02**	**42.43** ± **0.01**	**80.23** ± **0.01**

Table 4: Ablation study of the sub-layer roles on ASSIST17.

ASSIST09 and Junyi do not, ACD further relies on an unsupervised affect perception module on datasets without affect labels. These results suggest that the relation-aware graph learning and the MoE-based prediction can improve cognitive diagnosis even without external auxiliary features. In addition, RMCD is also more stable, yielding smaller standard deviations among models.

5.3 Ablation Study

We conduct ablation studies to shed light on the impact of each key design within RMCD. The findings from these ablations create possibilities for further optimization.

MoE Head & Load Balancing. To examine the effects of conditional prediction and training regularization in RMCD, we ablate the MoE-based prediction head and the load-balancing regularizer $\mathcal{L}_r$ (Fig. 3). Both components are beneficial. MoE brings larger gains, while $\mathcal{L}_r$ mainly regularizes training and improves ranking quality (RMSE/AUC). Removing MoE (w/o MoE) consistently degrades performance on all datasets, with the largest drop on ASSIST17, implying that under highly heterogeneous behaviors and complex relations, a single predictor is insufficient, whereas the MoE-based prediction better adapts to diverse samples. Removing $\mathcal{L}_r$ (w/o $\mathcal{L}_r$) has a smaller impact on overall performance but yields consistently mild degradations in RMSE and AUC, suggesting that it stabilizes predicted probabilities rather than directly improving accuracy. On ASSIST09, the w/o MoE variant slightly improves ACC but hurts RMSE/AUC, while remaining competitive with prior models (Tab. 2). This suggests that the MoE head refines response probability distributions, improving ranking stability.

We further analyze how the graph encoder depth affects performance under w/o MoE. Table 3 shows that without MoE, the model benefits from a deeper encoder, indicating that ASSIST17 involves more complex student-exercise-concept relations and a single predictor requires richer representations to capture mastery-difficulty discrepancies. With increased depth, the w/o MoE variant remains competitive with the compared models (Tab. 2), indicating that deeper graph encoding can partially compensate for the lack of conditional prediction. When MoE is used, RMCD achieves strong performance with a shallow encoder by adapting to fine-grained mastery-difficulty discrepancies, supporting the value of MoE in complex cognitive diagnosis scenarios.

Sub-layer Roles. We conduct sub-layer ablations of the relation-aware GNN layer to analyze the contribution of each sub-layer (Tab. 4). For brevity, the three sub-layers are denoted as N, E, and R. The results show that the sub-layers are complementary and play distinct roles. Prediction relies on edge representations, while gating is driven by node representations. Using only N or R yields limited performance due to missing explicit relation modeling and edge information. Using only E enables prediction but remains limited without node feedback for effective gating. Combining sub-layers leads to clear improvements. E+R outperforms N+E, high-

Order	ACC	RMSE	AUC
N $\rightarrow$ E $\rightarrow$ R	**72.89 $\pm$ 0.02**	**42.43 $\pm$ 0.01**	**80.23 $\pm$ 0.01**
N $\rightarrow$ R $\rightarrow$ E	72.69 $\pm$ 0.03	42.54 $\pm$ 0.01	79.93 $\pm$ 0.02
R $\rightarrow$ N $\rightarrow$ E	72.77 $\pm$ 0.06	42.53 $\pm$ 0.02	79.97 $\pm$ 0.03
R $\rightarrow$ E $\rightarrow$ N	72.74 $\pm$ 0.08	42.53 $\pm$ 0.03	80.00 $\pm$ 0.05
E $\rightarrow$ N $\rightarrow$ R	72.62 $\pm$ 0.08	42.59 $\pm$ 0.02	79.95 $\pm$ 0.03
E $\rightarrow$ R $\rightarrow$ N	72.73 $\pm$ 0.07	42.56 $\pm$ 0.01	79.97 $\pm$ 0.01

Table 5: Ablation study of the sub-layer order on ASSIST17.

Gating Input	ACC	RMSE	AUC
$\mathbf{v}^s$	71.62 $\pm$ 0.09	43.27 $\pm$ 0.03	78.60 $\pm$ 0.06
$\mathbf{v}^c$	71.50 $\pm$ 0.04	43.33 $\pm$ 0.01	78.54 $\pm$ 0.02
$\mathbf{v}^q$	71.57 $\pm$ 0.06	43.31 $\pm$ 0.03	78.50 $\pm$ 0.07
$[\mathbf{v}^s; \mathbf{v}^q]$	72.79 $\pm$ 0.02	42.46 $\pm$ 0.02	80.21 $\pm$ 0.02
$[\mathbf{v}^s; \mathbf{v}^c]$	71.65 $\pm$ 0.04	43.22 $\pm$ 0.04	78.68 $\pm$ 0.07
$[\mathbf{v}^q; \mathbf{v}^c]$	71.55 $\pm$ 0.05	43.32 $\pm$ 0.02	78.47 $\pm$ 0.05
$[\mathbf{v}^s; \mathbf{v}^q; \mathbf{v}^c]$	**72.89 $\pm$ 0.02**	**42.43 $\pm$ 0.01**	**80.23 $\pm$ 0.01**

Table 6: Ablation study of the gating input on ASSIST17.

lighting the importance of injecting relation-aware edge information into node representations. Building on this, RMCD progressively integrates N, E, and R, achieving the best performance across all three metrics.

Sub-layer Order. Building on the necessity of combining the three sublayers, we further study how their order affects performance. Table 5 shows that different orders lead to consistent changes in all metrics, indicating that relation learning is order-sensitive. Under our design, node representations are first learned, edge representations are then updated based on relational structures, and nodes are finally refined through node-edge interactions. This progressive scheme leads to smoother and more hierarchical message passing. In contrast, updating edges too early or skipping node refinement after edge updates weakens information propagation and degrades performance. Overall, the order N $\rightarrow$ E $\rightarrow$ R achieves the best and most stable results, supporting our design of aligning representation learning with MoE-based prediction.

Gating Input. We ablate the input of the gating function by feeding different combinations of student, exercise, and concept representations (Tab. 6). The results show that the gating input affects performance. Using a single representation yields limited results, indicating that one type alone is insufficient to combine experts effectively. Using two representations improves performance, with $[\mathbf{v}^s; \mathbf{v}^q]$ outperforming combinations involving concepts, suggesting that student and exercise information provides a stronger gating signal. Adding $\mathbf{v}^c$ brings only marginal but consistent gains, since concept information has been largely encoded by the relation-aware graph encoder and helps stabilize the gating decision.

Hyperparameter. We study hyperparameter sensitivity by varying the graph encoder depth n_l and the number of experts n_e on the three datasets, reporting mean results over three runs. Since ACC and RMSE show similar trends to AUC, we focus on AUC. Figure 4 shows that a shallow encoder already performs strongly and stably, and increasing n_l brings no consistent gains and may slightly hurt, likely due to redundant or noisy signals in the gating input. In contrast, under the w/o MoE setting (Tab. 3), deeper encoders consistently help, suggesting that MoE reduces the reliance on deeper graph encoding. RMCD also works well with few experts, and increasing n_e is not monotonic. In particular, ASSIST17 tends to drop with larger n_e, while ASSIST09 and Junyi are relatively flat, indicating that gains come from effective conditional prediction rather than many experts.

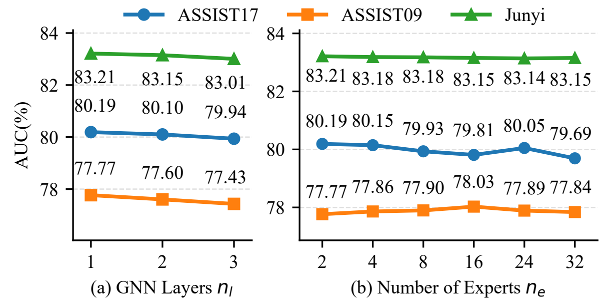

Figure 4: Ablation study of the key hyperparameters n_l and n_e.

Model	Baselines		RMCD (n_e)				
	RCD	SCD	2	4	8	16	32
Time(s)	963.04	3,060.12	24.87	23.06	24.39	24.20	26.46
Params(M)	2.752	2.676	10.126	10.126	10.127	10.129	10.133

Table 7: Efficiency comparison between baselines and RMCD with different numbers of experts on ASSIST09.

Efficiency. To analyze the effect of the MoE head on model complexity and efficiency, we vary the number of experts n_e while keeping all other settings fixed, and report results together with RCD and SCD in Tab. 7. For a fair comparison, RMCD, RCD, and SCD use the same hidden dimension of 123. RMCD trains much faster per epoch, averaged over three consecutive epochs, than both baselines, although it has more parameters due to the use of edge features. Increasing n_e from 2 to 32 only marginally increases parameters and slightly increases per-epoch time, indicating limited overhead. This is because MoE is only applied at the prediction stage. Overall, RMCD achieves a favorable balance between efficiency and performance.

6 Conclusion

In this paper, we propose RMCD, a cognitive diagnosis model that unifies relation-aware graph learning with Mixture-of-Experts prediction. RMCD builds a student-exercise-concept relational graph and jointly learns node and edge representations, where relation-strength vectors support edge updating, node refinement, and concept-level mastery-difficulty modeling. The MoE head models concept-level mastery-difficulty discrepancies with conditional expert combination. Experiments on three benchmarks show consistent gains over state-of-the-art methods. RMCD shows encouraging results but still has limitations. It builds student-concept edges with a heuristic rule and maps edge representations to relation strengths with a fixed squashing function, which may overlook richer dependency cues. In future work, we plan to improve relation construction and introduce stronger constraints for more interpretable relation strengths.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities (Nos. SWU-KR22032, SWU-XDJH202303, and SWU222001) and the Innovation and Entrepreneurship Education Teaching Reform Research Project of Southwest University (No. 26SCJY29).

References

[Anderson *et al.*, 2014] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. Engaging with massive online courses. In *Proc. of WWW*, pages 687–698, 2014.

[Burns *et al.*, 2014] Hugh Burns, Carol A Luckhardt, James W Parlett, and Carol L Redfield. *Intelligent tutoring systems: Evolutions in design*. Psychology Press, 2014.

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

[De La Torre, 2009] Jimmy De La Torre. DINA model and parameter estimation: A didactic. *J. Educ. Behav. Stat.*, 34(1):115–130, 2009.

[Eigen *et al.*, 2013] David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. Learning factored representations in a deep mixture of experts. *CoRR*, abs/1312.4314, 2013.

[Embretson and Reise, 2013] Susan E Embretson and Steven P Reise. *Item response theory for psychologists*. Psychology Press, 2013.

[Fedus *et al.*, 2022] William Fedus, Barret Zoph, and Noam Shazeer. Switch Transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, (120):1–39, 2022.

[Feng *et al.*, 2009] Mingyu Feng, Neil Heffernan, and Kenneth Koedinger. Addressing the assessment challenge with an online system that tutors as it assesses. *User Model. User-Adapt. Interact.*, 19(3):243–266, 2009.

[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 *Proc. of SIGIR*, pages 501–510, 2021.

[Jacobs *et al.*, 1991] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural Comput.*, 3(1):79–87, 1991.

[Kingma, 2014] Diederik P Kingma. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[Kipf, 2016] TN Kipf. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016.

[Kuh *et al.*, 2011] George D Kuh, Jillian Kinzie, Jennifer A Buckley, Brian K Bridges, and John C Hayek. *Piecing together the student success puzzle: Research, propositions, and recommendations: ASHE higher education report*. John Wiley & Sons, 2011.

[Li *et al.*, 2024] Chao Li, Zijie Guo, Kun He, et al. Long-range meta-path search on large-scale heterogeneous graphs. In *Proc. of NeurIPS*, pages 44240–44268, 2024.

[Li *et al.*, 2025] Quan Li, Yun Tian, Xiyuan Wang, Laixin Xie, Dandan Lin, Lingling Yi, and Xiaojuan Ma. MetapathVis: Inspecting the effect of metapath in heterogeneous network embedding via visual analytics. *Comput. Graph. Forum*, 44(1), 2025.

[Lin *et al.*, 2026] Bin Lin, Zhenyu Tang, Yang Ye, Jinfa Huang, Junwu Zhang, Yatian Pang, Peng Jin, Munan Ning, Jiebo Luo, and Li Yuan. MoE-LLaVA: Mixture of experts for large vision-language models. *IEEE Trans. Multimedia*, pages 1–14, 2026.

[Liu *et al.*, 2019] Qi Liu, Shiwei Tong, Chuanren Liu, Hongke Zhao, Enhong Chen, Haiping Ma, and Shijin Wang. Exploiting cognitive structure for adaptive learning. In *Proc. of SIGKDD*, pages 627–635, 2019.

[Masoudnia and Ebrahimpour, 2014] Saeed Masoudnia and Reza Ebrahimpour. Mixture of experts: A literature survey. *Artif. Intell. Rev.*, 42(2):275–293, 2014.

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *Proc. of NeurIPS*, pages 8024–8035, 2019.

[Reckase, 2009] Mark D. Reckase. *Multidimensional Item Response Theory*. Springer, New York, 2009.

[Schlichtkrull *et al.*, 2018] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *Proc. of ESWC*, pages 593–607, 2018.

[Shao *et al.*, 2025] Pengyang Shao, Yonghui Yang, Chen Gao, Lei Chen, Kun Zhang, Chenyi Zhuang, Le Wu, Yong Li, and Meng Wang. Exploring heterogeneity and uncertainty for graph-based cognitive diagnosis models in intelligent education. In *Proc. of SIGKDD*, pages 1233–1243, 2025.

[Shazeer *et al.*, 2017] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *CoRR*, abs/1701.06538, 2017.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *Proc. of ICLR*, 2018.

[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 *Proc. of WWW*, 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 *Proc. of AAAI*, pages 6153–6161, 2020.

[Wang *et al.*, 2023a] Haotao Wang, Ziyu Jiang, Yuning You, Yan Han, Gaowen Liu, Jayanth Srinivasa, Ramana Kompella, Zhangyang Wang, et al. Graph mixture of experts: Learning on large-scale graphs with explicit diversity modeling. In *Proc. of NeurIPS*, pages 50825–50837, 2023.

[Wang *et al.*, 2023b] Shanshan Wang, Zhen Zeng, Xun Yang, and Xingyi Zhang. Self-supervised graph learning for long-tailed cognitive diagnosis. In *Proc. of AAAI*, pages 110–118, 2023.

[Wang *et al.*, 2024] Shanshan Wang, Zhen Zeng, Xun Yang, Ke Xu, and Xingyi Zhang. Boosting neural cognitive diagnosis with student's affective state modeling. In *Proc. of AAAI*, pages 620–627, 2024.

[Wu *et al.*, 2020] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.*, 32(1):4–24, 2020.

[Zeng *et al.*, 2023] Hanqing Zeng, Hanjia Lyu, Diyi Hu, Yinglong Xia, and Jiebo Luo. Mixture of weak & strong experts on graphs. *CoRR*, abs/2311.05185, 2023.

[Zhou *et al.*, 2020] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, pages 57–81, 2020.

[Zhou *et al.*, 2025] Hao Zhou, Zhijun Wang, Shujian Huang, Xin Huang, Xue Han, Junlan Feng, Chao Deng, Weihua Luo, and Jiajun Chen. MoE-LPR: Multilingual extension of large language models through mixture-of-experts with language priors routing. In *Proc. of AAAI*, pages 26092–26100, 2025.