CAAI Transactions on Intelligence Technology

WILEY

ORIGINAL RESEARCH OPEN ACCESS

H2CD: Semantic-Enhanced Heterogeneous Hypergraph Network With Large Language Model for Cognitive Diagnosis

Youheng Bai | Zitao Liu | Mingliang Hou | Teng Guo | Weiqi Luo

Guangdong Institute of Smart Education, Jinan University, Guangzhou, China

Correspondence: Zitao Liu (liuzitao@jnu.edu.cn)

Received: 4 January 2026 | **Revised:** 17 April 2026 | **Accepted:** 27 May 2026

Keywords: AI for education | cognitive diagnosis | hypergraph learning | large language model | smart education

ABSTRACT

Cognitive diagnosis aims to infer learners' knowledge states from their exercise responses, enabling personalised education at scale. Existing methods represent exercises solely by coarse-grained knowledge component annotations, overlooking semantic content and step-level cognitive processes. This causes diagnostic confounding, in which exercises requiring fundamentally different reasoning pathways become indistinguishable in learned representations. Such confounding degrades diagnostic accuracy by conflating distinct learning deficiencies. To address this, we propose H2CD, a semantic-enhanced heterogeneous hypergraph framework that unifies LLM-based cognitive decomposition with structural learning. First, we leverage a large language model to decompose exercises into step-by-step solution paths and extract fine-grained step-level KCs, capturing procedural knowledge beyond predefined annotations. Second, we construct a heterogeneous hypergraph with two types of hyperedges: knowledge hyperedges that connect exercises to their associated multi-granular KCs, preserving compositional knowledge structures, and interaction hyperedges that capture personalised student-exercise relationships. Third, a two-stage hypergraph neural network performs knowledge structure propagation on static exercise-KC relationships and interaction propagation on dynamic student-exercise patterns. Finally, a KC-aware prediction layer projects representations to a learnable diagnostic axis, enabling interpretable mastery assessment at individual KC granularity. Extensive experiments on three real-world educational datasets demonstrate that H2CD outperforms state-of-the-art methods in both AUC and accuracy.

1 | Introduction

With the digital transformation of education, large-scale MOOC platforms such as *Coursera* and *edX* have become central infrastructures of modern education, offering learning opportunities to hundreds of millions of users worldwide [1, 2]. As these platforms scale, a persistent technical challenge emerges: how to accurately assess each learner's knowledge state from diverse learning behaviours [3–5]. Cognitive diagnosis (CD) has emerged as a prominent learner modelling task that leverages extensive interaction logs to infer learners' mastery of knowledge components (KCs)[1] [6, 7]. These diagnostics support adaptive services, from personalised exercise recommendation to learning path planning, thereby enabling personalised education at scale [8–12].

Traditional CDMs, such as item response theory (IRT) and DINA, provide a probabilistic foundation for modelling learner interactions and yield interpretable results [6, 13–15]. However, their hand-crafted interaction functions limit their capacity to

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2026 The Author(s). *CAAI Transactions on Intelligence Technology* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology and Chongqing University of Technology.

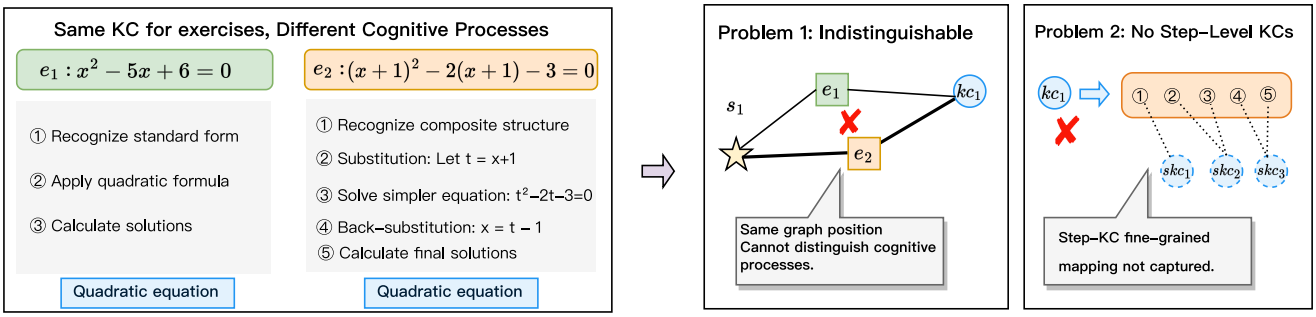


FIGURE 1 | Diagnostic confounding in existing methods. Left: Exercises with different cognitive complexity become indistinguishable when mapped to the same KC. Right: Step-level skills are not captured in current Q-matrix annotations.

capture complex learning patterns in large-scale data. Recent deep learning advances address this by learning representations directly from interaction logs. Models like NeuralCD and KaNCD employ neural networks to capture non-linear relations among learners, exercises, and KCs [7, 16]. More recently, graph-based approaches have modelled the tripartite structure of students, exercises, and KCs, using message passing to exploit high-order dependencies [17, 18]. Despite these structural advances, current methods treat exercises as atomic entities identified solely by their associated KCs, overlooking the rich semantic content and step-by-step cognitive processes embedded within each problem.

To illustrate this limitation concretely, consider two quadratic equation problems shown in Figure 1. Exercise e_1 $(x^2 - 5x + 6 = 0)$ is a standard-form equation solvable in three direct steps: recognise the standard form, apply a direct solution method, and calculate. Exercise e_2 $((x+1)^2 - 2(x+1) - 3 = 0)$, despite appearing similar and sharing the same KC annotation, demands a five-step reasoning chain: recognising the composite structure, introducing a substitution $(t = x + 1)$, solving the simplified equation, back-substituting, and calculating final solutions. A student who masters basic formula application but struggles with substitution strategies would succeed on e_1 but fail to solve e_2. But existing CDMs operate on exercise IDs and KC labels alone, seeing only that both e_1 and e_2 map to 'Quadratic Equation', and thus cannot distinguish between them. Such models would fail to identify that the student lacks substitution skills rather than basic equation-solving ability [19–21]. We term this phenomenon diagnostic confounding. It arises when two exercises share identical KC annotations yet require distinct step-level cognitive processes, namely different ordered sequences of reasoning steps and associated skills. A model that relies solely on KC-level representations maps such exercises to nearly indistinguishable embeddings, failing to differentiate the mastery states revealed by divergent response patterns.

This diagnostic confounding stems from two interrelated limitations in current approaches. The first is semantic indistinguishability in graph-based representations. As shown in the left panel of Figure 1, both e_1 and e_2 are mapped to the same KC kc_1 and thus occupy structurally equivalent positions in the graph. Message passing aggregates information along these identical paths, causing semantically distinct exercises to converge towards similar embeddings regardless of their underlying cognitive complexity. The second limitation is the absence of step-level cognitive mapping. Current Q-matrix annotations directly link exercises to coarse-grained KCs, with no representation of the intermediate reasoning steps. As illustrated in the right panel of Figure 1, the five steps in e_2, each requiring distinct skills such as *pattern recognition* (skc_1), *substitution* (skc_2), and *algebraic manipulation* (skc_3), are collapsed into a single exercise-level annotation. This prevents the model from pinpointing where in the reasoning chain a student struggles: failure at step 2 (substitution) versus step 4 (back-substitution) produces identical diagnostic signals. Beyond these two limitations, the underlying representation poses a structural challenge: each solution step may require multiple KCs and each KC appears across steps of different exercises. Capturing these multi-way dependencies demands representations beyond pairwise graph edges.

Resolving diagnostic confounding requires addressing both limitations: distinguishing semantically similar exercises despite shared KC annotations, and capturing step-level cognitive processes absent from current representations. To this end, we propose H2CD, a framework that integrates LLM-based semantic understanding with heterogeneous hypergraph learning. H2CD first leverages LLMs to decompose exercises into step-by-step solution paths and extract fine-grained step-level KCs, explicitly modelling the cognitive processes that differentiate e_1 from e_2. These multi-granular representations are then embedded into a heterogeneous hypergraph, where distinct hyperedge types capture exercise-KC knowledge structures and student-exercise interaction patterns. Through hypergraph message passing, H2CD learns unified embeddings that encode both semantic richness and structural relationships. These representations feed into a KC-aware prediction layer that projects student and exercise embeddings onto a learnable diagnostic axis, enabling direct comparison of mastery levels against exercise requirements at individual KC granularity. Extensive experiments on three real-world datasets demonstrate substantial improvements in diagnostic accuracy over existing methods. Our main contributions are:

1. We introduce an LLM-based cognitive decomposition approach that extracts step-by-step solution paths and associated KCs, enabling the capture of fine-grained cognitive processes rather than coarse exercise-level annotations.
2. We propose a heterogeneous hypergraph network that naturally models multi-way dependencies between students, exercises, and multi-granular KCs, capturing higher-order

knowledge structures and personalised learning patterns beyond traditional pairwise graphs.

3. Extensive experiments on three real-world datasets demonstrate that H2CD substantially outperforms state-of-the-art methods in diagnostic accuracy while maintaining interpretability.

2 | Related Work

2.1 | Cognitive Diagnosis Models

Traditional CDMs, such as IRT, MIRT, and DINA, explain student-exercise responses via hand-designed interaction functions [6, 13]. While interpretable, these parametric forms constrain expressiveness and struggle with complex, nonlinear learning patterns [22]. To address this, deep-learning-based CDMs learn the interaction function from data: NeuralCDM adopts multi-layer perceptrons to model student-KC interactions [7], and KaNCD further captures implicit associations among KCs [16], improving generalisation and interpretability. More broadly, neural network architectures have demonstrated strong predictive capabilities across diverse domains [23, 24].

However, these methods primarily diagnose individual students and under-utilise the structured relations among students, exercises, and KCs. Graph neural networks (GNNs) bring structured inductive bias to cognitive diagnosis. RCD constructs a heterogeneous graph over students, exercises, and KCs, and performs message passing to capture high-order dependencies [17, 25]. Such structured modelling paradigms have also proven effective in other prediction tasks requiring feature importance analysis [26]. Student-centric designs (e.g., ICDM) accelerate inference, while ORCDF mitigates over-smoothing in student mastery by introducing a response graph with edge types for correct/incorrect outcomes, thereby improving representation discrimination [27]. Despite these advances, two limitations remain: (1) items are often treated as atomic nodes, ignoring rich semantics and problem-solving processes; (2) pairwise edges make it difficult to encode collaborative dependencies among multiple KCs.

Recent work has demonstrated the effectiveness of hypergraph-structured knowledge representations for capturing complex multi-way relationships [28–30]. HyperCDM is the first to apply hypergraphs in cognitive diagnosis, using hyperedges to capture group-wise influences among students and KCs. But it still faces two issues [29]. First, the Q-matrix constructed using two types of hyperedges is extremely sparse in large-scale data, resulting in limited connectivity and underutilisation of high-order signals. Second, the structure is learned solely from interaction data without leveraging semantic information from exercise content, leaving substantial information untapped.

2.2 | Large Language Models in Education

The advent of Large Language Models (LLMs) has opened new possibilities for educational applications. LLMs have demonstrated strong capabilities in intelligent tutoring [31, 32], automated content generation [33], and educational assessment [34]. Their ability to understand and generate natural language enables sophisticated interactions with educational materials, from explaining complex concepts to providing personalised feedback [35]. Recent work has begun exploring LLMs for understanding exercise semantics and educational content analysis [36]. Studies have shown that LLMs can effectively parse problem statements, identify mathematical concepts, and generate step-by-step solutions [37, 38]. In automated grading and assessment, LLMs have been applied to evaluate open-ended responses and provide explanatory feedback [39, 40]. However, these applications typically focus on content generation or assessment rather than learner modelling.

KCD represents an early attempt, using LLMs to extract exercise and student features and generate embeddings for cold-start diagnostic models [41]. While this improves feature quality, it treats LLM outputs as static embeddings without leveraging the rich cognitive structures that LLMs can extract from problem-solving processes. Moreover, existing approaches fail to integrate LLM-derived semantic representations with structured relational models like graphs or hypergraphs [42, 43]. ChatKBQA explores combining fine-tuned LLMs with structured knowledge retrieval, but such integration remains unexplored in cognitive diagnosis [44]. This separation prevents models from simultaneously exploiting both the semantic richness of exercise content and the structural dependencies among students, exercises, and KCs. Our work addresses this gap through heterogeneous hypergraph integration.

3 | The H2CD Framework

Problem definition: In learning systems, let $\mathcal{S} = \{s_1, ..., s_i, ..., s_M\}$, $\mathcal{E} = \{e_1, ..., e_j, ..., e_N\}$, and $\mathcal{K} = \{k_1, ..., k_l, ..., k_K\}$ denote the sets of students, exercises, and KCs, respectively. Student interaction records are represented as triplets $\mathcal{I} = \{(s_i, e_j, r_{ij}) | s_i \in \mathcal{S}, e_j \in \mathcal{E}, r_{ij} \in \{0, 1\}\}$, where r_{ij} indicates whether student s_i answered exercise e_j correctly. The Q-matrix $\mathbf{Q} \in \{0, 1\}^{N \times K}$ encodes the mapping between exercises and KCs, where $Q_{jl} = 1$ if exercise e_j involves the l-th KC. Given $\mathcal{I}$ and $\mathbf{Q}$, CD aims to estimate students' knowledge mastery vectors $\theta_i \in \mathbb{R}^K$ through modelling their responses to predict performance on unseen exercises.

Overview: As shown in Figure 2, the H2CD framework consists of four components. We first leverage an LLM to decompose exercises into multi-granular semantic embeddings: exercise content with solution steps, and fine-grained step-level knowledge components. These feed into a heterogeneous hypergraph with two hyperedge types: exercise hyperedges linking exercises to KCs and step-KCs via attention, and student-exercise hyperedges modelling student-exercise interactions. A multi-layer hypergraph neural network then propagates information through two-stage mechanisms for static knowledge structures and dynamic interactions. Finally, a KC-aware prediction layer maps representations onto an ability-difficulty scale for interpretable diagnosis.

3.1 | Semantic-Enhanced Knowledge Representation via LLM

Different from existing CDMs that treat exercises as atomic entities or extract only content-level features, we leverage an

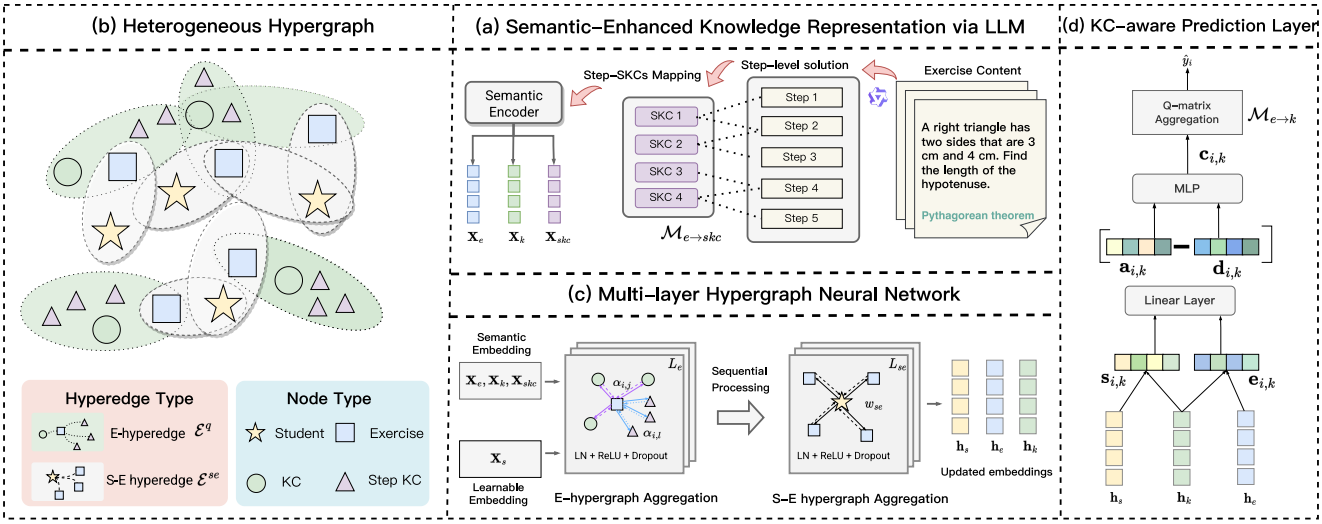


FIGURE 2 | Overview of the H2CD framework. (a) LLM decomposes exercises into step-level solutions and extracts step-KCs. (b) Heterogeneous hypergraph with E-hyperedges (exercise-KC structures) and S-E hyperedges (student-exercise interactions). (c) Multi-layer hypergraph neural network aggregates information through both hyperedge types. (d) KC-aware prediction layer computes response predictions at KC granularity.

LLM to perform deep cognitive decomposition of exercise content, extracting multi-faceted semantic representations that preserve the fine-grained cognitive structures inherent in problem-solving processes.

For each exercise, we employ a two-stage prompting strategy with the Qwen3-235B-A22B model[2] to extract comprehensive cognitive information. The first stage generates step-by-step solution paths for the correct answer following proper mathematical reasoning. The second stage analyzes these solutions to extract step-level knowledge components and create step-to-KC mappings. The two-stage prompts are detailed in Appendix A.

This two-stage approach ensures comprehensive cognitive profiling: Stage 1 captures structural knowledge through solution paths, while Stage 2 extracts fine-grained step-level KCs, creating a rich representation that distinguishes exercises requiring different cognitive processes despite their semantic similarity.

From the LLM-generated decomposition, we construct semantic representations at multiple granularities. At the node level, we generate three types of embeddings h_* using a pre-trained sentence transformer[3] $\phi : \mathcal{T} \to \mathbb{R}^d$:

Exercise Embeddings $(\mathbf{h}_e \in \mathbb{R}^d)$: For each exercise, we combine semantic information with the cognitive processes required to correctly answer the exercise:

$$\mathbf{h}_e = \frac{1}{2}(\mathbf{E}_{\text{stem}} + \mathbf{E}_{\text{steps}}), \tag{1}$$

where $\mathbf{E}_{\text{stem}} = \phi(\text{concat}(q, o_1, ..., o_n))$ encodes the question stem and all answer options. Each component is wrapped with `[START]` and `[END]` markers and the resulting string is fed as a single sequence to the sentence transformer. The equal weighting balances content with solution steps, ensuring both knowledge and procedural structure contribute equally to exercise representation.

KC Embeddings $(\mathbf{h}_{kc} \in \mathbb{R}^{N_{kc} \times d})$: Each pre-annotated exercise-level KC is encoded as $\mathbf{h}^i_{kc} = \phi(kc_i)$, preserving the semantic meaning of educational standards.

Step-KC Embeddings $(\mathbf{h}_{skc} \in \mathbb{R}^{N_{skc} \times d})$: Fine-grained step-level KCs are similarly encoded as $\mathbf{h}^j_{skc} = \phi(skc_j)$, capturing atomic skills required at each solution step.

Unlike pre-annotated KCs with a fixed vocabulary, LLM-generated step-KCs are free-form text that vary across exercises. We normalise step-KC text through lowercasing, stripping whitespace, and removing punctuation, then deduplicate via exact string matching. Semantically similar but textually distinct step-KCs are retained as separate nodes, as the attention-weighted E-hyperedge construction (Section 3.2) naturally assigns similar weights to them based on embedding similarity, achieving soft deduplication without premature information loss.

Additionally, we preserve the structural mappings through Q-matrices $\mathcal{M}_{e\to kc}$ and $\mathcal{M}_{e\to skc}$. Step-level KC extraction analyzes each solution step to identify required atomic skills, creating detailed step-to-KC mappings that capture procedural knowledge granularity. This fine-grained decomposition enables the model to distinguish exercises that may appear similar but require different cognitive processes.

3.2 | Heterogeneous Hypergraph Construction

Traditional graph-based CDMs rely on pairwise edges, which fail to capture the multi-way dependencies inherent in learning. A student's success on an exercise depends on jointly mastering the full set of required KCs and the step-level skills implicated by the solution procedure. Likewise, an exercise coherently ties together multiple KCs and step-KCs, rather than forming independent binary links. To preserve these higher-order, step-aware relationships and avoid information

loss from pairwise decomposition, we construct a heterogeneous hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

The heterogeneous nodes $\mathcal{V}$ include students $\mathcal{V}^s$, exercises $\mathcal{V}^e$, knowledge components $\mathcal{V}^{kc}$, and SKCs $\mathcal{V}^{skc}$. Node features are initialised from the semantic representations constructed in Section 3.1. $\mathbf{X}_s \in \mathbb{R}^{N_s \times d}$ are learnable student embeddings, while $\mathbf{X}_e \in \mathbb{R}^{N_e \times d}$, $\mathbf{X}_{kc} \in \mathbb{R}^{N_{kc} \times d}$, and $\mathbf{X}_{skc} \in \mathbb{R}^{N_{skc} \times d}$ are the projected semantic embeddings. Here $N_{kc} = K$ is the number of pre-annotated KCs and N_{skc} is the number of unique step-level KCs. We design two types of hyperedges to capture the natural groupings in CD:

1. E-hyperedge $\mathcal{E}^e$: Connects each exercise node to all its associated KC and step-KC nodes simultaneously, preserving the fact that these KCs work together to define the exercise's cognitive requirements. The connections are determined by the Q-matrices $\mathcal{M}_{e\to kc}$ and $\mathcal{M}_{e\to skc}$ and weighted by attention mechanisms.
2. S-E hyperedge $\mathcal{E}^{se}$: Connects student-exercise pairs during interactions, capturing how a student's knowledge state relates to the exercise's entire requirement set. For each batch of B interactions $\{(s_i, e_i)\}_{i=1}^{B}$, we create $H_{se} \in \mathbb{R}^{N \times B}$ where column i connects student s_i and exercise e_i.

Let $N = N_s + N_e + N_{kc} + N_{skc}$ denote the total number of nodes. For E-hyperedges, we construct the incidence matrix $H_e \in \mathbb{R}^{N \times |\mathcal{E}^e|}$ where each column represents a hyperedge connecting an exercise to its associated KCs. The structure is defined with attention-weighted connections:

$$H_e[i,j] = \begin{cases} 1 & \text{if node } i \text{ is the exercise node of hyperedge } j \\ \alpha_{i,j} & \text{if node } i \text{ is a KC/step-KC node in hyperedge } j \\ 0 & \text{otherwise} \end{cases}, \quad (2)$$

where the attention weights $\alpha_{i,j}$ are computed using the initial semantic embeddings:

$$\alpha_{i,j} = \frac{\exp\left(\tau^{-1} \cdot \mathbf{h}_{e_j}^{(0)T} \mathbf{W}_a \mathbf{h}_i^{(0)}\right)}{\sum_{k \in \mathcal{K}_j} \exp\left(\tau^{-1} \cdot \mathbf{h}_{e_j}^{(0)T} \mathbf{W}_a \mathbf{h}_k^{(0)}\right)}. \quad (3)$$

Here, e_j denotes the exercise node associated with hyperedge j, $\mathcal{K}_j$ denotes all KC/step-KC nodes in hyperedge j, τ is a learnable temperature parameter controlling attention sharpness, and $\mathbf{W}_a \in \mathbb{R}^{d \times d}$ is a learnable weight matrix for attention scoring.

This complementary design allows the model to capture both universal cognitive structures through static E-hyperedges and personalised learning dynamics through dynamic S-E hyperedges, providing a comprehensive foundation for multi-layer propagation in the subsequent hypergraph neural network.

3.3 | Multi-Layer Hypergraph Neural Network

To effectively propagate information through the heterogeneous hypergraph, we design a multi-layer neural network with two distinct propagation mechanisms: static E-hypergraph convolution for knowledge structure learning and dynamic S-E hypergraph convolution for personalised interaction modelling.

Let $\mathbf{X} = [\mathbf{X}_s; \mathbf{X}_e; \mathbf{X}_{kc}; \mathbf{X}_{skc}] \in \mathbb{R}^{N \times d}$ denote the concatenated node features, where $N = N_s + N_e + N_{kc} + N_{skc}$ is the total number of nodes and d is the feature dimension. Here, $\mathbf{X}_s \in \mathbb{R}^{N_s \times d}$ are learnable student embeddings, while $\mathbf{X}_e$, $\mathbf{X}_{kc}$, and $\mathbf{X}_{skc}$ are the semantic embeddings from Section 3.1, projected to dimension d:

$$\mathbf{X}_e = \text{Proj}_e(\mathbf{h}_e), \quad \mathbf{X}_{kc} = \text{Proj}_{kc}(\mathbf{h}_{kc}), \quad \mathbf{X}_{skc} = \text{Proj}_{skc}(\mathbf{h}_{skc}). \quad (4)$$

The E-hypergraph captures static exercise-knowledge relationships that remain invariant across students. The propagation for L_q layers is defined as:

$$\mathbf{X}^{(l+1)} = \mathbf{X}^{(l)} + \alpha_e \cdot \text{Drop}\big(\text{ReLU}\big(\text{HyperConv}\big(\text{LN}_e^{(l)}(\mathbf{X}^{(l)}), H_e, w_e\big)\big)\big), \quad (5)$$

where α_e is a learnable message scaling factor, and the hypergraph convolution operator is:

$$\text{HyperConv}(\mathbf{X}, H, w) = D_v^{-1/2} H W_e D_e^{-1} H^T D_v^{-1/2} \mathbf{X}. \quad (6)$$

Here, H_e is the attention-weighted incidence matrix defined in Section 3.2, $D_v = \text{diag}\left(\sum_j H_{ij}\right)$ and $D_e = \text{diag}\left(\sum_i H_{ij}\right)$ are the node and edge degree matrices respectively, and $W_e = \text{diag}(w_e)$ with edge weights:

$$w_e = \text{softplus}(\mathbf{W}_{\text{edge}}) + \epsilon, \ \mathbf{W}_{\text{edge}} \in R^{|\mathcal{E}^e|}, \quad (7)$$

where $\mathbf{W}_{\text{edge}}$ is a learnable parameter vector and ϵ ensures numerical stability.

After L_q layers of static E-hypergraph propagation, we apply L_{se} layers of dynamic S-E propagation. Unlike the static E-hypergraph, S-E hyperedges require dynamic edge weights that adapt to evolving node representations. At each layer l, we recompute edge weights using current student and exercise embeddings:

$$w_{se}^{(l)} = \text{softplus}\left(\left[\mathbf{h}_s^{(l)} \oplus \mathbf{h}_e^{(l)}\right] \mathbf{W}_{se}^T\right) + \epsilon, \quad (8)$$

where $\mathbf{h}_s^{(l)}, \mathbf{h}_e^{(l)} \in \mathbb{R}^{B \times d}$ are the current student and exercise embeddings for the batch, and $\mathbf{W}_{se} \in \mathbb{R}^{1 \times 2d}$ is a learnable weight matrix. The propagation follows:

$$\mathbf{X}^{(l+1)} = \mathbf{X}^{(l)} + \alpha_{se} \cdot \text{Drop}\big(\text{ReLU}\big(\text{HyperConv}\big(\text{LN}_{se}^{(l)}(\mathbf{X}^{(l)}), H_{se}, w_{se}^{(l)}\big)\big)\big), \quad (9)$$

where α_{se} is a learnable scaling factor and $H_{se} \in \mathbb{R}^{N \times B}$ is the S-E incidence matrix. This dynamic reweighting enables the model to progressively refine personalised cognitive patterns as node representations evolve through layers.

This layer-wise recomputation follows the same paradigm as attention-based message passing [45, 46]. Training stability is maintained by the residual connections, layer normalisation, and learnable scaling factor already present in Equation (9).

The combination of static knowledge structure learning through E-hypergraph and dynamic interaction modelling through S-E

hypergraph produces node embeddings that preserve both semantic content and structural patterns.

3.4 | KC-Aware Prediction Layer

After multi-layer propagation on the heterogeneous hypergraph, node representations encode both semantic and structural information. Inspired by IRT theory, where a student succeeds on an exercise if their ability on each required KC exceeds the corresponding difficulty, we compare students and exercises in a KC-aware manner and then aggregate the confidence across multiple KCs.

For each student and exercise interaction (s_i, e_i) in a batch and each associated KC k, we contextualise the student and exercise embeddings with the KC embedding $\mathbf{h}_k$:

$$\mathbf{s}_{i,k} = \tanh\left(\mathbf{W}_{sk}\left[\mathbf{h}_{s_i} \oplus \mathbf{h}_k\right]\right), \quad \mathbf{e}_{i,k} = \tanh\left(\mathbf{W}_{ek}\left[\mathbf{h}_{e_i} \oplus \mathbf{h}_k\right]\right), \tag{10}$$

where $\mathbf{W}_{sk}, \mathbf{W}_{ek} \in \mathbb{R}^{d \times 2d}$ and $\oplus$ denotes concatenation. These contextualised vectors are then projected onto a *learnable one-dimensional diagnostic axis* $\mathbf{p} \in \mathbb{R}^d$ to yield ability and difficulty scores:

$$a_{i,k} = \mathbf{p}^\top \mathbf{s}_{i,k}, \quad d_{i,k} = \mathbf{p}^\top \mathbf{e}_{i,k}, \quad c_{i,k} = \mathrm{MLP}\left(a_{i,k} - d_{i,k}\right), \tag{11}$$

where $\mathbf{p}$ is a learnable diagnostic axis shared across all KCs.

To account for nonlinear mastery thresholds, we map the difference through a small MLP, implemented as a two-layer network with hidden dimension 16 and ReLU activation. This design preserves the intuition of IRT but operates at the KC level.

Let $K_i = \{k | Q_{e_i,k} = 1\}$ denote the set of KCs required by exercise e_i according to the Q-matrix $\mathbf{Q} \in \{0,1\}^{N_e \times K}$. We aggregate the per-KC scores with masked averaging and apply a sigmoid:

$$\hat{y}_i = \sigma\left(\frac{1}{|K_i|} \sum_{k \in K_i} c_{i,k}\right). \tag{12}$$

The equal weighting follows the conjunctive assumption in classical CDMs such as DINA [13], where mastery of all required KCs is equally necessary. The nonlinear MLP mapping (Equation (11)) before averaging ensures that the aggregation is not strictly compensatory; extreme deficiency on any single KC produces a low score that cannot be fully offset by strengths on others.

3.5 | Training and Optimization

To train H2CD, we optimise the model parameters using binary cross-entropy loss with standard regularisation techniques (normalisation and residual connections).

For a batch of student-exercise interactions $\{(s_i, e_i, y_i)\}_{i=1}^B$ where $y_i \in \{0,1\}$ indicates whether student s_i correctly answered exercise e_i, the loss function is:

$$\mathcal{L} = -\frac{1}{B} \sum_{i=1}^{B} \left[y_i \log \hat{y}_i + (1 - y_i) \log\left(1 - \hat{y}_i\right)\right], \tag{13}$$

where $\hat{y}_i$ is the predicted probability from the KC-aware prediction layer in Section 3.4.

4 | Experiments

4.1 | Experimental Setup

Datasets and preprocessing: Our datasets are collected from a large online learning platform that provides multiple-choice exercises to support students' homework. Each record contains an anonymised student ID, exercise ID, timestamp, and response data (student response and answer options), along with textual content (exercise text and associated KCs). We selected thinking and mathematics domains to construct three datasets, namely THINK, MATH_G4-5, and MATH_G7.

Please note that our method requires complete exercise text to enable LLM-based cognitive decomposition. Existing public datasets such as ASSIST and Math contain only KC annotations and response logs without full exercise content, making them incompatible with semantic-enhanced approaches. In contrast, our datasets provide detailed exercise text and answer options necessary for step-level KC extraction.

To ensure sufficient data for cognitive diagnosis, we excluded students with fewer than 15 response records. For students who answered the same exercise multiple times, only the first response was retained for static cognitive diagnosis [7, 17]. Table 1 summarises the statistics of the preprocessed datasets. We perform an 80%/20% random split on each student's response sequences for training/validation and testing respectively [7]. All models are evaluated using 5-fold cross-validation to ensure the reliability and generalisation capability of the results.

Baselines: We compared H2CD with 10 state-of-the-art baselines, which are described below:

1. IRT: A classical psychometric model using logistic functions to model response probability based on student ability and item difficulty [6].
2. MIRT: Extends IRT to model multiple latent abilities simultaneously using multidimensional logistic functions [47].

TABLE 1 | Dataset statistics. *Avg. KCs* and *Avg. Qs* denote the number of KCs/exercises per exercise/KC.

	THINK	MATH_G4-5	MATH_G7
# Students	534	962	701
# Questions	8027	20,460	17,944
# KCs	844	1483	931
# Responses	24,866	50,936	30,418
# Step KCs	6989	26,793	20,104
Avg. KCs	1.63	1.69	1.29
Avg. Qs	15.53	23.26	24.84
Correct rate	71.80%	66.68%	66.45%

3. DINA: A rule-based cognitive diagnostic model assuming binary mastery states for each knowledge component [13].
4. NeuralCDM: Uses neural networks with monotonicity constraints to learn complex non-linear student-exercise interactions [7].
5. KaNCD: Extends NeuralCDM by modelling implicit knowledge relations through multiple simple latent interaction representations [16].
6. RCD: A relation-driven graph neural network framework using multi-level attention to model hierarchical student-exercise-concept relations [17].
7. ORCDF: Addresses the oversmoothing problem via response-aware graph convolution and noise-aware training [27].
8. HyperCDM: A hypergraph-based framework that captures homogeneous influence among students through separate hypergraphs for students, exercises, and KCs, using momentum hypergraph convolution networks to mitigate oversmoothing [29].
9. ICDM: An inductive CDM that enables fast mastery inference in large-scale web platforms for new students through neighbour aggregation [48].
10. KCD: Uses LLMs to generate textual diagnoses of students and exercises from collaborative information and response logs, then aligns these semantic representations with behavioural representations from CDMs through a cognitive level alignment module. We use the NCD-Gen implementation [41].

Evaluation metrics and implementation details: To evaluate model performance, we use three metrics following prior work [7]: accuracy (ACC), area under the curve (AUC), and root mean square error (RMSE). Higher ACC and AUC values indicate better performance, whereas lower RMSE values are preferred.

For hyperparameter configuration, we set the unified feature dimension d from {64, 128, 256}, which is shared across all node types (student, exercise, KC, and step-KC embeddings). The number of E-hypergraph layers and S-E hypergraph layers are selected from {1, 2, 3}. Dropout is tuned within {0.05, 0.1, 0.2, 0.3}. The temperature parameter τ for attention-weighted hyperedge construction is selected from {0.5, 1.0, 2.0}. We use the Adam optimiser with learning rates in {5e-5, 1e-4, 3e-4, 5e-4, 1e-3} and batch sizes from {64, 128, 256}. Random seeds are set to {42, 3407} for reproducibility.

For LLM-based cognitive decomposition (Section 3.1), we employ the open-source Qwen3-235B-A22B model (Apache 2.0 license)[4] with temperature = 0 for deterministic outputs. This decomposition is a one-time offline preprocessing step, enabling full reproducibility of downstream experiments without requiring LLM inference.

For model training, 5-fold cross-validation is performed on the training/validation set using Bayesian optimization for hyperparameter tuning and early stopping. Models with the best hyperparameters are then evaluated on the 20% test set. Experiments are implemented in PyTorch 2.1.0 and executed on NVIDIA RTX 4090 GPUs, with Weights & Biases[5] (WandB) used for hyperparameter search and tracking.

4.2 | Overall Performance

We conducted a comprehensive comparison of H2CD against 10 baseline methods on three real-world datasets. The results in Table 2 demonstrate H2CD's superiority across all metrics, with statistically significant improvements ($p < 0.05$) over the strongest baselines. This stems from three synergistic factors:

First, multi-granular semantic representation via LLM decomposition: H2CD leverages LLMs to extract step-by-step solution paths and fine-grained step-level KCs, capturing cognitive structures that distinguish superficially similar exercises. This addresses the fundamental limitation of traditional Q-matrices, which annotate exercises at a single granularity and cannot reveal internal problem-solving processes. The multi-granular representation enables fine-grained diagnosis of specific cognitive deficits rather than coarse exercise-level assessments.

Second, higher-order structural modelling via heterogeneous hypergraph: Unlike pairwise graph CDMs that decompose multi-way dependencies into binary relationships, H2CD's hypergraph naturally preserves collaborative structures where exercises connect to multiple KCs and step-level KCs simultaneously. This higher-order modelling, combined with an attention mechanisms over step-level skills, captures information about which KCs are critical, which simple graph structures tend to collapse.

Third, KC-aware prediction with learnable diagnostic axis: The projection-based prediction layer enables direct comparison of student mastery against exercise requirements at individual KC granularity. By contextualising representations with specific KCs and aggregating ability-difficulty scores across all required components, H2CD achieves interpretable diagnosis aligned with educational theory.

When comparing to baseline categories, the advantages become clear. Traditional CDMs (DINA, IRT, MIRT) lack expressiveness for complex cognitive patterns. Neural CDMs (NeuralCDM, KaNCD) improve generalisation but operate on exercise-level annotations. Graph-based methods (RCD, HyperCDM, ORCDF, ICDM) capture structural patterns but treat exercises as atomic nodes, missing semantic content. Notably, H2CD's advantage over HyperCDM (e.g., MATH_G7: AUC 0.7341 vs. 0.6699) stems from the heterogeneous hypergraph design with multi-granular KC nodes. Even KCD, which integrates LLMs, focuses on semantic-behavioural alignment without the compositional structure that hypergraphs provide.

The improvements quantify H2CD's efficacy: on THINK, 1.13% in ACC and 0.57% in AUC over the best baseline; on MATH_G4-5, 0.39% improvement in ACC; on MATH_G7, 1.41% in AUC and 2.23% in ACC. These consistent gains across datasets with varying characteristics demonstrate robustness and suggest that uniting LLM semantic understanding, heterogeneous hypergraph modelling, and KC-aware prediction improves cognitive diagnosis effectiveness.

4.3 | Ablation Study

To validate the contribution of key architectural components, we conducted ablation studies by systematically removing core

TABLE 2 | Comparison across datasets.

Model	AUC↑	ACC↑	RMSE↓
THINK			
DINA	0.5402 ± 0.0006	0.4918 ± 0.0180	0.5528 ± 0.0378
IRT	0.5697 ± 0.0006	0.6856 ± 0.0054	0.4720 ± 0.0030
MIRT	0.6372 ± 0.0044	0.6687 ± 0.0057	0.4850 ± 0.0091
NeuralCDM	0.6980 ± 0.0015	0.7191 ± 0.0020	0.4370 ± 0.0010
KaNCD	0.7238 ± 0.0028	0.7237 ± 0.0035	0.4307 ± 0.0052
RCD	0.7270 ± 0.0089	0.7215 ± 0.0084	0.4355 ± 0.0094
HyperCDM	0.6862 ± 0.0052	0.6991 ± 0.0078	0.4657 ± 0.0039
ORCDF	0.7066 ± 0.0031	0.7215 ± 0.0027	0.4312 ± 0.0007
ICDM	0.7169 ± 0.0028	0.7204 ± 0.0012	0.4321 ± 0.0026
KCD	0.6970 ± 0.0014	0.7176 ± 0.0028	0.4351 ± 0.0007
H2CD	**0.7327 ± 0.0010**[a]	**0.7350 ± 0.0014**[a]	**0.4254 ± 0.0012**[a]
MATH_G4-5			
DINA	0.5462 ± 0.0026	0.5142 ± 0.0022	0.5243 ± 0.0007
IRT	0.5687 ± 0.0093	0.6446 ± 0.0129	0.4947 ± 0.0090
MIRT	0.6317 ± 0.0073	0.6432 ± 0.0067	0.4935 ± 0.0112
NeuralCDM	0.6905 ± 0.0017	0.6832 ± 0.0027	0.4540 ± 0.0060
KaNCD	0.7064 ± 0.0025	0.6966 ± 0.0024	0.4459 ± 0.0008
RCD	0.6734 ± 0.0172	0.6816 ± 0.0083	0.4713 ± 0.0289
HyperCDM	0.6657 ± 0.0023	0.6643 ± 0.0033	0.4758 ± 0.0187
ORCDF	0.7131 ± 0.0029	0.6994 ± 0.0021	0.4483 ± 0.0026
ICDM	0.6988 ± 0.0019	0.6926 ± 0.0002	0.4473 ± 0.0005
KCD	0.6888 ± 0.0028	0.6847 ± 0.0030	0.4515 ± 0.0031
H2CD	**0.7161 ± 0.0003**[a]	**0.7033 ± 0.0022**[a]	**0.4407 ± 0.0006**[a]
MATH_G7			
DINA	0.5535 ± 0.0033	0.5479 ± 0.0023	0.5144 ± 0.0012
IRT	0.5689 ± 0.0099	0.6365 ± 0.0135	0.5054 ± 0.0122
MIRT	0.6324 ± 0.0085	0.6397 ± 0.0087	0.5037 ± 0.0153
NeuralCDM	0.7017 ± 0.0033	0.6874 ± 0.0009	0.4606 ± 0.0065
KaNCD	0.7200 ± 0.0068	0.6980 ± 0.0062	0.4487 ± 0.0080
RCD	0.7212 ± 0.0026	0.6957 ± 0.0037	0.4565 ± 0.0022
HyperCDM	0.6699 ± 0.0023	0.6619 ± 0.0022	0.4647 ± 0.0038
ORCDF	0.7058 ± 0.0007	0.6930 ± 0.0020	0.4476 ± 0.0003
ICDM	0.7071 ± 0.0020	0.6895 ± 0.0040	0.4527 ± 0.0041
KCD	0.7023 ± 0.0032	0.6893 ± 0.0019	0.4485 ± 0.0008
H2CD	**0.7341 ± 0.0007**[a]	**0.7193 ± 0.0015**[a]	**0.4361 ± 0.0007**[a]

[a]Denotes that our results are significantly better than the best baselines ($p < 0.05$). The best results are in bold and the second-best results are underlined.

modules: (1) *w/o SE* removes LLM-based semantic enhancement, using only raw learnable embeddings; (2) *w/o E* removes the E-hypergraph structure; (3) *w/o S-E* removes S-E hypergraph structure, relying only on static E-hypergraph; (4) *w/o PL* removes the KC-aware prediction layer. Results in Figure 3 reveal the importance of each component:

LLM-based semantic enhancement is the most critical component: Removing semantic enhancement (w/o SE) causes the most severe performance degradation across datasets. On THINK, AUC drops by 0.98 percentage points (0.7327 → 0.7229), representing the largest decline among all ablations. Similar patterns emerge on MATH_G4-5 (0.78% AUC drop) and MATH_G7 (1.63% AUC drop). This demonstrates that LLM-generated solution steps and step-level KCs capture fine-grained cognitive structures essential for distinguishing exercises with similar surface features but fundamentally different reasoning processes. Without this deep semantic decomposition,

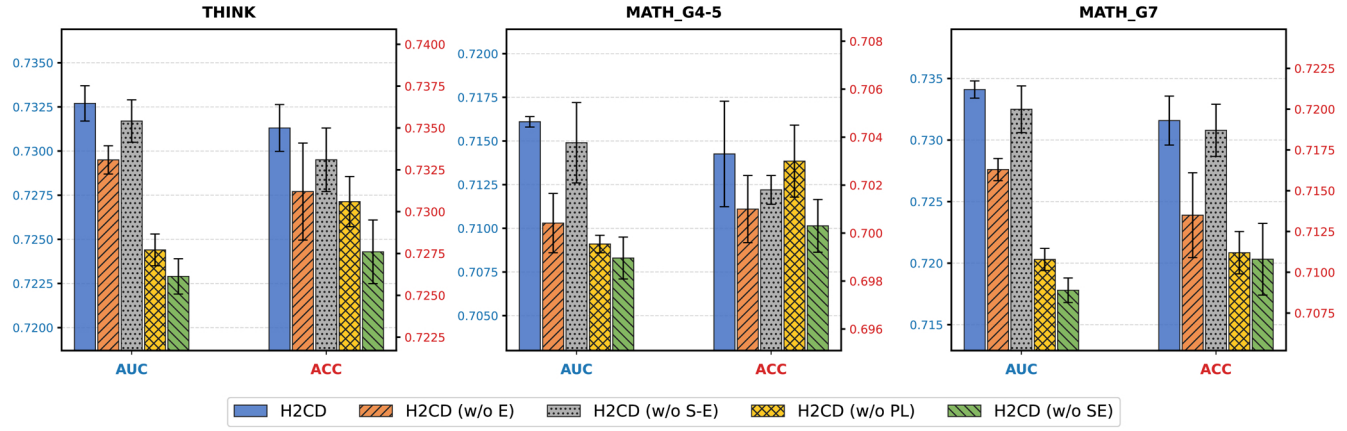


FIGURE 3 | Ablation study on three datasets. Error bars indicate standard deviation across 5 folds.

the model loses its ability to represent procedural knowledge at the granularity required for accurate diagnosis.

KC-aware prediction layer enables interpretable diagnosis: The w/o PL variant shows substantial drops (0.83% in AUC on THINK, 1.38% on MATH_G7), confirming that contextualising student and exercise representations with specific KCs, then projecting onto a learnable diagnostic axis, is crucial for both accuracy and interpretability. Without this design, H2CD cannot pinpoint which knowledge components drive success or failure.

E-hypergraph propagation captures static knowledge structures: The w/o E variant exhibits notable degradation (0.65% on MATH_G7), demonstrating the importance of the first-stage E-hypergraph propagation for learning static exercise-KC relationships. Without this stage, the model loses the ability to capture multi-way dependencies where exercises simultaneously connect to multiple KCs at different granularities, compromising its understanding of compositional knowledge structures.

S-E hypergraph propagation enhances interaction modelling: The w/o S-E variant shows relatively smaller performance drops (0.10% AUC on THINK, 0.16% on MATH_G7), primarily because the learnable student embeddings can still capture personalised characteristics even without explicit interaction modelling through hypergraph propagation. However, the consistent improvements with S-E hypergraph demonstrate that explicitly modelling dynamic interactions between students and exercises through the second-stage propagation does provide additional benefits.

In summary, semantic enhancement emerges as the cornerstone of H2CD's success, followed by the KC-aware prediction layer for interpretability, E-hypergraph propagation for capturing static knowledge structures, and S-E hypergraph propagation for personalised dynamics. The two-stage hypergraph design proves effective, with the static stage providing foundational knowledge modelling and the dynamic stage adding personalisation.

Coarse KCs and step-KCs provide complementary signals: The ablation above establishes semantic enhancement as the most impactful module, but leaves open whether the two KC granularities within it are complementary or redundant. To address this, we analyse the structural overlap between coarse KCs and step-KCs directly from the data, independent of model behaviour. For each pair of exercises sharing at least one coarse KC, we compute the Jaccard similarity of their step-KC sets: $J(e_i, e_j) = |S_i \cap S_j| / |S_i \cup S_j|$, where S_i denotes the set of step-KCs associated with exercise e_i. A low Jaccard indicates that exercises grouped under the same coarse KC require substantially different procedural skills.

The results are consistent across all three datasets in Figure 4: mean Jaccard ranges from 0.053 to 0.074, and over 93% of same-KC exercise pairs have Jaccard ≤ 0.3. Exercises that share the same coarse KC, and would therefore be indistinguishable in traditional Q-matrix representations, overwhelmingly require different combinations of step-level skills. Coarse KCs provide categorical structure aligned with curriculum standards, enabling cross-exercise regularisation. Step-KCs decompose the procedural requirements within each domain, enabling discrimination among exercises that coarse KCs collapse together. The two granularities are structurally complementary, not redundant.

4.4 | Hyperparameter Sensitivity

To understand H2CD's sensitivity to architectural hyperparameters, we analysed the impact of hypergraph layer depth on three datasets. We varied the number of E-hypergraph layers (L_e) and S-E hypergraph layers (L_{se}) from 0 to 3, with results shown in Figures 5 and 6.

E-hypergraph layers benefit from deeper aggregation: As shown in Figure 5, performance consistently improves as L_e increases across all datasets. On THINK, AUC rises from 0.7290 ($L_e = 0$) to 0.7327 ($L_e = 3$), while ACC increases from 0.7305 to 0.7343. Similar upward trends appear on MATH_G4-5 and MATH_G7. This suggests that multi-layer propagation on E-hypergraph effectively captures higher-order relationships between exercises and multi-granular KCs. Since E-hypergraph models static knowledge structures rather than dynamic interactions, deeper layers can aggregate richer semantic information without significant oversmoothing risks.

S-E hypergraph layers show diminishing returns: In contrast, Figure 6 reveals that S-E hypergraph performance peaks at shallow depths ($L_{se} = 0$ or 1) and then declines. On THINK,

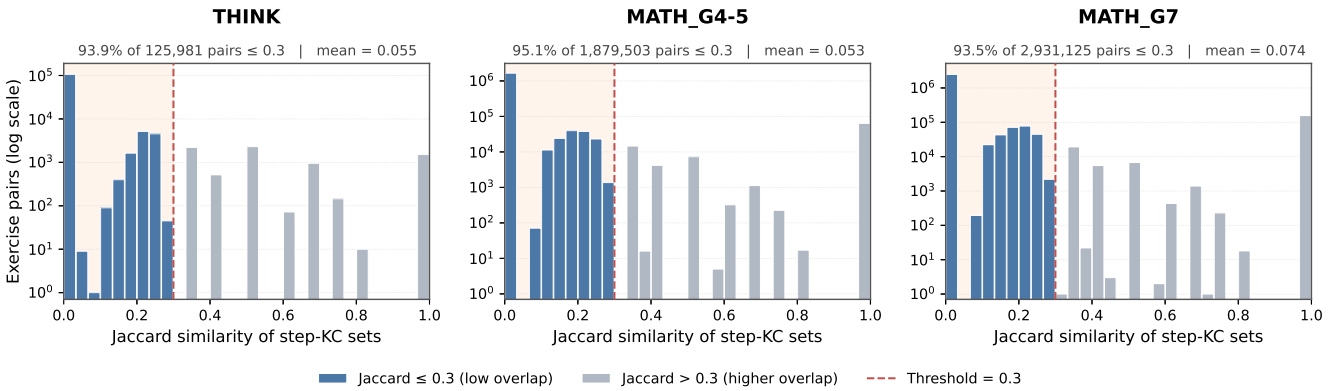


FIGURE 4 | Distribution of step-KC Jaccard similarity among exercise pairs sharing the same coarse KC (log scale). Mean Jaccard: 0.055 (THINK), 0.053 (MATH_G4-5), 0.074 (MATH_G7).

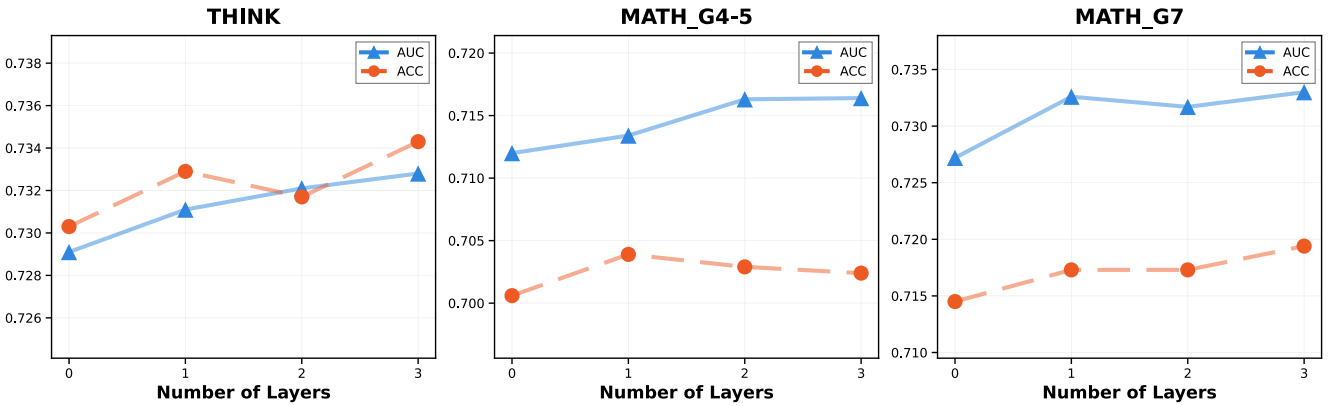


FIGURE 5 | Impact of E-hypergraph layer depth (L_e) on model performance. AUC and ACC improve as L_e increases from 0 to 3 across all datasets, indicating that deeper propagation on static knowledge structures captures richer exercise-KC relationships without oversmoothing.

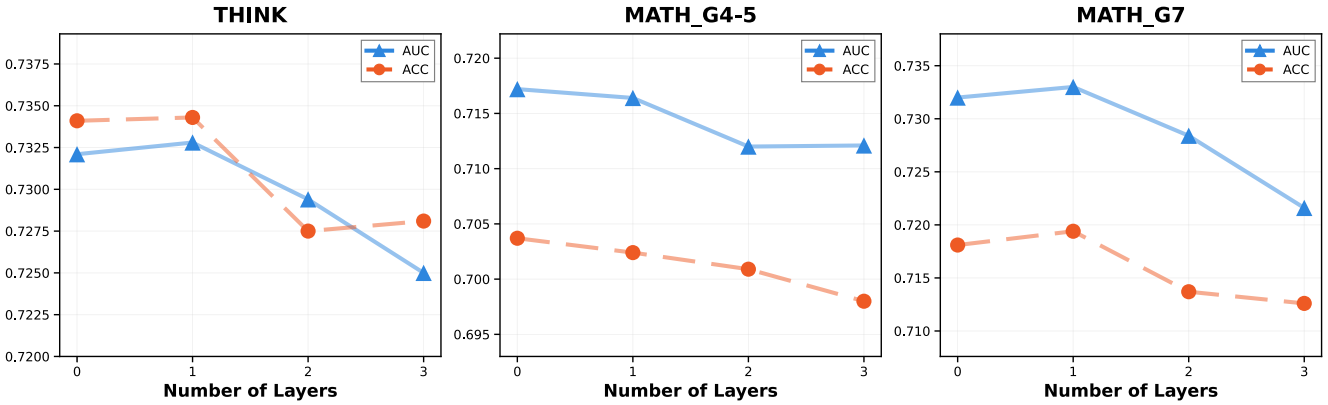


FIGURE 6 | Impact of S-E hypergraph layer depth (L_{se}) on model performance. Performance peaks at shallow depths ($L_{se} = 0$ or 1) and then declines, suggesting that excessive propagation on student-exercise interactions introduces noise from heterogeneous learning behaviours.

AUC drops from 0.7320 ($L_{se} = 0$) to 0.7255 ($L_{se} = 3$). This pattern indicates that while S-E hypergraphs capture personalised student-exercise interactions, excessive propagation may obscure individual learning patterns or introduce noise from heterogeneous student behaviours. The optimal configuration uses minimal or no S-E layers, relying primarily on E-hypergraph for structural learning while S-E hyperedges provide direct personalised signals.

These findings suggest H2CD benefits from asymmetric depth design: deeper E-hypergraph layers ($L_e = 3$) for knowledge structure modelling and shallow S-E layers ($L_{se} = 1$) for personalised interactions. This aligns with the distinct roles of static knowledge propagation versus dynamic behavioural modelling in our framework.

4.5 | Analysis of LLM's Role in H2CD

A natural question about H2CD is how the LLM contributes to diagnostic performance: whether the framework depends critically on LLM output quality, and whether the LLM alone can replace the structural modelling. We address these two concerns

through a noise injection experiment and a direct-prediction experiment.

4.5.1 | *Robustness to LLM Decomposition Noise*

Since H2CD relies on LLM-generated step-level KCs, a natural concern is whether errors in LLM decomposition could degrade diagnostic performance. To evaluate this, we conduct a noise injection experiment that simulates LLM errors by randomly corrupting step-KC mappings at varying noise ratios.

Specifically, for each noise ratio $\rho \in \{0.1, 0.2, 0.3\}$, we randomly select ρ fraction of all step-KC mapping entries and replace each selected step-KC with a randomly sampled step-KC from the global pool, effectively simulating incorrect KC extraction by the LLM. This perturbation disrupts both the semantic content of step-KC nodes and the structural connectivity of E-hyperedges. For efficiency, experiments are conducted on a single fold. Results are shown in Figure 7.

Two key findings emerge from this analysis. First, H2CD exhibits graceful degradation across all datasets. On AUC, the maximum drop from 0% to 30% noise is 0.74 percentage points (THINK), 0.40% (MATH_G4-5), and 0.12% (MATH_G7), indicating that the model is highly resilient to noisy step-KC mappings. Second, performance stabilises quickly after an initial small drop at 10% noise, with minimal further degradation from 10% to 30%. This plateau effect suggests that the attention-weighted E-hyperedge construction (Section 3.2) naturally downweights noisy step-KC connections, as corrupted nodes receive lower attention scores due to reduced semantic similarity with their associated exercises.

Even under the extreme scenario of 30% corrupted mappings, H2CD's AUC on THINK (0.7269) still surpasses all baselines except RCD (0.7270), confirming robustness far beyond realistic LLM error rates.

4.5.2 | *Comparison With Direct LLM Prediction*

A complementary concern is whether H2CD's improvements are primarily driven by the LLM itself rather than the proposed framework. To isolate the contribution of structural modelling, we evaluate Qwen3-235B-A22B as a direct predictor: given a target exercise and the student's contextual information, the LLM directly predicts whether the student will answer correctly. We design three prompt variants with varying context richness: *Population-Only* provides only the target exercise text along with group-level response statistics from the training set; *Student-Only* provides the target student's historical responses on related KCs and an overall learner profile; *Student + Population* combines both information sources. The three prompt variants are provided in Appendix B. Results on the full test sets of all three datasets are summarised in Table 3.

Two observations emerge. First, even under the most informative *Student + Population* condition, Qwen3-235B achieves only 62.5%–66.9% ACC, substantially below H2CD (70.3%–73.5%). This 6%–9% gap confirms that H2CD's gains stem from the hypergraph framework rather than the LLM itself. Second, providing richer context to the LLM does not consistently improve its predictions. Adding population statistics on top of student history (*Student + Population* vs. *Student-Only*) yields no improvement on THINK and slight degradation on both MATH datasets. This suggests that the LLM cannot effectively fuse heterogeneous signals such as student history and group patterns without the structural inductive bias provided by hypergraph modelling. Together with the noise-robustness results, these findings show that H2CD uses the LLM as a semantic decomposer rather than a predictor.

4.6 | Model Complexity Analysis

Table 4 compares H2CD with representative baselines in terms of time and space complexity. The time complexity of H2CD per forward pass is $\mathcal{O}(LNd + B\bar{k}d^2)$, where N is total nodes, L is total layers (typically 1–3), d is feature dimension, B is batch size, and $\bar{k}$ is average KCs per exercise. The model contains $\mathcal{O}(N_s d + d^2)$ parameters: $N_s d$ from student embeddings and d^2 from trainable layers. With typical settings ($d = 128$, $L = 1$), the learnable parameters comprise: student embeddings ($N_s \times d$), three projection layers ($3 \times d_{\text{st}} \times d$, where $d_{\text{st}} = 768$ is the sentence transformer output dimension), attention and prediction weights ($\mathcal{O}(d^2)$). The total ranges from ~300 K ($d = 64$) to ~900 K ($d = 256$) depending on configuration.

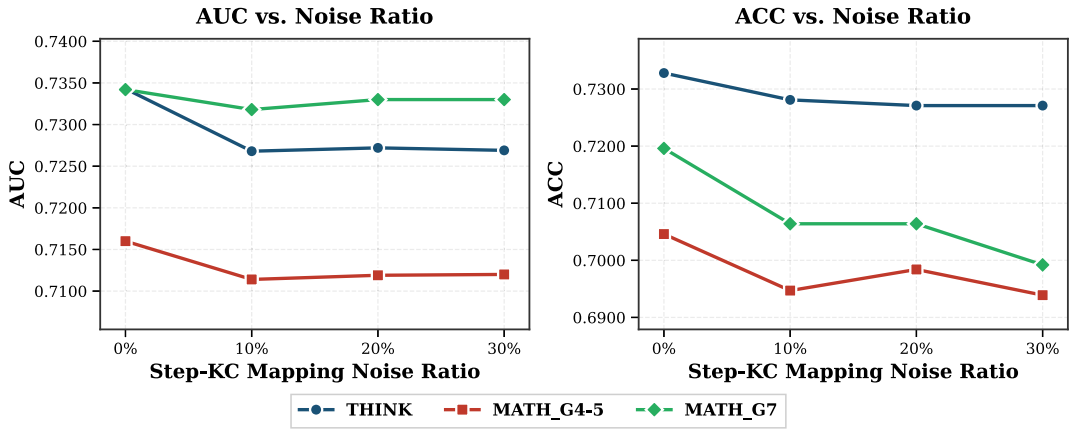


FIGURE 7 | Impact of step-KC mapping noise on model performance. H2CD exhibits graceful degradation: AUC drops less than 1% even at 30% noise across all datasets, demonstrating robustness to LLM decomposition errors.

TABLE 3 | Direct prediction accuracy of Qwen3-235B-A22B under different prompt variants, compared with H2CD.

Method	THINK (n = 4654)	MATH_G4-5 (n = 10,262)	MATH_G7 (n = 6127)
Qwen3-235B-A22B (*Population-Only*)	0.5741	0.5410	0.5739
Qwen3-235B-A22B (*Student-Only*)	0.6685	0.6368	0.6375
Qwen3-235B-A22B (*Student* + *Population*)	0.6661	0.6249	0.6302
H2CD (ours)	**0.7350**	**0.7033**	**0.7193**

Note: Even with the most informative context, the LLM consistently lags behind H2CD by 7–11 percentage points in ACC across all three datasets. The best results are in bold.

TABLE 4 | Complexity comparison with baseline models.

Model	Time complexity	Parameters
IRT	$\mathcal{O}(B)$	$\mathcal{O}(N_s + N_e)$
MIRT	$\mathcal{O}(BK)$	$\mathcal{O}(N_sK + N_eK)$
NeuralCDM	$\mathcal{O}(BKd)$	$\mathcal{O}(N_sK + d^2)$
KaNCD	$\mathcal{O}(BKd)$	$\mathcal{O}((N_s + N_e + K)d + d^2)$
RCD	$\mathcal{O}(\lvert E\rvert K + BK^3)$	$\mathcal{O}((N_s + N_e + K)K + d^2)$
HyperCDM	$\mathcal{O}(L\lvert E\rvert d + BKd)$	$\mathcal{O}((N_s + N_e + K)d + d^2)$
H2CD	$\mathcal{O}(LNd + B\bar{k}d^2)$	$\mathcal{O}(N_sd + d_{st}d + d^2)$

Notably, H2CD does not require learnable embeddings for exercises, KCs, or step-KCs; these are frozen sentence transformer outputs transformed by small projection layers. This replaces the $(N_e + K)d$ terms in baselines like KaNCD and HyperCDM with a fixed $3d_{st}d$ cost, substantially reducing parameters when the dataset is large.

Compared to baselines, H2CD offers two efficiency advantages. First, the prediction term scales with $\bar{k}$ (actual KCs per exercise, typically 3–5) rather than K (total KCs, often 100+), providing substantially better efficiency than RCD's $\mathcal{O}(BK^3)$. Second, despite the d^2 factor from KC-specific contextualisation, the overall complexity remains comparable to HyperCDM while enabling more expressive KC-level representations. This demonstrates that H2CD achieves richer semantic and structural modelling without significant computational overhead. Note that the LLM-based decomposition is a one-time offline preprocessing step, and its cost is not included in Table 4 as it is amortised across all subsequent training and inference.

4.7 | Quality Assurance for LLM-Generated Outputs

We validate two critical outputs from LLM decomposition: exercise solution paths and step-level KC mappings.

For solution path quality, two mathematics education experts independently reviewed 50 randomly sampled exercises, assessing logical coherence, mathematical correctness, and plausibility of reasoning. Inter-rater agreement reached Cohen's $\kappa = 0.87$, with 94% (47/50) of exercises approved.

For step-KC mapping correctness, the same experts evaluated the 50 sampled exercises on two additional criteria: (1) whether each step-KC accurately describes the skill required by its associated step, and (2) whether the step-to-KC assignment is appropriate. Inter-rater agreement reached Cohen's $\kappa = 0.82$, with 88% (44/50) of exercises having all mappings judged correct and the remaining six containing only minor granularity disagreements (e.g., a single step-KC judged overly specific) rather than outright errors.

TABLE 5 | Quality evaluation of LLM-extracted step-level KCs.

Metric	Value
Pre-annotated KC coverage (cosine $>$ 0.75)	91.2%
Pre-annotated KC coverage (cosine $>$ 0.80)	88.4%
Mean semantic similarity	0.84
Step-KC mapping completeness	97.3%
Expert-judged mapping correctness	88% (44/50)

For step-KC quality, we evaluate against pre-annotated KCs in the Q-matrix using three metrics: coverage rate (percentage of pre-annotated KCs captured, cosine > 0.75), semantic similarity (average cosine between pre-annotated and extracted KCs), and mapping completeness (percentage of steps mapped to KCs). The cosine threshold of 0.75 follows standard practice in semantic textual similarity benchmarks, where 0.75 indicates strong paraphrase-level similarity under sentence-transformer embeddings [49]. We also verified robustness to this choice: coverage remains above 88% at the stricter threshold of 0.80. Results are shown in Table 5.

4.8 | Case Study

CD aims to provide interpretable cognitive assessments of how student abilities match exercise difficulties, yet this relationship remains opaque at the individual interaction level. We address this through visualisation on MATH_G7 to reveal whether the KC-aware prediction layer learns meaningful ability-difficulty scales.

We select six students from the MATH_G7 test set across three ability tiers (High/Mid/Low, two per tier) based on their observed accuracy rates on the test set. For each student, we extract 10 questions and compute KC-aware advantage scores following Equation (11), where the difference between projected ability and difficulty captures the student's advantage on each KC. The visualisation employs a diverging colourmap (green for positive advantage, red for negative) with symmetric range centred at zero. Each cell displays the advantage score alongside

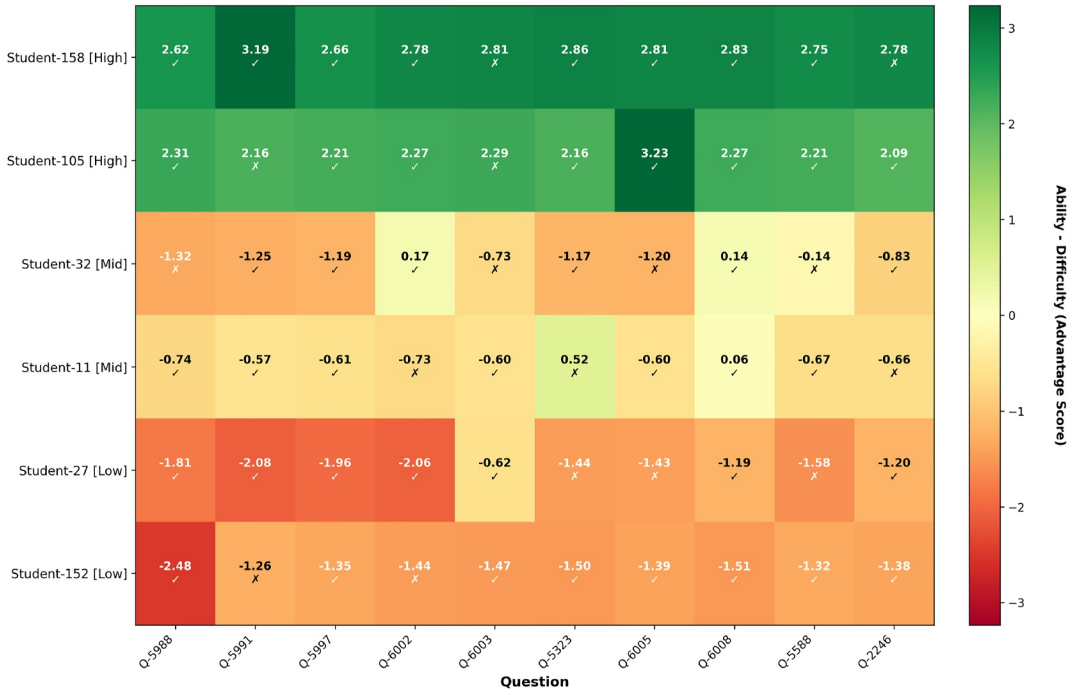


FIGURE 8 | Ability-difficulty matching visualisation for six students on the MATH_G7 dataset. Cell colours represent advantage scores (green: ability exceeds difficulty, red: difficulty exceeds ability). Markers indicate whether the model's prediction matched the true label (✓ correct, ✗ incorrect).

prediction correctness markers (✓ for correct prediction, ✗ for incorrect prediction).

Figure 8 demonstrates three key properties of the learned ability-difficulty scale. First, clear ability stratification emerges: High-tier students exhibit predominantly green cells (+2.09 to +3.23), Low-tier students show red/orange cells (−2.48 to −0.62), and Mid-tier students occupy the transitional zone with mixed colours (−1.32 to +0.52). Second, the model captures within-tier variations—for instance, Student-158 ranges from +2.62 (Q-5988) to +3.19 (Q-5991), enabling fine-grained diagnosis beyond coarse groupings. Third, prediction accuracy strongly correlates with advantage magnitude: cells with extreme advantage values show predominantly ✓ markers (80% for High-tier, 65% for Low-tier), while ✗ markers concentrate in the transitional zone (−0.8 to +0.2) where ability-difficulty balance creates intrinsic uncertainty, reflected in Mid-tier's lower accuracy (75%). The high accuracy in red cells demonstrates the model correctly predicts student struggles when difficulty exceeds ability. Examining Q-5991 vertically reveals consistent difficulty ordering across tiers: advantage scores decrease monotonically from + 3.19 (High) to −2.08 (Low), confirming the model learns coherent relative difficulty rather than fixed absolute values.

This case study validates that the KC-aware prediction layer learns a meaningful ability-difficulty scale with interpretable semantics at the individual interaction level. By preserving IRT theoretical foundations while operating at the fine-grained KC level, the model achieves both accurate predictions and transparent diagnoses, which are essential for adaptive learning systems that require explainable student assessments.

5 | Conclusion

In this paper, we propose H2CD, a semantically enhanced heterogeneous hypergraph framework that incorporates LLMs for cognitive diagnosis. To address the diagnostic confounding problem where exercises with similar annotations require different cognitive processes, we leverage an LLM to extract multi-granular semantic exercise representations through cognitive decomposition. A heterogeneous hypergraph with two types of hyperedges is designed to model multi-way dependencies, and a KC-aware prediction layer enables fine-grained diagnosis at individual KC granularity. Through comprehensive experiments on three real-world datasets, H2CD is verified as effective with significant improvements over state-of-the-art baselines. In the future, we will explore how to fine-tune LLMs to generate cognitive decompositions more aligned with educational standards, investigate cross-domain transfer of step-level KCs to address cold-start problems in new subjects, and combine our semantic-enhanced exercise representations with inductive approaches to enable efficient diagnosis for new students with limited interaction records.

Funding

This work was supported in part by NSFC under Grant 62477025; in part by Beijing Natural Science Foundation under Grant L252010; in part by Key Laboratory of Smart Education of Guangdong Higher Education Institutes, Jinan University (Grant 2022LSYS003); and in part by the Outstanding Innovative Talents Cultivation Funded Programs for Graduate Students of Jinan University (Grant 2025CXY347); and in part by the Guangdong University Student Science and Technology Innovation Cultivation Special Fund under Grant pdjh2026ak027.

Ethics Statement

The authors have nothing to report.

Consent

The authors have nothing to report.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

Data and code are available upon reasonable request.

Endnotes

[1] A KC is a generalization of everyday terms like concept, principle, fact, or skill.

[2] https://huggingface.co/Qwen/Qwen3-235B-A22B-Thinking-2507.

[3] https://huggingface.co/sentence-transformers/all-mpnet-base-v2.

[4] https://huggingface.co/Qwen/Qwen3-235B-A22B-Thinking-2507.

[5] https://wandb.ai/.

References

1. M. Barak and S. Z. Wandering, "A Web-Based Platform for the Creation of Location-Based Interactive Learning Objects," *Computers & Education* 62 (2013): 159–170, https://doi.org/10.1016/j.compedu.2012.10.015.

2. P. Anderson, *What Is Web 2.0?: Ideas, Technologies and Implications for Education*, vol. 1 (JISC Technology and Standards Watch, 2007).

3. S. Y. Chen and X. Liu, "An Integrated Approach for Modeling Learning Patterns of Students in Web-Based Instruction: A Cognitive Style Perspective," *ACM Transactions on Computer-Human Interaction* 15, no. 1 (2008): 1–28, https://doi.org/10.1145/1352782.1352783.

4. Z. Liu, T. Guo, Q. Liang, et al., "Deep Learning Based Knowledge Tracing: A Review, a Tool and Empirical Studies," *IEEE Transactions on Knowledge and Data Engineering* 37, no. 8 (2025): 4512–4536, https://doi.org/10.1109/tkde.2025.3552759.

5. Y. Bai, M. Hou, T. Guo, Z. Liu, and W. Luo, "A Frequency-Aware Mixture of Heterogeneous Experts Framework for Knowledge Tracing," in *Proceedings of the ACM Web Conference 2026*, (2026), 7080–7090.

6. S. E. Embretson and S. P. Reise, *Item Response Theory for Psychologists* (Psychology Press, 2013), https://doi.org/10.4324/9781410605269.

7. F. Wang, Q. Liu, E. Chen, et al., "Neural Cognitive Diagnosis for Intelligent Education Systems," in *Proceedings of the AAAI Conference on Artificial Intelligence, the AAAI Conference on Artificial Intelligence*, 34, no. 4 (2020): 6153–6161, https://doi.org/10.1609/aaai.v34i04.6080.

8. H. Ma, Z. Huang, W. Tang, and X. Zhang, "Exercise Recommendation Based on Cognitive Diagnosis and Neutrosophic Set," in *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, (2022), 1467–1472.

9. H. Du, N. Li, F. Ma, and T. Palaoag, "Personalization Exercise Recommendation Based on Cognitive Diagnosis," in *The 6th International Conference on Computer Science and Application Engineering*, (2022), 1–5.

10. K. Wancham and K. Tangdhanakanond, "Development of Learning Path Map of Work and Energy for High Schoolers by Using Cognitive Diagnostic Assessment," *Eurasia Journal of Mathematics, Science and Technology Education* 19, no. 11 (2023): em2360, https://doi.org/10.29333/ejmste/13812.

11. Y. Bai, Z. Liu, T. Guo, M. Hou, and K. Xiao, "Prerequisite Relation Learning: A Survey and Outlook," *ACM Computing Surveys* 57, no. 11 (2025): 1–28, https://doi.org/10.1145/3733593.

12. Y. Bai, Y. Zhang, K. Xiao, Y. Lou, and K. Sun, "A BERT-Based Approach for Extracting Prerequisite Relations Among Wikipedia Concepts," *Mathematical Problems in Engineering* 2021, no. 1 (2021): 3510402–3510408, https://doi.org/10.1155/2021/3510402.

13. J. De La Torre, "The Generalized DINA Model Framework," *Psychometrika* 76, no. 2 (2011): 179–199, https://doi.org/10.1007/s11336-011-9207-7.

14. W. Ma and J. De La Torre, "GDINA: An R Package for Cognitive Diagnosis Modeling," *Journal of Statistical Software* 93 (2020): 1–26, https://doi.org/10.18637/jss.v093.i14.

15. J. De La Torre, "DINA Model and Parameter Estimation: A Didactic," *Journal of Educational and Behavioral Statistics* 34, no. 1 (2009): 115–130, https://doi.org/10.3102/1076998607309474.

16. F. Wang, Q. Liu, E. Chen, et al., "NeuralCD: A General Framework for Cognitive Diagnosis," *IEEE Transactions on Knowledge and Data Engineering* 35, no. 8 (2022): 8312–8327, https://doi.org/10.1109/tkde.2022.3201037.

17. W. Gao, Q. Liu, Z. Huang, et al., "RCD: Relation Map Driven Cognitive Diagnosis for Intelligent Education Systems," in *The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, (2021), 501–510.

18. S. Yang, M. Chen, Z. Wang, et al., "DisenGCD: A Meta Multigraph-Assisted Disentangled Graph Learning Framework for Cognitive Diagnosis," *Advances in Neural Information Processing Systems* 37 (2024): 91532–91559, https://dl.acm.org/doi/10.5555/3737916.3740821.

19. J. Hu, W. Xia, X. Zhang, et al., "Enhancing Sequential Recommendation via LLM-Based Semantic Embedding Learning," in *Companion the ACM Web Conference*, (2024), 103–111.

20. M. Hong, Y. Xia, Z. Wang, et al., "EAGER-LLM: Enhancing Large Language Models as Recommenders Through Exogenous Behavior-Semantic Integration," in *The ACM on Web Conference*, (2025), 2754–2762.

21. S. Huang, Z. Liu, Q. Liu, J. Chen, and Y. Huang, "Dual-Attentional Time-Aware Fusion Networks for Knowledge Tracing," *Information Fusion* 126 (2026): 103573, https://doi.org/10.1016/j.inffus.2025.103573.

22. Y.-W. Lee and Y. Sawaki, "Cognitive Diagnosis Approaches to Language Assessment: An Overview," *Language Assessment Quarterly* 6, no. 3 (2009): 172–189, https://doi.org/10.1080/15434300902985108.

23. Y. S. Li Hang and Z. Liu, "Identifying At-Risk K-12 Students in Multimodal Online Environments: A Machine Learning Approach," in *Proceedings of the 13th International Conference on Educational Data Mining, EDM 2020, Fully Virtual Conference*, (2020), 113–117, https://doi.org/10.1007/978-981-97-4442-8_9.

24. Z. Liu, G. Xu, T. Liu, et al., "Dolphin: A Spoken Language Proficiency Assessment System for Elementary Education," in *Proceedings of the Web Conference 2020*, (2020), 2641–2647.

25. J. Feng, Y. Chen, F. Li, A. Sarkar, and M. Zhang, "How Powerful Are k-Hop Message Passing Graph Neural Networks," *Advances in Neural Information Processing Systems* 35 (2022): 4776–4790, https://dl.acm.org/doi/10.5555/3600270.3600615.

26. J. Liu and Z. Hou, "Establishment of Second-Hand Sailboats Price Prediction Model Based on Random Forest and Exploration of Influencing Factors," in *2023 IEEE 3rd International Conference on Data Science and Computer Application (ICDSCA)*, (2023), 1337–1342.

27. H. Qian, S. Liu, M. Li, B. Li, Z. Liu, and A. Z. Orcdf, "An Oversmoothing-Resistant Cognitive Diagnosis Framework for Student Learning in Online Education Systems," in *The 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, (2024), 2455–2466.

28. H. Luo, E. Haihong, G. Chen, et al., "HyperGraphRAG: Retrieval-Augmented Generation via Hypergraph-Structured Knowledge Representation," in *Advances in Neural Information Processing Systems*, vol. 38 (2025), 152206–152234.

29. J. Shen, H. Qian, S. Liu, W. Zhang, B. Jiang, and A. Zhou, "Capturing Homogeneous Influence Among Students: Hypergraph Cognitive Diagnosis for Intelligent Education Systems," in *The 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, (2024), 2628–2639.

30. S. Huang, H. Li, Y. Gu, X. Hu, Q. Li, and G. Xu. HyperG, "Hypergraph-Enhanced LLMs for Structured Knowledge," in *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, (2025), 1218–1228.

31. J. Stamper, R. Xiao, and X. Hou, "Enhancing LLM-Based Feedback: Insights From Intelligent Tutoring Systems and the Learning Sciences," in *International Conference on Artificial Intelligence in Education*, (2024), 32–43.

32. T. Wang, Y. Zhan, J. Lian, et al., "LLM-Powered Multi-Agent Framework for Goal-Oriented Learning in Intelligent Tutoring System," in *Companion the ACM on Web Conference*, (2025), 510–519.

33. Y. Zheng, X. Li, Y. Huang, et al., "Automatic Lesson Plan Generation via Large Language Models With Self-Critique Prompting," in *International Conference on Artificial Intelligence in Education*, (2024), 163–178.

34. Y. Liu, S. Bhandari, and Z. A. Pardos, "Leveraging LLM Respondents for Item Evaluation: A Psychometric Analysis," *British Journal of Educational Technology* 56, no. 3 (2025): 1028–1052, https://doi.org/10.1111/bjet.13570.

35. Q. Jia, J. Cui, H. Du, et al., "LLM-Generated Feedback in Real Classes and Beyond: Perspectives From Students and Instructors," in *The 17th International Conference on Educational Data Mining*, (2024), 862–867.

36. C. Hankeln, "Challenges in Using ChatGPT for Assessing Conceptual Understanding in Mathematics Education," *Journal of Mathematics Education* 17, no. 1 (2024): 1–15, https://doi.org/10.26711/007577152790171.

37. R. Meissner, A. Pögelt, K. Ihsberner, et al., "LLM-Generated Competence-Based E-Assessment Items for Higher Education Mathematics: Methodology and Evaluation," *Frontiers in Education* 9 (2024): 1427502, https://doi.org/10.3389/feduc.2024.1427502.

38. Z. Shao, S. Yuan, L. Gao, Y. He, D. Yang, and S. Chen, "Unlocking Scientific Concepts: How Effective Are LLM-Generated Analogies for Student Understanding and Classroom Practice?," in *The 2025 CHI Conference on Human Factors in Computing Systems*, (2025), 1–19.

39. D. Armfield, E. Chen, A. Omonkulov, et al., "Avalon: A Human-in-the-Loop LLM Grading System With Instructor Calibration and Student Self-Assessment," in *International Conference on Artificial Intelligence in Education*, (2025), 111–118.

40. O. Fagbohun, N. P. Iduwe, M. Abdullahi, A. Ifaturoti, and O. Nwanna, "Beyond Traditional Assessment: Exploring the Impact of Large Language Models on Grading Practices," *Journal of Artificial Intelligence, Machine Learning and Data Science* 2, no. 1 (2024): 1–8, https://doi.org/10.51219/jaimld/oluwole-fagbohun/19.

41. Z. Dong, J. Chen, and F. Wu, "Knowledge Is Power: Harnessing Large Language Models for Enhanced Cognitive Diagnosis," *AAAI Conference on Artificial Intelligence* 39, no. 1 (2025): 164–172, https://doi.org/10.1609/aaai.v39i1.31992.

42. Y. Feng, C. Yang, X. Hou, et al., "Beyond Graphs: Can Large Language Models Comprehend Hypergraphs?," in *The Thirteenth International Conference on Learning Representations*, (2025), 1–28.

43. Z. Shu, X. Sun, and H. Cheng, "When LLM Meets Hypergraph: A Sociological Analysis on Personality via Online Social Networks," in *The 33rd ACM International Conference on Information and Knowledge Management*, (2024), 2087–2096.

44. H. Luo, E. Haihong, Z. Tang, et al., "Chatkbqa: A Generate-Then-Retrieve Framework for Knowledge Base Question Answering With Fine-Tuned Large Language Models," in *Findings of the Association for Computational Linguistics: ACL 2024*, (2024), 2039–2056.

45. P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph Attention Networks," *arXiv preprint arXiv:1710.10903* (2017).

46. Z. Huang and Z. L. Hcgkt, "Hierarchical Contrastive Graph Knowledge Tracing With Multi-Level Feature Learning," in *International Conference on Artificial Intelligence in Education* (Springer, 2025), 274–288.

47. M. D. Reckase, "Multidimensional Item Response Theory," *Handbook of Statistics* 26 (2006): 607–642, https://doi.org/10.1007/978-0-387-89976-3.

48. S. Liu, J. Shen, H. Qian, and A. Zhou, "Inductive Cognitive Diagnosis for Fast Student Learning in Web-Based Intelligent Education Systems," in *The ACM Web Conference*, (2024), 4260–4271.

49. N. Reimers and I. Gurevych, "Sentence-Bert: Sentence Embeddings Using Siamese Bert-Networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (EMNLP-IJCNLP, 2019), 3982–3992.

Appendix A: LLM-Based Exercise Decomposition Protocol

Experimental Configuration: We employ Qwen3-235B-A22B with temperature = 0, max_tokens = 1000 (Stage 1) and 2000 (Stage 2). All prompts enforce strict JSON output for structural consistency across three datasets.

Stage 1: Exercise Solution Generation

Objective: Generate step-by-step reasoning for all answer options.

System Prompt:

'Generate step-by-step reasoning for ALL options. For the CORRECT option: proper solution steps. For INCORRECT options: plausible but flawed reasoning leading to that answer. Output ONLY valid JSON.'

Input:

- Question text q, Options $\{o_0, o_1, ...\}$
- Correct answer labelled
- Prior KCs: $\{kc_1, ..., kc_m\}$

Output Schema: {

```
'step_by_step_solution_0':
['step1', 'step2', …],
'step_by_step_solution_1':
['step1', 'step2', …],
…
}
```

Stage 2: Knowledge Concept Mapping

Objective: Analyse the correct solution through three phases: KC identification, step-KC mapping, and structured output generation.

System Prompt: 'Analyse the correct solution to: (1) Identify required KCs using Common Core State Standards, with justification; (2) Map solution steps to KCs (many-to-many); (3) Output structured JSON. Output ONLY valid JSON.'

Input:

- Question metadata + all solution paths from Stage 1
- Correct option index identified

Output Schema: {

```
'question_id': '<id>',
'knowledge_concepts_list':
['<KC1>', '<KC2>', …],
'kc_mapping': {
'reasoning': '<justification>',
'step_kc_mapping': '0-0,0-1,1-2,…'
}
}
```

End-to-End Example

Input: 'A number multiplied by 5 equals 40, this number multiplied by 9 is ( ).' Options: A. 72, B. 45, C. 60. Correct: A.

Stage 1 Output (Correct option):

['Let x be the unknown: 5x = 40″, 'Solve: $x = 40 \div 5 = 8$″, 'Calculate 9x: $9 \times 8 = 72$']

Stage 2 Output:

KCs: ['Determine unknown in multiplication equation (3.OA.A.4)', 'Division as unknown-factor problem (3.OA.B.6)', 'Fluently multiply and divide (3.OA.C.7)']

Mapping: '0–0, 0–1, 1–0, 1–1, 1–2, 2–2'

(Step 0 → KC 0,1; Step 1 → KC 0,1,2; Step 2 → KC 2)

Quality Control: Each LLM output is validated against the expected JSON schema; malformed responses trigger automatic retries (up to 3 attempts). If parsing fails after retries, regex-based extraction recovers structured fields from raw text. Exercises with missing solution steps or empty KC mappings after all recovery attempts are excluded (< 1% across all datasets). Temperature = 0 ensures deterministic outputs across runs.

Appendix B: LLM-Based Prediction Prompts

Experimental Configuration: We evaluate three prompt variants for LLM-based prediction: *Population-Only*, *Student-Only*, and *Student + Population*. All prompts require strict JSON output.

Prompt Variant 1: Population-Only

Objective: Predict which option a student is most likely to choose using only question-level population statistics.

System Prompt:

```
'You are a math education expert. You only see
population-level statistics about how stu-
dents answer this question (from a training
cohort), not any individual student's his-
tory. Predict which option a student is most
likely to choose.'
```

Input:

- A population-statistics block for the current question, formatted in natural language:
  - number of students
  - correct rate
  - wrong-option distribution
- Current question block:
  - KC
  - optional difficulty
  - question text
  - options
  - correct answer

Task Prompt:

```
'You do NOT have any individual student's
history. Use only the population statistics
(if any) and the question text to predict
which option a student would most likely
choose.'
```

Output Schema: {

```
'prediction': '<option letter>',
'confidence': {'A': 0.xx, 'B': 0.xx, …},
}
```

Prompt Variant 2: Student-Only

Objective: Predict which option the student will choose using only student-specific response history.

System Prompt:

'You are a math education expert specializing in diagnosing elementary school students' misconceptions. Based on the student's known response records, predict which option the student will choose.'

Input:

- A same-KC history block containing detailed prior records on the current KC
- A student overall profile block summarising all available prior records
- Optionally, when the number of other-KC records is small, a few sample records from other KCs
- Current question block:
  - KC
  - optional difficulty
  - question text
  - options
  - correct answer

Task Prompt:

'Based only on this student's response history (no population statistics), predict which option they will choose for the current question.'

'The student may choose the correct answer or a wrong option. Infer likely misconceptions from their past wrong answers on similar skills.'

Output Schema: {

'prediction': '<option letter>',

'confidence': {'A': 0.xx, 'B': 0.xx, …},

}

Prompt Variant 3: Student + Population

Objective: Predict which option the student will choose by combining student-specific history with population-level statistics.

System Prompt:

'You are a math education expert specializing in diagnosing elementary school students' misconceptions. Based on the student's known response records and the population error patterns, predict which option the student will choose.'

Input:

- A same-KC history block containing detailed prior records on the current KC
- A student overall profile block summarising all available prior records
- Optionally, when the number of other-KC records is small, a few sample records from other KCs
- A population-statistics block for the current question
- Current question block:
  - KC
  - optional difficulty
  - question text
  - options
  - correct answer

Task Prompt:

'Based on the student's response history and the population statistics, predict which option they will choose for the current question.'

'The student may choose the correct answer or any wrong option. Combine individual patterns with population error tendencies.'

Output Schema: {

'prediction': '<option letter>',

'confidence': {'A': 0.xx, 'B': 0.xx, …},

}