MicroC-KT: Modeling Community Effect via Learning Micro-Environment for Evidence-Grounded Explainable Knowledge Tracing

Zhiyi Duan[1], Zixing Shi[1], Bing Jia[1]*, Qi Wang[2]*

[1]Department of Computer Science, Inner Mongolia University, Hohhot, Inner Mongolia
[2]School of Artificial Intelligence, Jilin University, Changchun, Jilin
{duanzy, jiabing}@imu.edu.cn, zixingshi@mail.imu.edu.cn
qiwang@jlu.edu.cn

Abstract

Knowledge Tracing (KT) is essential for tracking students' evolving knowledge states and predicting their future performance. While current graph-based methods focus on exercise-concept relations, they often overlook the inherent group structures among students. Similarly, emerging LLM-based approaches rely on individual histories, lacking the broader context of group references and contrastive evidence. As a result, existing individual-isolation paradigms fail to provide stable predictions and evidence-based explanations. To bridge this gap, we propose *Micro-Community Knowledge Tracing (MicroC-KT)*, a framework that incorporates learning micro-environments to provide social-cognitive anchors for KT. MicroC-KT identifies latent learning communities via hypergraph modeling and generates dual-granular summaries to facilitate community matching and peer retrieval. By extracting contrastive group evidence, the model prompts an LLM to generate both accurate answer predictions and verifiable analysis reports. Experiments on four public datasets demonstrate that MicroC-KT significantly outperforms state-of-the-art baselines in predictive performance while providing more reliable and evidence-based explanations.

1 Introduction

Knowledge Tracing (KT) aims to model students' evolving knowledge states from their historical interaction sequences to predict their performance on subsequent tasks (Corbett and Anderson, 1994). Its significance spans critical educational applications, including personalized practice recommendations, instructional interventions, and learning diagnosis (Duan et al., 2024; Sun et al., 2025). In real educational scenarios, learning is far more than an isolated cognitive accumulation, it also involves frequent feedback-driven regulation and strategy adjustment (Shen et al., 2024). Thus, beyond achieving high predictive accuracy, it is essential for KT systems to offer clear, verifiable explanations that enhance the transparency and practical utility of instructional decisions (Huang et al., 2024; Zhang et al., 2024).

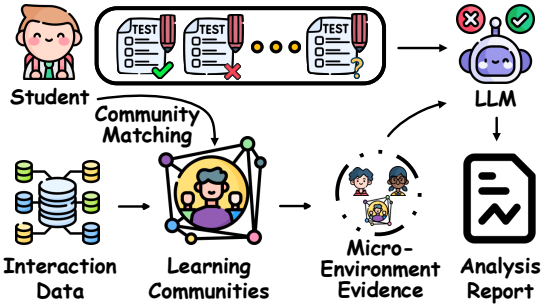

Figure 1: MicroC-KT converts group cognitive structure into a semantic evidence chain for KT prediction and analysis report generation.

Despite the strong predictive performance of deep learning (DL)-based KT methods (e.g., DKT (Piech et al., 2015), AKT (Ghosh et al., 2020)), their black-box nature hinders the generation of interpretable textual outputs, failing to provide intuitive and educationally meaningful diagnostic feedback for teachers and students. Recently, Large Language Model (LLM)-based KT approaches (e.g., EFKT (Li et al., 2025a), CIKT (Li et al., 2025c)) have shown promise in generating natural language explanations. However, these methods often incur substantial training and fine-tuning costs. More importantly, when relying solely on limited individual interaction histories, they are prone to factual hallucinations. Without broader contextual references, the generated content can deviate from a student's true knowledge state, undermining the reliability in high-stakes educational applications.

More importantly, these two lines of work typically build models for a single student in isolation and neglect the community effects that widely exist among students. In real learning processes,

* Corresponding authors.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 25781–25803

students' behaviors and cognitive states are influenced not only by personal history. Rather, they are also closely related to interactions with other members of the learning group, shared cognitive challenges, and group-level knowledge evolution dynamics (Evans et al., 1992; Gao et al., 2025). Ignoring such group-level interactions and influences may prevent the model from capturing important social learning signals (Zheng et al., 2025; Li et al., 2024a). Such an omission not only limits the model's ability to disambiguate sparse individual data but also restricts its overall robustness and generalization across diverse learning environments.

The necessity of capturing these social learning signals is deeply rooted in established learning theories. For example, *Social Constructivism* highlights the role of social interaction and contextual support in cognitive development (Adams, 2006), while the *Knowledge Community* view posits that learners form stable group structures around shared goals and practical norms (Paavola et al., 2004). Inspired by these principles, we propose *Micro-Community Knowledge Tracing (MicroC-KT)*, an explainable KT framework that leverages evidence from learning micro-environments. As shown in Fig. 1, our core idea is to transform implicit group cognitive structures into an explicit semantic evidence chain, enabling LLMs to perform reasoning with group-level context without training.

Specifically, we first construct a learning environment hypergraph that integrates student ability and multi-dimensional performance into a high-dimensional structural space. We then apply hypergraph spectral clustering to identify learning communities, which are treated as observable micro-environments. To bridge the high-dimensional graph space and the semantic space of LLMs, we design a dual-granularity summarization technique that converts structured student and community profiles into textual summaries, forming textual and statistical representation. This representation aligns learners with suitable communities and organizes corresponding micro-environment evidence. Within the matched community, we further retrieve similar peers to construct contrastive peer evidence. Finally, the target student, question, and accumulated micro-environment evidence are fed into an LLM for training-free inference. In our work, training-free strictly denotes the absence of any gradient-based backpropagation; no neural network parameters are updated. This definition follows the standard convention in the machine learning community (Yoon et al., 2024), where methods without gradient-based parameter updates are classified as training-free. **Our major contributions are summarized as follows**:

- We propose MicroC-KT, the first training-free framework modeling the group effect and improves prediction and explanation by leveraging learning micro-environments and community cognitive context.
- We construct a learning environment hypergraph and discover learning communities via hypergraph spectral clustering, providing a structural basis for stable group-level cognitive references.
- Based on structured statistical summaries, we introduce a text summarization technique to form a bi-granular representation, aligning graph representations with the natural language semantic space of LLMs for community matching, peer retrieval, and evidence-grounded analysis report generation.
- Experiments on multiple public educational datasets show that our method outperforms state-of-the-art baselines in both predictive performance and interpretability.

2 Related Work

2.1 Graph-based KT Methods

Building on deep sequential KT models, researchers have further introduced graph modeling to better characterize the higher-order interactions among students, knowledge concepts (KCs), and questions (e.g., GKT (Nakagawa et al., 2019), DyGKT (Cheng et al., 2024), SimQE (Sun et al., 2024), and STHKT (Li et al., 2025d)). More recently, some studies have adopted hypergraphs for finer-grained relational modeling (e.g., HyperKT (Li et al., 2024b), DGEKT (Cui et al., 2024), and HDKT (Hou et al., 2025)). These methods use hyperedges to capture higher-order many-to-many interactions among multiple entities, thereby enriching the structural representations at the question and knowledge concept levels. However, these graph-based methods still mainly model learning from a single student's interaction sequence. Without explicit student-group structures, it is often hard to build a comparable group reference for robust and explainable prediction in practice.

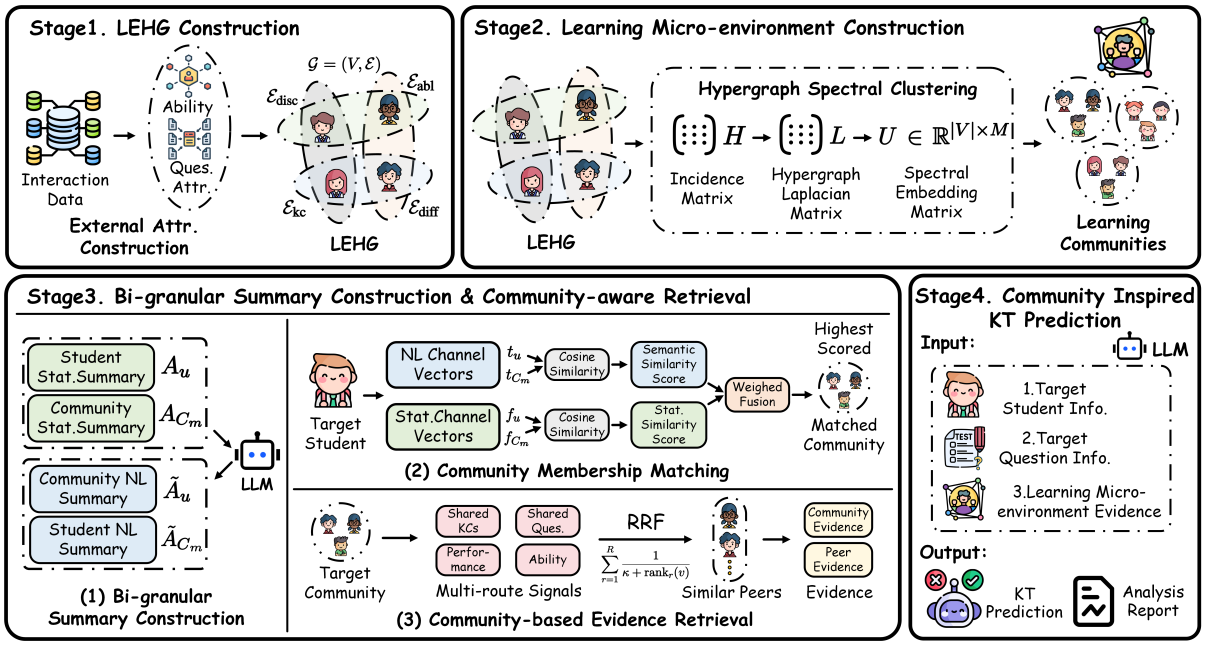

Figure 2: The overall framework of MicroC-KT: (1) Construct a learning environment hypergraph; (2) Perform spectral clustering on the hypergraph to obtain learning communities as observable micro-environments; (3) Build bi-granular summaries to match the target student and retrieve community-based evidence; (4) Feed student and question information and micro-environment evidence into an LLM for KT prediction and analysis report generation.

2.2 LLM-based KT Methods

Leveraging their strong language understanding capabilities, LLMs have been introduced into KT to enhance interpretability. One line of work injects knowledge tracing capabilities into LLM via instruction tuning (e.g., LLM-KT (Wang et al., 2025), CIKT (Li et al., 2025c), and EPFL (Neshaei et al., 2024)), enabling end-to-end generation for prediction and explanation. Another line mainly relies on prompt engineering (e.g., EFKT (Li et al., 2025a) and 2T-KT (Li et al., 2025b)), where structured prompts organize a student's interaction history, questions, and KCs to guide LLM reasoning and explanation generation. However, both lines largely overlook explicit group-level modeling, making it difficult to build comparable group references. Instruction-tuned methods face high training barriers and require substantial data as well as hardware resources, while prompt-based methods may hallucinate when evidence is scarce, undermining reliability and explanation verifiability.

3 Methodology

In this section, we propose MicroC-KT, whose overall framework is illustrated in Fig. 2. The framework is organized into four sequential stages.

3.1 Learning Environment Hypergraph Construction

This module constructs the Learning Environment Hypergraph (LEHG) to explicitly model higher-order relations of student ability and learning performance at the student level. We take the student set V as the only vertex set, and define LEHG as $\mathcal{G} = (V, \mathcal{E})$, where $\mathcal{E} = \mathcal{E}_{\text{abl}} \cup \mathcal{E}_{\text{perf}}$, $\mathcal{E}_{\text{abl}}$ denotes the ability-aware hyperedge set, and $\mathcal{E}_{\text{perf}}$ denotes the performance-aware hyperedge set. The former connects students with similar global abilities into space structures, and the latter organizes students with similar learning performance into higher-order interaction units.

3.1.1 External Attribute Construction

We first employ IRT-2PL (Birnbaum, 1968) model to construct interpretable external attributes for students and questions (e.g., student ability, question difficulty and discrimination). Subsequently, we discretize ability and question-level performance into a finite set of levels, with the ability level $c_u^\theta \in \{\text{low}, \text{mid}, \text{high}\}$ and the performance level $\rho(\cdot) \in \{\text{poor}, \text{avg}, \text{good}\}$, to ensure consistent semantic definitions of subsequent hyperedges. The specific attribute estimation method, discretization rules, and symbol definitions see Appendix A.

3.1.2 Ability-aware Hyperedge Construction

Ability-aware hyperedges provide a stable global group reference. We aggregate students into corresponding higher-order relations based on the ability level c_u^θ, and form the ability-aware hyperedge set $\mathcal{E}_{\text{abl}} = \{e_{\text{low}}^\theta, e_{\text{mid}}^\theta, e_{\text{high}}^\theta\}$. The specific definition and construction details see Appendix B.

3.1.3 Performance-aware Hyperedge Construction

Relying solely on ability-aware hyperedges is insufficient to reveal fine-grained differences in students across specific cognitive dimensions and question-attribute dimensions. Based on the performance level $\rho(\cdot)$, we further construct the performance-aware hyperedge set $\mathcal{E}_{\text{perf}} = \mathcal{E}_{\text{kc}} \cup \mathcal{E}_{\text{diff}} \cup \mathcal{E}_{\text{disc}}$, which correspond to (1) *knowledge-concept performance hyperedges*, (2) *difficulty-stratified performance hyperedges*, (3) *discrimination-stratified performance hyperedges*, respectively. These hyperedges characterize multi-dimensional learning behaviors. The specific construction of the three types of performance-aware hyperedges can be found in Appendix C.

3.2 Learning Micro-environment Construction

We model the group effect by viewing a learning community as an observable learning micro-environment. It aggregates students with similar ability and multi-dimensional learning performance, providing a stable community reference and group-level cognitive context.

To construct learning communities at the student level, we perform hypergraph spectral clustering (Zhou et al., 2006) based on LEHG. Formally, let the student set be V, and denote the incidence matrix of LEHG as $H \in \{0,1\}^{|V|\times|\mathcal{E}|}$, where $H_{u,e} = 1$ indicates student u belongs to hyperedge e. We first compute the hyperedge degree $\delta(e) = \sum_{u\in V} H_{u,e}$ and construct the diagonal matrix $D_e = \text{diag}(\delta(e))$. Meanwhile, we compute the vertex degree $d(u) = \sum_{e\in\mathcal{E}} H_{u,e}$ and construct the diagonal matrix $D_v = \text{diag}(d(u))$. Based on these definitions, we construct the normalized hypergraph Laplacian as:

$$L = I - D_v^{-\frac{1}{2}} H D_e^{-1} H^\top D_v^{-\frac{1}{2}} \tag{1}$$

where $I \in \mathbb{R}^{|V|\times|V|}$ is the identity matrix.

We then take the eigenvectors corresponding to the smallest M eigenvalues of L to form an spectral embedding matrix $U \in \mathbb{R}^{|V|\times M}$, and cluster the row vectors of U to obtain a community partition $\{C_1, \ldots, C_M\}$, where $C_m \subseteq V$, satisfies $\bigcup_{m=1}^{M} C_m = V$ and $C_i \cap C_j = \varnothing\ (i \neq j)$.

The resulting communities characterize the local group structure of students in the joint space of ability and multi-dimensional learning performance. We use this community partition as a structured representation of the learning micro-environment to describe the group context for each student.

3.3 Bi-granular Summary Construction and Community-aware Retrieval

Inspired by Collaborative Learning Theory (O'Donnell and Hmelo-Silver, 2013), we argue that an individual learner's ability and performance should be understood within the group reference of the learning micro-environment they belong to, and further contrasted with similar peers within the same community to reveal differences. We propose a bi-granular summary construction and community-aware retrieval mechanism to characterize students and learning communities by ability and multi-dimensional learning performance, yielding reusable evidence. This module consists of the following three stages.

3.3.1 Bi-granular Summary Construction

For each target student u, we construct a student-level statistical summary A_u, which includes a characterization of student ability and multi-dimensional learning performance statistics, specifically including overall performance, KC-level performance, and question-level performance. Correspondingly, for each learning community C_m, we construct a community-level statistical summary A_{C_m}, which includes a characterization of community ability, overall community performance, KC-level group performance, and an overview of question-level group performance. More details of statistical information see Appendix D.

Based on the statistical summaries, we use an LLM to generate natural language summaries. Let the prompt templates for students and communities be P_{stu} and P_{com}, respectively. The natural language summaries are defined as:

$$\tilde{A}_u = \text{LLM}(P_{\text{stu}}, A_u) \tag{2}$$

$$\tilde{A}_{C_m} = \text{LLM}(P_{\text{com}}, A_{C_m}) \tag{3}$$

The resulting bi-granular natural language summaries for students and communities provide a more compact and readable semantic expression

while maintaining evidence consistency. The detailed prompt template see Appendix N.

3.3.2 Community Membership Matching

To map a target student at test time to existing learning communities, we adopt a dual-channel matching scheme with a statistical channel and a textual channel, and obtain the final matching score via weighted fusion.

In the statistical channel, we represent the student statistical summary A_u and the community statistical summary A_{C_m} as vector features f_u and f_{C_m}. The consistency between them is then measured by cosine similarity:

$$s_{\text{feat}}(u, C_m) = \cos(f_u, f_{C_m}) \tag{4}$$

In the textual channel, we feed the student natural language summary $\tilde{A}_u$ and the community natural language summary $\tilde{A}_{C_m}$ into a Sentence-BERT encoder $\phi(\cdot)$ to obtain semantic vectors $t_u = \phi(\tilde{A}_u)$ and $t_{C_m} = \phi(\tilde{A}_{C_m})$, and likewise using cosine similarity:

$$s_{\text{text}}(u, C_m) = \cos(t_u, t_{C_m}) \tag{5}$$

The two channel scores are fused with coefficient λ to obtain the final matching score:

$$s(u, C_m) = \lambda\, s_{\text{text}}(u, C_m) + (1 - \lambda)\, s_{\text{feat}}(u, C_m) \tag{6}$$

based on which the target student is assigned to the community with the highest score:

$$m^\star = \arg \max_{m \in \{1,\dots,M\}} s(u, C_m) \tag{7}$$

3.3.3 Community-based Evidence Retrieval

After determining the learning community of the target student, we further collect community-based evidence to support a more reliable and comparable inference context for the target student. Specifically, for any target student u and its community $C_{m^\star}$, we consider the candidate peer set $V_u = \{\, v \in C_{m^\star} \mid v \neq u \,\}$. We then assess how representative each peer v is for forming the target student's learning micro-environment evidence by measuring the similarity between (u, v) from four complementary signals (e.g., ability, performance, shared questions and KCs), and fuse the ranking results induced by these signals to obtain a robust peer ordering. The specific four complementary signals can be found in Appendix E.

To fuse multi-route ranking results, we adopt reciprocal rank fusion (RRF) (Cormack et al., 2009) to define the aggregated score of a candidate peer v as:

$$\text{RRF}(v) = \sum_{r=1}^{R} \frac{1}{\kappa + \text{rank}_r(v)} \tag{8}$$

where $\text{rank}_r(v)$ denotes the rank of v under the r-th criterion, κ is a smoothing constant, and R is the number of routes that actually cast votes. Finally, we select the Top-K candidates with the highest aggregated scores as the similar peer set N_u, and use the peers' information together with the community information to form the key community-based evidence for the target student.

3.4 Community Inspired KT Prediction

Traditional KT models typically output only numerical predictions and lack high-level semantic explanations of the evidence behind the predictions, making it difficult to meet the interpretability demands in educational scenarios. To this end, we incorporate group-referential evidence from the learning micro-environment into the prompt and employ a LLM to simultaneously perform KT prediction and generate an explanatory analysis report. Since this stage adopts a zero-shot setting without fine-tuning the LLM, the structured design of the prompt template is particularly critical. The complete prompt template is provided in Appendix O.

The structured prompt consists of three types of information: (1) ***target student information***: including the student's ability, multi-dimensional learning performance summaries, and the historical interaction sequence; (2) ***target question information***: including the question attributes; (3) ***learning micro-environment evidence***: including the assigned community information of the target student and similar peers information.

Given the prompt template P_{pred} and evidence $\mathcal{X}_{u,q}$, the LLM output is defined as:

$$(\hat{y}_{u,q},\ p_{u,q},\ \mathcal{R}_{u,q}) = \text{LLM}(P_{\text{pred}}, \mathcal{X}_{u,q}) \tag{9}$$

where $\hat{y}_{u,q} \in \{\textit{correct}, \textit{wrong}\}$ is the predicted answer outcome, $p_{u,q} \in [0, 1]$ is the confidence probability of being correct, and $\mathcal{R}_{u,q}$ is the natural language analysis report that explains the predictive evidence and reasoning process.

4 Experiment

In this section, we conduct extensive experiments to illustrate the effectiveness of our method.

Category	Model	Assistment09		Statics2011		DBE-KT22		Frcsub	
		ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
DL-based Methods	DKT (Piech et al., 2015)	0.7218	0.7452	0.7915	0.7927	0.7122	0.7219	0.7952	0.8695
	DKVMN (Zhang et al., 2017)	0.7129	0.7442	0.7668	0.7774	0.7038	0.7157	0.8287	0.9023
	AKT (Ghosh et al., 2020)	0.7380	0.7788	0.8230	0.8242	0.7292	0.7539	0.8218	0.8954
	SAINT+ (Shin et al., 2021)	0.7323	0.7658	0.8092	0.8164	0.6987	0.7178	0.8139	0.8775
	DTransformer (Yin et al., 2023)	0.7534	0.8036	0.8068	0.8322	0.7254	0.7391	0.8254	0.8861
Graph-based Methods	GKT (Nakagawa et al., 2019)	0.7224	0.7459	0.7862	0.7888	0.6707	0.6620	0.7589	0.8387
	GIKT (Yang et al., 2020)	0.7598	0.7852	0.8036	0.8256	0.7254	0.7475	0.7928	0.8643
	PEBG+DKT (Liu et al., 2020)	0.7668	0.8083	0.8013	0.8179	0.7319	0.7582	0.8246	0.9187
	SimQE (Sun et al., 2024)	0.7643	0.8027	0.8111	0.8321	0.7255	0.7467	0.8255	0.9163
	STHKT (Li et al., 2025d)	0.7682	0.8056	0.8082	0.8295	0.7038	0.7145	0.8212	0.9135
	DyGKT (Cheng et al., 2024)	0.7644	0.7983	0.8125	0.8296	0.7051	0.7262	0.8039	0.8837
Hypergraph-based Methods	HyperKT (Li et al., 2024b)	0.7589	0.7933	0.7993	0.8126	0.7228	0.7320	0.8042	0.8756
	DGEKT (Cui et al., 2024)	0.7413	0.7856	0.7748	0.7849	0.7105	0.7369	0.8352	0.8964
	HDKT (Hou et al., 2025)	0.7524	0.7835	0.8032	0.8346	0.7273	0.7438	0.8349	0.8861
LLM-based Methods	EPFL (Neshaei et al., 2024)	0.6013	0.5827	0.6962	0.7189	0.5526	0.5599	0.8538	0.7446
	EFKT (Li et al., 2025a)	0.7641	0.7945	0.7342	0.7671	0.5702	0.5708	0.8091	0.8758
	LLM-KT (Wang et al., 2025)	0.7752	0.8132	0.8146	0.8423	0.6983	0.7234	0.8231	0.8056
	CIKT (Li et al., 2025c)	0.7523	0.7635	0.7325	0.7868	0.6854	0.7058	0.8048	0.8376
	HISE-KT (Duan et al., 2025)	0.7834	0.8166	0.8201	0.8459	0.6842	0.7071	0.8524	0.9209
	2T-KT (Li et al., 2025b)	0.7392	0.7634	0.7698	0.7825	0.7036	0.7325	0.8036	0.8203
Ours	MicroC-KT$_{\text{DeepSeek-V3}}$	0.7708	0.8263	0.8287	0.8693	0.7276	0.7866	0.8545	0.9006
	MicroC-KT$_{\text{GPT-4o}}$	0.7940	0.8380	0.8324	0.8841	0.7223	0.7731	0.8685	0.9264
	MicroC-KT$_{\text{Qwen-Plus}}$	**0.8106**	**0.8595**	**0.8485**	**0.8987**	**0.7565**	**0.8084**	**0.8727**	**0.9486**

Table 1: Main results on four benchmark datasets. All results are reported as the mean of five runs. Best result is indicated in **bold**, and the second-best result is indicated underlined.

4.1 Experimental Settings

Datasets. We evaluate the performance of MicroC-KT on four benchmark datasets: **Assistment09** (Feng et al., 2009), **Statics2011** (Koedinger et al., 2010), **DBE-KT22** (Abdelrahman et al., 2022) and **Frcsub** (Wu et al., 2015). More details are provided in Appendix F.

Baselines. For a comprehensive comparison, we evaluate our proposed method against four categories of baselines: Deep Learning (DL)-based, Graph-based, Hypergraph-based, and LLM-based methods. The specific descriptions of each baseline are provided in Appendix G.

Implementation Details. We split each dataset into training, validation, and test sets with a ratio of 8:1:1. The hypergraph is constructed solely from the training set. For summary generation and KT prediction, we use DeepSeek-V3 (Liu et al., 2024), GPT-4o (Hurst et al., 2024), and Qwen-Plus (Bai et al., 2023) in a zero-shot manner without additional fine-tuning. To ensure a fair comparison, all LLM-based baselines were reproduced under the exact same LLM backbones and context budgets. We report their experimental results based on the best-performing backbone. In community membership matching, the fusion coefficient λ between the textual and statistical channels is selected via grid search on the validation set over $\{0.0, 0.1, \ldots, 1.0\}$. For reciprocal rank fusion, the smoothing constant κ is set to 60. More implementation details are provided in Appendix H.

4.2 Main Results

Tab. 1 reports the ACC and AUC of all compared methods. Overall, MicroC-KT consistently achieves the best performance across all categories of baseline methods, indicating the effectiveness of our proposed method. This advantage mainly stems from introducing learning micro-environment evidence into the KT process. By constructing a learning environment hypergraph and deriving community-level micro-environments, our framework provides a structured group reference for each target student, and further retrieves comparable peers within the same community. Such community-aware evidence complements individual interaction histories with higher-order and student-side relational signals, enabling more reliable and informative context for LLM-based prediction. It reduced reliance on often overly brittle single-student cues, especially when observations are sparse or noisy. We further evaluate the effi-

Method	Assistment09		Statics2011		DBE-KT22		Frcsub	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
Full	**0.8106**	**0.8595**	**0.8485**	**0.8987**	**0.7565**	**0.8084**	**0.8727**	**0.9486**
w/o Comm.	0.7733	0.8128	0.8182	0.8472	0.7038	0.7419	0.8453	0.8592
w/o Peers	0.7241	0.7932	0.7576	0.8213	0.7094	0.7366	0.8364	0.8306
w/o SS	0.7758	0.8263	0.7883	0.8234	0.7012	0.7409	0.8569	0.8451
w/o TS	0.7608	0.8343	0.7273	0.8393	0.7179	0.7730	0.8545	0.8196
RC	0.6974	0.7535	0.6970	0.7996	0.6667	0.6909	0.6818	0.6685
RP	0.7575	0.8174	0.7879	0.8611	0.7009	0.7373	0.8545	0.8397
CCP	0.7741	0.8227	0.7612	0.8358	0.7265	0.7569	0.7455	0.7436

Table 2: Ablation results of MicroC-KT on four benchmark datasets.

ciency of MicroC-KT and all baselines. The detailed results are presented in Appendix I.

4.3 Ablation Study

To further investigate the contribution of each component in MicroC-KT, we conduct ablation study with seven variants. As shown in Tab. 2: (1) **w/o Comm.** indicates removing the community-level evidence of the assigned learning community at inference time, so the community summary and group reference are not provided; (2) **w/o Peers** indicates removing the retrieved similar peers at inference time, so no peer-contrast evidence is provided; (3) **w/o SS** indicates removing the *structured statistical summaries* at inference time and keeping only the textual summaries as evidence input; (4) **w/o TS** indicates removing the *textual summaries* at inference time and keeping only the structured statistical summaries; (5) **RC** (**Random Community**) indicates randomly assigning the target student to a community, thus using an incorrect community reference and organizing subsequent evidence within that random community; (6) **RP** (**Random Peers**) indicates keeping community matching unchanged but replacing the retrieved peers with randomly sampled students from the matched community; (7) **CCP** (**Cross Community Peers**) indicates removing the community constraint and retrieving peers from other communities, resulting in cross-community peer retrieval.

As shown in Tab. 2, removing community information or similar peers leads to clear performance drops, indicating that *learning micro-environment evidence* provides essential support for LLM-based prediction. Using only natural-language summaries or only statistical features also degrades performance, suggesting that textual and statistical evidence are complementary and their combination offers more consistent decision cues. Randomly assigning communities causes the largest degradation

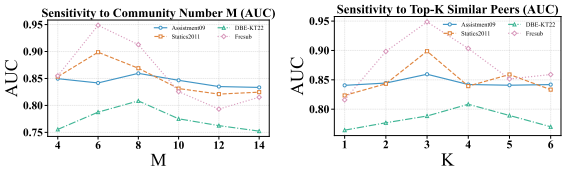

Figure 3: Parameter sensitivity of the community number M and the number of similar peers Top-K on AUC.

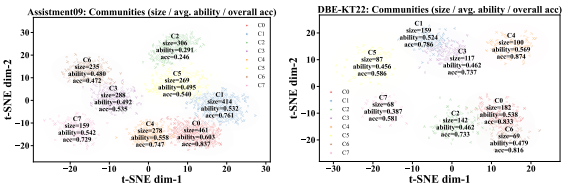

Figure 4: Visualizations of community clustering analysis on Assistment09 and DBE-KT22 datasets.

because the micro-environment evidence becomes systematically corrupted and can mislead the reasoning process; in contrast, randomly selecting peers within the community still outperforms w/o peers, showing that the community constraint itself provides a meaningful local reference. Finally, retrieving peers across communities performs worse than community-restricted retrieval, further validating community partitioning imposes an effective local constraint, making the retrieved peers more comparable and the evidence more coherent.

4.4 Parameter Sensitivity

Fig. 3 shows the impact of the community number M and the number of similar peers Top-K on performance in terms of AUC (the complete ACC and AUC results are provided in Appendix J). For the community number M, too few communities will coarsely mix learners with different cognitive states, making it difficult to characterize clear cognitive groups; too many communities tend to form fragmented groups, which makes the learning micro-environment evidence unstable and weakens comparability. Overall, $M = 6 \sim 8$ is more appropriate, enabling a clearer partition of learning micro-environments with different cognitive states.

For the number of similar peers Top-K, a too small K leads to insufficient peer evidence, while a too large K may introduce weakly-related noise and dilute key information. $K = 3$ usually works best, achieving a better balance between evidence sufficiency and noise control.

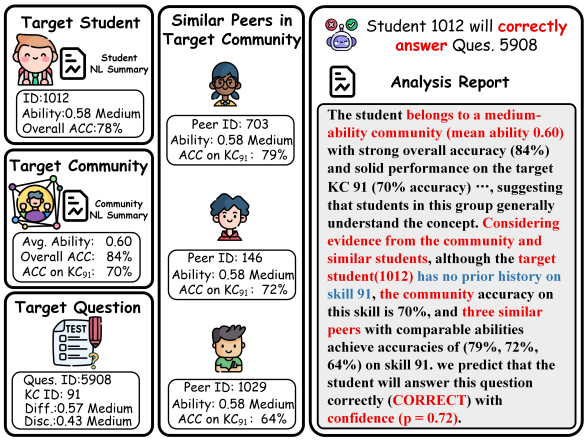

Figure 5: Case study of MicroC-KT. MicroC-KT leverages in-community evidence to predict whether the target student can answer the target question correctly and generates an analysis report.

Method	EG	FC	TA	PU	CC	Overall
EFKT	2.35	2.41	2.28	2.12	2.50	2.33
CIKT	3.08	3.12	3.01	2.86	3.22	3.06
HISE-KT	4.18	4.23	4.12	4.05	4.31	4.18
MicroC-KT$_{\text{Qwen-Plus}}$	4.82	4.86	4.79	4.74	4.90	4.82

Table 3: Human evaluation of analysis report quality on 40 sampled students (10 per dataset) across four datasets. We evaluate report quality from five dimensions (*Evidence Groundedness (EG)*, *Factual Consistency (FC)*, *Targeted Analysis (TA)*, *Pedagogical Usefulness (PU)*, and *Clarity and Coherence (CC)*). Each dimension is rated on a 1–5 scale (higher is better), and **Overall** is the average over five dimensions. Five experts independently rated all reports, with inter-rater agreement measured by the Cohen's kappa coefficient $\kappa^{\dagger} = 0.82$.

4.5 Clustering Analysis

As shown in Fig. 4 (results on Assistment09 and DBE-KT22; full visualizations on all datasets are provided in Appendix K), we further conduct clustering analysis to intuitively validate the effectiveness of the learning micro-environment. Based on the community partition, we visualize student representations using t-SNE (Maaten and Hinton, 2008). Distinct communities form clusters with clear boundaries and high separability in the 2D space, indicating that MicroC-KT can aggregate students with similar cognitive states and response behaviors into the same community and effectively distinguish different learning micro-environments.

Meanwhile, Fig. 4 presents several interpretable attributes for each community, such as community size, average ability, and overall accuracy, to summarize the micro-environment characteristics of different clusters. Overall, students within the same community exhibit similar cognitive levels and learning performance, and clusters that are closer in the embedding space tend to correspond to more similar group features. These results support the rationality of our community partition and provide a reliable group foundation for community-restricted retrieval and evidence provision.

4.6 Case Study

As shown in Fig. 5, MicroC-KT performs inference by combining the target student and target question information while introducing the evidence from the learning micro-environment. MicroC-KT first matches the target student to the assigned learning community. The community-level accuracy on the target knowledge concept KC_{91} is 70%, which serves as a stable community reference. It then retrieves several similar peers within the community. Their accuracies on KC_{91} are 79%, 72%, and 64%, respectively, providing peer contrastive evidence. Based on the consistent support from the community reference and the peer evidence, LLM can still make an accurate prediction even if the target student has no prior responses on KC_{91}, and provides the corresponding analysis report. The natural-language summaries of the target student and the matched community see Appendix. M.

4.7 Quality Evaluation of Analysis Reports

We conduct a human evaluation to assess the quality of the analysis reports generated by different LLM-based KT methods. Tab. 3 summarizes the results, MicroC-KT achieves the highest score among all compared methods. It consistently yields more grounded and factually consistent analyses, indicating that the generated reports better align with the provided evidence. This suggests that incorporating micro-environment evidence can improve the reliability of LLM-generated reports. Detailed evaluation criteria and results analysis see Appendix L.

4.8 Token Consumption and Cost Analysis

To evaluate the deployment feasibility of our framework, we conduct a fine-grained quantitative analysis of the token consumption based on the prompt structure. As detailed in Tab. 4, the prompt comprises several modules, including system instructions, target student information, target question information, and multi-granularity micro-environment evidence. The total input length is tightly controlled at approximately 1,200 tokens,

and the generated diagnostic report consumes about 200 tokens, resulting in a total token usage of roughly 1,400 tokens per inference.

Based on this estimation, when utilizing Qwen-Plus as the LLM backbone, the API cost for generating 1,000 diagnostic predictions and reports is merely about $1.0 USD. Even when employing a more advanced model like GPT-4o, the cost per 1,000 diagnoses remains highly controllable at approximately $5.0 USD. Furthermore, MicroC-KT is fully compatible with locally deployed open-source LLMs, which can completely eliminate API token costs, making the framework highly scalable and economical for real-world educational applications.

Stage	Component	Estimated Tokens
Input	System Instructions	∼220
	Target Student Information	∼400
	Target Question Information	∼50
	Target Community Evidence	∼250
	Similar Peer Evidence	∼100
	Total Input	*∼1200*
Output	Prediction Result & Diagnostic Report	∼200
Total		**∼1400**

Table 4: Estimated token consumption for a single prediction and diagnostic report generation based on the prompt structure.

5 Conclusion

In this paper, we propose MicroC-KT, an innovative knowledge tracing framework tailored for learning micro-environments, aiming to deliver more accurate and trustworthy predictions with explanations. The key idea is to explicitly model group cognition and community reference at the student level. Specifically, we construct a learning environment hypergraph and perform spectral clustering to discover learning communities. Furthermore, we build bi-granular summaries for students and communities, enabling community matching and within-community peer evidence retrieval, thus providing the LLM with consistent and contrastive evidence support. Finally, MicroC-KT jointly feeds the target student, question, and learning micro-environment evidence into the LLM, achieving KT prediction and generating the corresponding analysis report simultaneously. Extensive experiments on four public datasets demonstrate that MicroC-KT outperforms strong baselines in both predictive performance and explanation reliability.

Limitations

While MicroC-KT demonstrates promising results, it has several limitations. First, its predictive performance may fluctuate across LLMs of different scales and reasoning capacities, and smaller models with fewer parameters and weaker reasoning ability can incur a slight performance drop. Second, constrained by the context window and inference cost, we do not fully incorporate richer fine-grained information such as question text or longer interaction histories, which may limit more detailed modeling and explanations.

Acknowledgments

This work was funded by the National Natural Science Foundation of China (Nos. 62567005 and 62206107), and Natural Science Foundation of Inner Mongolia Autonomous Region of China (No. 2025MS06004), and in part by Program for Young Talents of Science and Technology in Universities of Inner Mongolia A. R. of China under Grant NJYT25011.

References

Ghodai Abdelrahman, Sherif Abdelfattah, Qing Wang, and Yu Lin. 2022. Dbe-kt22: A knowledge tracing dataset based on online student evaluation. *arXiv preprint arXiv:2208.12651*.

Paul Adams. 2006. Exploring social constructivism: Theories and practicalities. *Education*, 34(3):243–257.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Allan Birnbaum. 1968. Some latent trait models. *Statistical theories of mental test scores*.

Ke Cheng, Linzhi Peng, Pengyang Wang, Junchen Ye, Leilei Sun, and Bowen Du. 2024. Dygkt: Dynamic graph learning for knowledge tracing. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 409–420.

Albert T Corbett and John R Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759.

Chaoran Cui, Yumo Yao, Chunyun Zhang, Hebo Ma, Yuling Ma, Zhaochun Ren, Chen Zhang, and James Ko. 2024. Dgekt: A dual graph ensemble learning method for knowledge tracing. *ACM Transactions on Information Systems*, 42(3):1–24.

Zhiyi Duan, Xiaoxiao Dong, Hengnian Gu, Xiong Wu, Zhen Li, and Dongdai Zhou. 2024. Towards more accurate and interpretable model: Fusing multiple knowledge relations into deep knowledge tracing. *Expert Systems with Applications*, 243:122573.

Zhiyi Duan, Zixing Shi, Hongyu Yuan, and Qi Wang. 2025. Hise-kt: Synergizing heterogeneous information networks and llms for explainable knowledge tracing with meta-path optimization. *arXiv preprint arXiv:2511.15191*.

William N Evans, Wallace E Oates, and Robert M Schwab. 1992. Measuring peer group effects: A study of teenage behavior. *Journal of Political Economy*, 100(5):966–991.

Mingyu Feng, Neil Heffernan, and Kenneth Koedinger. 2009. Addressing the assessment challenge with an online system that tutors as it assesses. *User modeling and user-adapted interaction*, 19(3):243–266.

Xiang Gao, Weiqing Chen, Ying Cui, Xiang Dai, and Lican Dai. 2025. Progressive adversarial contrastive learning: Towards efficient data augmentation in adversarial defense. *DATA INTELLIGENCE*, 7(4):1169–1191.

Aritra Ghosh, Neil Heffernan, and Andrew S Lan. 2020. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2330–2339.

Chenyu Hou, Aochen You, Bin Cao, Dongjing Wang, and Jing Fan. 2025. Hypergraph-based de-conjecture knowledge tracing for intelligent educational services. In *2025 IEEE International Conference on Web Services (ICWS)*, pages 931–942. IEEE.

Chang-Qin Huang, Qiong-Hao Huang, Xiaodi Huang, Hua Wang, Ming Li, Kwei-Jay Lin, and Yi Chang. 2024. Xkt: toward explainable knowledge tracing model with cognitive learning theories for questions of multiple knowledge concepts. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):7308–7325.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Kenneth R Koedinger, Ryan SJd Baker, Kyle Cunningham, Alida Skogsholm, Brett Leber, and John Stamper. 2010. A data repository for the edm community: The pslc datashop. *Handbook of educational data mining*, 43:43–56.

Deyi Li, Jialun Yin, Tianlei Zhang, Wei Han, and Hong Bao. 2024a. The four most basic elements in machine cognition. *DATA INTELLIGENCE*, 6(2):297–319.

Haoxuan Li, Jifan Yu, Yuanxin Ouyang, Zhuang Liu, Wenge Rong, Huiqin Liu, Juanzi Li, and Zhang Xiong. 2025a. Explainable few-shot knowledge tracing. *Frontiers of Digital Education*, 2(4):34.

Jiawei Li, Yuanfei Deng, Yixiu Qin, Shun Mao, and Yuncheng Jiang. 2024b. Dual-channel adaptive scale hypergraph encoders with cross-view contrastive learning for knowledge tracing. *IEEE Transactions on Neural Networks and Learning Systems*, 36(4):6752–6766.

Linqing Li, Zhifeng Wang, Joemon M Jose, and Xuri Ge. 2025b. Llm supporting knowledge tracing leveraging global subject and student specific knowledge graphs. *Information Fusion*, page 103577.

Runze Li, Siyu Wu, Jun Wang, and Wei Zhang. 2025c. CIKT: A collaborative and iterative knowledge tracing framework with large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 19332–19345.

Shuting Li, Shuanghong Shen, Yu Su, Xinjie Sun, Junyu Lu, Qi Mo, Zhenyi Wu, and Qi Liu. 2025d. Sthkt: Spatiotemporal knowledge tracing with topological hawkes process. *Expert Systems with Applications*, 259:125248.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Yunfei Liu, Yang Yang, Xianyu Chen, Jian Shen, Haifeng Zhang, and Yong Yu. 2020. Improving knowledge tracing via pre-training question embeddings. *arXiv preprint arXiv:2012.05031*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Hiromi Nakagawa, Yusuke Iwasawa, and Yutaka Matsuo. 2019. Graph-based knowledge tracing: modeling student proficiency using graph neural network. In *IEEE/WIC/aCM international conference on web intelligence*, pages 156–163.

Seyed Parsa Neshaei, Richard Lee Davis, Adam Hazimeh, Bojan Lazarevski, Pierre Dillenbourg, and Tanja Käser. 2024. Towards modeling learner performance with large language models. *arXiv preprint arXiv:2403.14661*.

Angela M O'Donnell and Cindy E Hmelo-Silver. 2013. Introduction: What is collaborative learning?: An overview. *The international handbook of collaborative learning*, pages 1–15.

Sami Paavola, Lasse Lipponen, and Kai Hakkarainen. 2004. Models of innovative knowledge communities and three metaphors of learning. *Review of educational research*, 74(4):557–576.

Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. *Advances in neural information processing systems*, 28.

Shuanghong Shen, Qi Liu, Zhenya Huang, Yonghe Zheng, Minghao Yin, Minjuan Wang, and Enhong Chen. 2024. A survey of knowledge tracing: Models, variants, and applications. *IEEE Transactions on Learning Technologies*, 17:1858–1879.

Dongmin Shin, Yugeun Shim, Hangyeol Yu, Seewoo Lee, Byungsoo Kim, and Youngduck Choi. 2021. Saint+: Integrating temporal features for ednet correctness prediction. In *LAK21: 11th international learning analytics and knowledge conference*, pages 490–496.

Jianwen Sun, Shangheng Du, Jianpeng Zhou, Xin Yuan, Xiaoxuan Shen, and Ruxia Liang. 2024. Question embedding on weighted heterogeneous information network for knowledge tracing. *ACM Transactions on Knowledge Discovery from Data*, 19(1):1–28.

Mengshu Sun, Yichi Zhang, Zhiqiang Liu, Lei Liang, and Wen Zhang. 2025. Cognitive-uncertainty guided knowledge distillation for accurate classification of student misconceptions. *DATA INTELLIGENCE*, 7(3):765–775.

Ziwei Wang, Jie Zhou, Qin Chen, Min Zhang, Bo Jiang, Aimin Zhou, Qinchun Bai, and Liang He. 2025. Llm-kt: Aligning large language models with knowledge tracing using a plug-and-play instruction. *arXiv preprint arXiv:2502.02945*.

Run-ze Wu, Qi Liu, Yuping Liu, Enhong Chen, Yu Su, Zhigang Chen, and Guoping Hu. 2015. Cognitive modelling for predicting examinee performance. In *IJCAI*, pages 1017–1024.

Yang Yang, Jian Shen, Yanru Qu, Yunfei Liu, Kerong Wang, Yaoming Zhu, Weinan Zhang, and Yong Yu. 2020. Gikt: a graph-based interaction model for knowledge tracing. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 299–315. Springer.

Yu Yin, Le Dai, Zhenya Huang, Shuanghong Shen, Fei Wang, Qi Liu, Enhong Chen, and Xin Li. 2023. Tracing knowledge instead of patterns: Stable knowledge tracing with diagnostic transformer. In *Proceedings of the ACM web conference 2023*, pages 855–864.

Jaehong Yoon, Shoubin Yu, Vaidehi Patil, Huaxiu Yao, and Mohit Bansal. 2024. Safree: Training-free and adaptive guard for safe text-to-image and video generation. *arXiv preprint arXiv:2410.12761*.

Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. 2017. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th international conference on World Wide Web*, pages 765–774.

Xilin Zhang, Zhixin Mao, Ziwen Chen, and Shen Gao. 2024. Effective tool augmented multi-agent framework for data analysis. *DATA INTELLIGENCE*, 6(4):923–945.

Lixiao Zheng, Jipeng Xiao, Shuai Ma, Zuxi Chen, and Xiangyu Luo. 2025. Temporal cycle enumeration for detecting financial fraud. *DATA INTELLIGENCE*, 7(3):567–590.

Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. 2006. Learning with hypergraphs: Clustering, classification, and embedding. *Advances in neural information processing systems*, 19.

A External Attribute Construction

Given the student interaction logs, we denote the response outcome of student u on question q as $y_{u,q} \in \{0, 1\}$. We employ the IRT-2PL model to estimate an ability parameter θ_u for each student, and estimate a difficulty parameter b_q and a discrimination parameter a_q for each question. The probability of a correct response is defined as:

$$P(y_{u,q} = 1 \mid \theta_u, a_q, b_q) = \sigma\big(a_q(\theta_u - b_q)\big) \quad (10)$$

where $\sigma(\cdot)$ is the Sigmoid function. The parameters $\{\theta_u\}$, $\{b_q\}$, and $\{a_q\}$ are obtained via maximum likelihood estimation. To improve comparability and numerical stability, we standardize the ability parameters as $\theta_u \leftarrow (\theta_u - \mu_\theta)/\sigma_\theta$, where μ_θ and σ_θ are the mean and standard deviation of $\{\theta_u\}$, respectively.

To obtain consistent and interpretable semantic labels, we discretize continuous attributes into finite levels. The ability level is defined as $c_u^\theta \in \{\text{low}, \text{mid}, \text{high}\}$, using a three-interval partition based on the mean and standard deviation:

$$c_u^\theta = \begin{cases} \text{low}, & \theta_u \le \mu_\theta - \sigma_\theta \\ \text{mid}, & \mu_\theta - \sigma_\theta < \theta_u < \mu_\theta + \sigma_\theta \\ \text{high}, & \theta_u \ge \mu_\theta + \sigma_\theta \end{cases} \quad (11)$$

When it is necessary to compute stratified performance statistics by question attributes, we also discretize question difficulty and discrimination into three levels: $c_q^b \in \{\text{low}, \text{mid}, \text{high}\}$ and $c_q^a \in \{\text{low}, \text{mid}, \text{high}\}$, and apply the same three-interval partition rule as ability to $\{b_q\}$ and $\{a_q\}$, respectively.

We use accuracy as the learning performance statistic. For student u, the overall accuracy is:

$$\text{acc}_u = \frac{\sum_{(u,q)\in\Omega} y_{u,q}}{\sum_{(u,q)\in\Omega} 1} \quad (12)$$

At the knowledge concept level, let $\Omega_u(k)$ denote the interaction set of student u on questions associated with concept k, we have:

$$\text{acc}_u(k) = \frac{\sum_{(u,q)\in\Omega_u(k)} y_{u,q}}{\sum_{(u,q)\in\Omega_u(k)} 1} \quad (13)$$

At the difficulty-stratified level, let $\Omega_u^{\text{diff}}(d)$ denote the interaction set of student u on questions whose difficulty level is $c_q^b = d$, we have:

$$\text{acc}_u^{\text{diff}}(d) = \frac{\sum_{(u,q)\in\Omega_u^{\text{diff}}(d)} y_{u,q}}{\sum_{(u,q)\in\Omega_u^{\text{diff}}(d)} 1} \quad (14)$$

At the discrimination-stratified level, let $\Omega_u^{\text{disc}}(t)$ denote the interaction set of student u on questions whose discrimination level is $c_q^a = t$, we have:

$$\text{acc}_u^{\text{disc}}(t) = \frac{\sum_{(u,q)\in\Omega_u^{\text{disc}}(t)} y_{u,q}}{\sum_{(u,q)\in\Omega_u^{\text{disc}}(t)} 1} \quad (15)$$

Finally, we map any performance value $z \in [0, 1]$ to a performance level $\rho(z) \in \{\text{poor}, \text{avg}, \text{good}\}$ by defining two thresholds $\tau_{\text{poor}} < \tau_{\text{good}}$ as:

$$\rho(z) = \begin{cases} \text{poor}, & z < \tau_{\text{poor}} \\ \text{avg}, & \tau_{\text{poor}} \le z < \tau_{\text{good}} \\ \text{good}, & z \ge \tau_{\text{good}} \end{cases} \quad (16)$$

Here, we define $\tau_{\text{poor}} = 0.4$ and $\tau_{\text{good}} = 0.6$. This discretization provides an external attribute basis for hyperedge definitions.

B Ability-aware Hyperedge Construction

Given the student set V, we discretize each student's ability parameter θ_u into an ability level $c_u^\theta \in \{\text{low}, \text{mid}, \text{high}\}$. Ability-aware hyperedges provide a stable global ability reference: we use the ability level as the semantic anchor and aggregate students in the same ability tier into the same higher-order relation. Specifically, for any ability level $l \in \{\text{low}, \text{mid}, \text{high}\}$, we define an ability-aware hyperedge as:

$$e_l^\theta = \{\, u \in V \mid c_u^\theta = l \,\} \quad (17)$$

This yields the ability-aware hyperedge set:

$$\mathcal{E}_{\text{abl}} = \{e_{\text{low}}^\theta,\ e_{\text{mid}}^\theta,\ e_{\text{high}}^\theta\} \quad (18)$$

We assign each student u to the corresponding hyperedge e_l^θ according to its ability level c_u^θ, thereby forming higher-order aggregation relations for the low-ability, mid-ability, and high-ability groups, and satisfies $\bigcup_l e_l^\theta = V$.

C Performance-aware Hyperedge Construction

Given the student set V, we further construct the performance-aware hyperedge set $\mathcal{E}_{\text{perf}}$ to characterize group consistency in students' multi-dimensional learning performance. This set consists of three types of hyperedges:

$$\mathcal{E}_{\text{perf}} = \mathcal{E}_{\text{kc}} \cup \mathcal{E}_{\text{diff}} \cup \mathcal{E}_{\text{disc}} \quad (19)$$

where $\mathcal{E}_{\text{kc}}$ denotes the knowledge-concept performance hyperedge set, $\mathcal{E}_{\text{diff}}$ denotes the difficulty-stratified performance hyperedge set, and $\mathcal{E}_{\text{disc}}$ denotes the discrimination-stratified performance hyperedge set. All three types of hyperedges are constructed based on the performance level mapping function $\rho(\cdot)$, where $\rho(\cdot) \in \{\text{poor}, \text{avg}, \text{good}\}$.

C.1 Knowledge-concept Performance Hyperedges

For any knowledge concept k and any performance level $r \in \{\text{poor}, \text{avg}, \text{good}\}$, we define a knowledge-concept performance hyperedge as:

$$e^{\text{kc}}_{k,r} = \{\, u \in V \mid \rho(\text{acc}_u(k)) = r \,\} \tag{20}$$

This yields the knowledge-concept performance hyperedge set:

$$\mathcal{E}_{\text{kc}} = \{\, e^{\text{kc}}_{k,r} \mid k \in \mathcal{K},\ r \in \{\text{poor}, \text{avg}, \text{good}\} \,\} \tag{21}$$

where $\mathcal{K}$ is the set of knowledge concepts. This construction aggregates students with the same performance level on the same knowledge concept into the same higher-order relation.

C.2 Difficulty-stratified Performance Hyperedges

For any difficulty level $d \in \{\text{low}, \text{mid}, \text{high}\}$ and any performance level $r \in \{\text{poor}, \text{avg}, \text{good}\}$, we define a difficulty-stratified performance hyperedge as:

$$e^{\text{diff}}_{d,r} = \{\, u \in V \mid \rho(\text{acc}^{\text{diff}}_u(d)) = r \,\} \tag{22}$$

This yields the difficulty-stratified performance hyperedge set:

$$\mathcal{E}_{\text{diff}} = \{\, e^{\text{diff}}_{d,r} \mid d \in \{\text{low}, \text{mid}, \text{high}\}, \quad r \in \{\text{poor}, \text{avg}, \text{good}\} \,\} \tag{23}$$

This construction aggregates students with the same performance level under the same difficulty tier into the same higher-order relation.

C.3 Discrimination-stratified Performance Hyperedges

For any discrimination level $t \in \{\text{low}, \text{mid}, \text{high}\}$ and any performance level $r \in \{\text{poor}, \text{avg}, \text{good}\}$, we define a discrimination-stratified performance hyperedge as:

$$e^{\text{disc}}_{t,r} = \{\, u \in V \mid \rho(\text{acc}^{\text{disc}}_u(t)) = r \,\} \tag{24}$$

This yields the discrimination-stratified performance hyperedge set:

$$\mathcal{E}_{\text{disc}} = \{\, e^{\text{disc}}_{t,r} \mid t \in \{\text{low}, \text{mid}, \text{high}\}, \quad r \in \{\text{poor}, \text{avg}, \text{good}\} \,\} \tag{25}$$

This construction aggregates students with the same performance level under the same discrimination tier into the same higher-order relation.

D Statistical Summary Construction

For each target student u, we construct a student-level statistical summary A_u to characterize student ability and multi-dimensional learning performance statistics, including the following components:

(1) **Student ability.** The ability parameter θ_u and the discretized ability level $c^{\theta}_u \in \{\text{low}, \text{mid}, \text{high}\}$.

(2) **Overall performance.** The overall accuracy acc_u.

(3) **KC-level performance.** For each knowledge concept k, we compute the KC-level accuracy $\text{acc}_u(k)$ to form a concept-wise performance profile.

(4) **Question-attribute-level performance.** For each difficulty level $d \in \{\text{low}, \text{mid}, \text{high}\}$, we compute $\text{acc}^{\text{diff}}_u(d)$, and for each discrimination level $t \in \{\text{low}, \text{mid}, \text{high}\}$, we compute $\text{acc}^{\text{disc}}_u(t)$, to capture student behavior differences under different question-attribute conditions.

For each learning community C_m, we construct a community-level statistical summary A_{C_m} to characterize community ability and group-level multi-dimensional learning performance statistics, including the following components:

(1) **Community ability.** The community mean ability θ_{C_m} and the community ability level $c^{\theta}_{C_m}$ (aggregated from members' ability levels).

(2) **Overall community performance.** The overall community accuracy acc_{C_m}.

(3) **KC-level group performance.** For each knowledge concept k, we compute the group accuracy $\text{acc}_{C_m}(k)$ to form the community's concept-wise performance profile.

(4) **Question-level group performance overview.** From the question-attribute perspective, we provide a question-level overview of the community, including the group accuracy across difficulty levels $\text{acc}^{\text{diff}}_{C_m}(d)$ with $d \in \{\text{low}, \text{mid}, \text{high}\}$, and the group accuracy

across discrimination levels $\mathrm{acc}_{C_m}^{\mathrm{disc}}(t)$ with $t \in \{\mathrm{low}, \mathrm{mid}, \mathrm{high}\}$.

E Four-route Complementary Signals for Similar Peers Retrieval

The four complementary signals characterize the similarity of a student pair (u, v) within the community from the perspectives of ability, knowledge-concept performance consistency, shared questions, and shared knowledge coverage. We define the four signal functions as follows:

(1) **Ability-gap signal**:

$$s_{\mathrm{abl}}(u, v) = |\theta_u - \theta_v| \tag{26}$$

where θ_u denotes the ability value of student u.

(2) **KC performance gap signal**:

$$s_{\mathrm{kc}}(u, v) = \frac{1}{|\mathcal{K}_{uv}|} \sum_{k \in \mathcal{K}_{uv}} |\mathrm{acc}_u(k) - \mathrm{acc}_v(k)| \tag{27}$$

where $\mathcal{K}_{uv}$ denotes the set of shared knowledge concepts between u and v, and $\mathrm{acc}_u(k)$ denotes the correctness rate of student u on knowledge concept k.

(3) **Shared question signal**:

$$s_{\mathrm{q}}(u, v) = |\mathcal{Q}_u \cap \mathcal{Q}_v| \tag{28}$$

where $\mathcal{Q}_u$ denotes the set of questions attempted by student u.

(4) **Shared KC signal**:

$$s_{\mathrm{k}}(u, v) = |\mathcal{K}_u \cap \mathcal{K}_v| \tag{29}$$

where $\mathcal{K}_u$ denotes the set of knowledge concepts covered by student u.

These four signals induce four candidate rankings, where $s_{\mathrm{abl}}(u, v)$ and $s_{\mathrm{kc}}(u, v)$ are ranked in ascending order, while $s_{\mathrm{q}}(u, v)$ and $s_{\mathrm{k}}(u, v)$ are ranked in descending order. Notably, $s_{\mathrm{kc}}(u, v)$ participates in fusion only when $|\mathcal{K}_{uv}| > 0$ and the corresponding correctness rates are computable; otherwise, this route does not vote for the candidate pair (u, v).

F Datasets

We evaluate the performance of MicroC-KT on four commonly used public educational datasets. Tab. 5 reports the statistics of the processed datasets.

Datasets	Students	Questions	KCs	Interactions
Assistment09	3,013	9,795	107	297,575
Statics2011	331	633	97	111,468
DBE-KT22	1,140	212	93	306,499
Frcsub	536	20	8	98,624

Table 5: Statistics of the four processed datasets.

- **Assistment09** is collected from the ASSISTments online learning platform during the 2009–2010 academic year. The questions primarily focus on mathematics, and the dataset is widely used for benchmarking knowledge tracing models.
- **Statics2011** is collected from an Engineering Statics course taught at Carnegie Mellon University in Fall 2011. It records students' learning interactions in an engineering statics setting and has been commonly adopted in knowledge tracing literature.
- **DBE-KT22** is a knowledge tracing dataset collected from an online student exercise system in a database course taught at the Australian National University. It is designed to support tracking students' knowledge progress over time for online education tasks, and is released for public access through the Australian Data Archive platform.
- **Frcsub** mainly focuses on middle school students' responses to fraction subtraction questions, providing a compact yet representative benchmark for evaluating knowledge tracing models.

For all datasets, we remove students with fewer than 10 interaction records and questions that are answered fewer than 10 times.

G Baselines

To evaluate the performance of our proposed method, we compare it with four categories of baselines: DL-based, Graph-based, Hypergraph-based, and LLM-based methods.

G.1 DL-based KT Methods

The DL-based KT methods include:

- **DKT** uses recurrent neural networks to encode a student's response sequence and predict the correctness of the next interaction.

- **DKVMN** uses a key value memory network with static concept keys and dynamic values that update to track a student's mastery of each concept.
- **AKT** uses a monotonic, decay-based attention mechanism over past interactions, with Rasch-model regularization to improve KT performance.
- **SAINT+** is an encoder–decoder Transformer for knowledge tracing that separately models exercises and responses, and augments responses with elapsed-time and lag-time embeddings.
- **DTransformer** builds a diagnostic Transformer that infers knowledge-level proficiency from question-level mastery and is trained with a contrastive objective to produce more stable knowledge state tracing.

G.2 Graph-based KT Methods

The Graph-based KT methods include:

- **GKT** formulates knowledge tracing as a time-series node-level prediction task on a knowledge graph, using graph neural networks to model latent relationships among exercises.
- **GIKT** uses graph convolutional networks to propagate embeddings over the question–skill graph and model student, question and skill interactions for performance prediction.
- **PEBG+DKT** pretrains question embeddings from side information and then feeds these embeddings into DKT for improved knowledge tracing.
- **SimQE** learns similarity enhanced question embeddings by extracting and fusing multi-relation question similarity from weighted, attributed meta-paths to alleviate data sparsity in knowledge tracing.
- **STHKT** models dynamic structural dependencies in knowledge tracing by using a tripartite graph with graph convolutional networks and a Hawkes process spatiotemporal attention mechanism.
- **DyGKT** builds a continuous time dynamic question answering graph and uses dual time encoding to capture evolving student, question, and concept relations for knowledge tracing.

G.3 Hypergraph-based KT Methods

The Hypergraph-based KT methods include:

- **HyperKT** uses dual channel adaptive hypergraph encoders to capture higher order response relations and multigranularity knowledge states, with cross view contrastive learning for stronger supervision.
- **DGEKT** ensembles a hypergraph model for exercise concept associations and a directed graph model for interaction transitions, combined via online knowledge distillation for improved knowledge tracing.
- **HDKT** enhances question representations with a hypergraph over multi-attribute information and traces knowledge state evolution with session based forgetting and a de conjecture mechanism to reduce guessing bias.

G.4 LLM-based KT Methods

The LLM-based KT methods include:

- **EPFL** applies large language models to knowledge tracing via zero shot prompting and fine tuning, showing that fine tuned LLMs can model learning trajectories and reach performance comparable to Bayesian knowledge tracing.
- **EFKT** formulates explainable few shot knowledge tracing and uses large language models with cognition guided prompting to predict performance from limited records while generating natural language explanations.
- **LLM-KT** aligns large language models with knowledge tracing using plug and play instruction and plug in context and sequence adapters to inject compressed history and behavior representations for better prediction.
- **CIKT** uses an LLM to iteratively generate explainable student profiles and refine them through a collaborative loop with a predictor to improve knowledge tracing accuracy and interpretability.
- **HISE-KT** integrates a heterogeneous information network with an LLM that scores and filters meta-path instances to improve KT interpretability.

Model	Training Time (s)	Offline Time (s)	Inference Time (s)
	DL-based		
DKT	608.6	–	5.2
DKVMN	742.9	–	6.4
AKT	1189.7	–	18.6
SAINT+	1418.2	–	22.9
DTransformer	2196.5	–	31.7
	Graph-based		
GKT	987.4	–	12.8
GIKT	1562.8	–	16.9
PEBG+DKT	1719.6	–	14.6
SimQE	1637.3	–	15.8
STHKT	2574.9	–	20.7
DyGKT	2128.4	–	19.4
	Hypergraph-based		
HyperKT	3327.2	–	24.6
DGEKT	2711.9	–	21.8
HDKT	3588.5	–	27.9
	LLM-based		
EPFL	–	–	96.4
EFKT	0	–	134.7
LLM-KT	–	–	149.3
CIKT	–	–	128.6
HISE-KT	0	–	104.8
2T-KT	0	–	161.9
	Ours		
MicroC-KT$_{\text{Qwen-Plus}}$	0	244.1	111.6

Table 6: Efficiency comparison in terms of training time, offline construction time, and inference time.

- **2T-KT** uses an LLM with a teacher thinking prompt pipeline and augmented local and global knowledge graphs to predict student performance on exercises involving new knowledge concepts.

H More Implementation Details

We strictly enforce the separation of training and testing throughout the IRT estimation and hypergraph construction. Specifically, the item parameters of IRT-2PL (difficulty b_q and discrimination a_q) are estimated using only the training set and then kept fixed in subsequent stages; the hypergraph is also constructed solely from the training interactions. Meanwhile, we estimate student ability θ_u for training students based on the training data via IRT. During testing, for any target student u, at prediction time step t, we estimate the ability $\theta_{u,t}$ online using only the observed history in the test sequence (i.e., interactions up to time $t-1$), ensuring that ability estimation does not access future interactions or future labels.

For community membership matching, the textual channel employs the Sentence-BERT encoder `sentence-transformers/all-mpnet-base-v2` to obtain representations of the student-level and community-level natural language summaries, which are then used for subsequent matching.

I Efficiency and Computational Cost

Tab. 6 reports our efficiency evaluation results, including the training time, offline construction time, and inference time of all compared methods. All efficiency experiments were conducted under the same hardware and software environment, with all models implemented in `PyTorch` and evaluated using an identical data preprocessing and testing pipeline for fair comparison. Our server configuration is: a single RTX 5090 GPU (32GB), 25 vCPUs Intel Xeon Platinum 8470Q, and 90GB RAM. For trainable models (DL-based, graph-based, and hypergraph-based methods), we report the *total training time to convergence* on the Assistment09 dataset as *Training Time*. We define *Inference Time* as the *total time* required to complete one full evaluation pass on the test set. Since these trainable baselines also require certain preprocessing steps, they do incur a small amount of offline cost; however, this cost is typically negligible compared with the overall training time under our implementation and is therefore omitted, with *Offline Time* marked as "–".

For LLM-based methods, existing implementations vary substantially across papers. Some are purely prompting-based with no training, while others involve additional LLM fine-tuning. For prompting-based methods, we set *Training Time* to 0; however, their offline procedures are often implementation-specific and difficult to reproduce and align under a unified setting, so we do not report *Offline Time*. For fine-tuned methods, the training process and compute budget are hard to track consistently across implementations, so we mark *Training Time* as "–" and likewise do not report *Offline Time*. Therefore, for all LLM-based approaches, we report only *Inference Time*. Notably, the inference latency of LLM-based approaches is primarily influenced by the prompt length and structure, the API response rate, and the generated output length. As a result, the time differences among LLM-based methods typically reflect their relative prompting complexity and information volume, rather than exhibiting the same magnitude of scaling as training costs in deep models.

For our method MicroC-KT, since it is *training-free*, the *Training Time* is 0. We additionally report *Offline Time*, which covers one-time procedures including hypergraph construction, hypergraph spectral clustering, summary generation, peer retrieval, and so on. This offline stage is executed only once

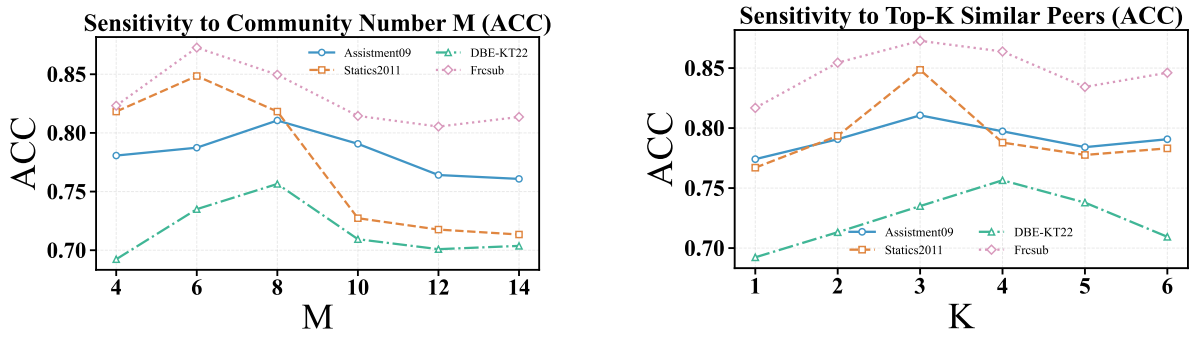

Figure 6: Parameter sensitivity of the community number M and the number of similar peers Top-K on ACC.

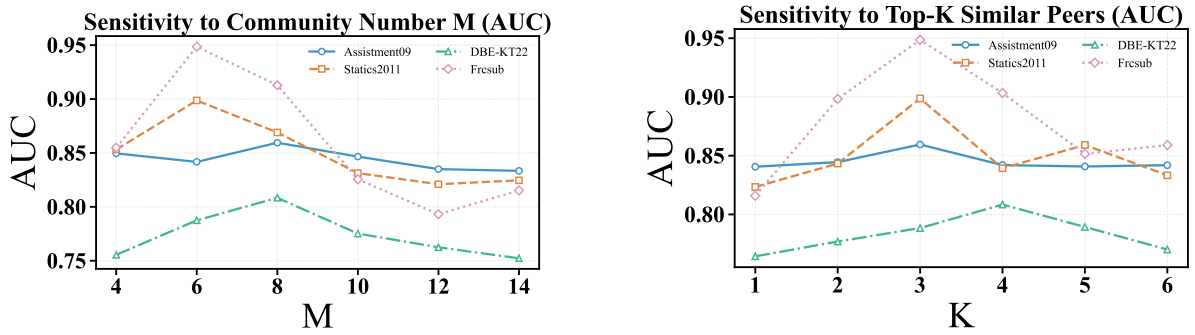

Figure 7: Parameter sensitivity of the community number M and the number of similar peers Top-K on AUC.

and can be reused during online prediction, making our approach deployment-friendly.

Trends and conclusions. Overall, the statistics indicate that the training cost of trainable models increases markedly with architectural complexity, and methods based on attention as well as graph or hypergraph modeling are more sensitive to compute resources. In contrast, our approach is training-free: it acquires structured priors via a one-time offline construction step while maintaining an acceptable online inference cost. In real-world applications, under a limited time budget, our method not only achieves substantial performance improvements but also produces human-readable interpretability reports, offering a more favorable balance between predictive performance and computational cost, and aligning well with resource-constrained and rapid-deployment educational scenarios.

J Parameter Sensitivity

The complete ACC and AUC results of parameter sensitivity are shown in Fig. 6 and Fig. 7.

K Clustering Analysis

The clustering analysis of all the datasets used in this paper is shown in the Fig. 8 (for Assistment09 and Statics2011 datasets) and Fig. 9 (for DBE-KT22 and Frcsub datasets).

Method	EG	FC	TA	PU	CC	Overall
EFKT	2.35	2.41	2.28	2.12	2.50	2.33
CIKT	3.08	3.12	3.01	2.86	3.22	3.06
HISE-KT	4.18	4.23	4.12	4.05	4.31	4.18
MicroC-KT$_{\text{Qwen-Plus}}$	4.82	4.86	4.79	4.74	4.90	4.82

Table 7: Human evaluation of analysis report quality on 40 sampled students (10 per dataset) across four datasets. We evaluate report quality from five dimensions (*Evidence Groundedness (EG)*, *Factual Consistency (FC)*, *Targeted Analysis (TA)*, *Pedagogical Usefulness (PU)*, and *Clarity and Coherence (CC)*). Each dimension is rated on a 1–5 scale (higher is better), and **Overall** is the average over five dimensions. Five experts independently rated all reports, with inter-rater agreement measured by the Cohen's kappa coefficient $\kappa^{\dagger} = 0.82$.

L Quality evaluation of The Analysis Report

To systematically evaluate the explanation quality of the analysis reports generated by LLM-based KT methods, we compare **EFKT**, **CIKT**, **HISE-KT**, and our **MicroC-KT**. We conduct a human evaluation on reports produced from the four datasets used in this paper. Specifically, we randomly sam-

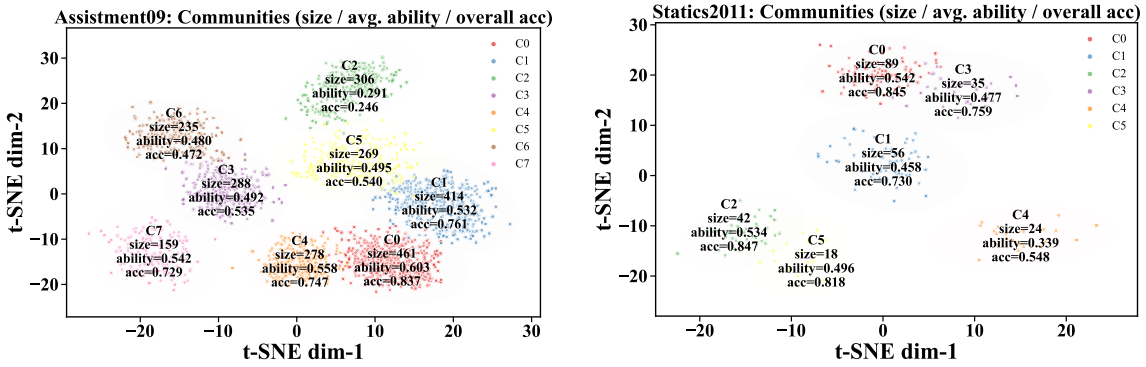

Figure 8: Visualizations of community clustering analysis on Assistment09 and Statics2011.

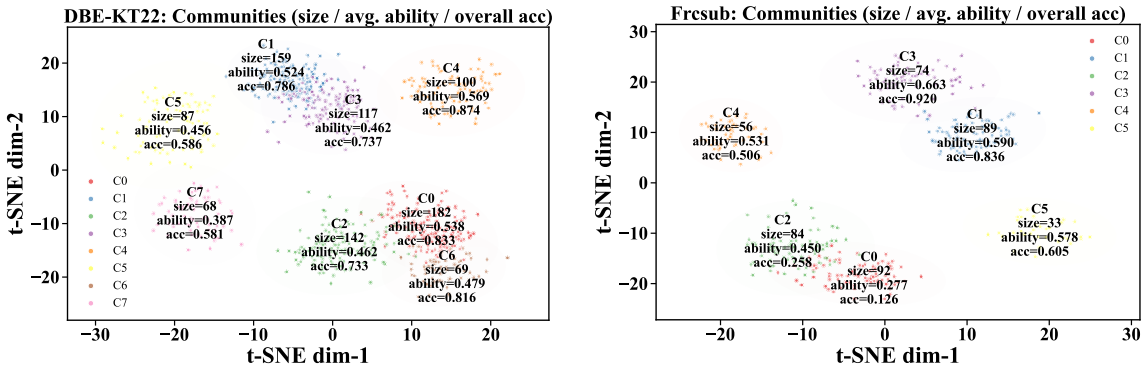

Figure 9: Visualizations of community clustering analysis on DBE-KT22 and Frcsub.

ple **10** students from each dataset, resulting in **40** students in total, and collect their corresponding analysis reports for all compared methods. We invite **five** computer science graduate students, who are trained under a unified rubric by a professor, to independently score the reports. To quantify inter-rater reliability, we report the **Cohen's kappa coefficient** (denoted as $\kappa^{\dagger}$) and obtain $\kappa^{\dagger} = 0.82$, indicating strong agreement among the experts.

L.1 Evaluation Criteria

We adopt a 5-point Likert scale (1-5; higher is better) and score each report from five dimensions:
(1) **Evidence Groundedness (EG)**, measuring whether the key claims and reasoning are clearly traceable to the provided evidence (e.g., the target student's history, community statistics, and in-community peer comparisons);
(2) **Factual Consistency (FC)**, assessing whether the report stays faithful to the given inputs without misreading, exaggeration, or unsupported additions, especially when referencing concept-level and group-level statistics;
(3) **Targeted Analysis (TA)**, evaluating whether the report is specific to the target student and the target question (and its related concepts or attributes), rather than generic statements;
(4) **Pedagogical Usefulness (PU)**, measuring whether the report provides actionable diagnostic conclusions and concrete learning suggestions (e.g., concept-level weaknesses and strategy-level guidance);
(5) **Clarity and Coherence (CC)**, assessing readability and logical organization, i.e., whether the report presents a coherent chain from evidence to analysis and then to conclusions or suggestions.
We additionally report an **Overall** score, computed as the average of the five dimensions.

L.2 Results and Discussion

Tab. 7 summarizes the human evaluation results. Overall, **EFKT** receives the lowest scores, suggesting that reports relying on sparse individual evidence tend to be less grounded and less actionable. **CIKT** improves over EFKT, indicating that iterative profile generation can enhance structure and consistency to some extent. **HISE-KT** further surpasses CIKT, benefiting from richer structured evidence organization. Finally, **MicroC-KT** achieves the best performance across all dimensions, with

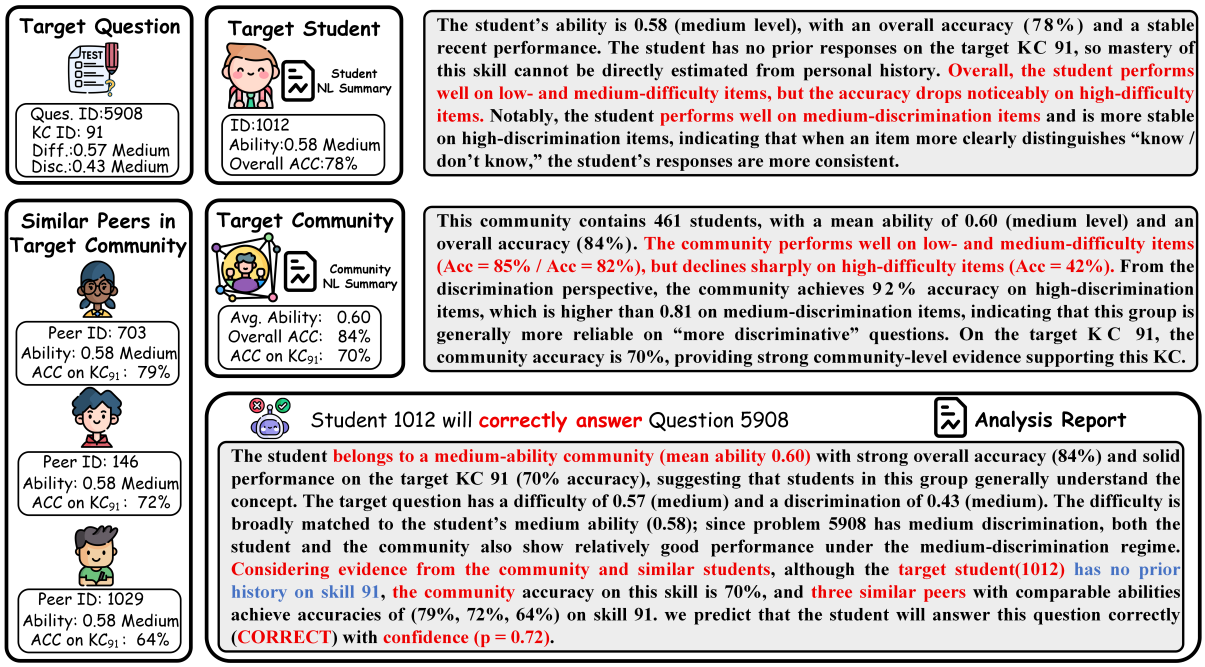

Figure 10: More details in case study. The figure shows the natural-language summary of the target student, the natural-language summary of the matched learning community, and the detailed analysis report generated from these evidence sources for KT prediction.

particularly clear advantages in evidence groundedness and factual consistency, demonstrating that learning micro-environment references and peer-contrast evidence provide more verifiable and stable support for LLM reasoning, thereby improving explanation reliability and pedagogical utility.

M Detailed Natural-Language Summaries in Case Study

In Fig. 10, we present the natural-language summaries of the target student and the matched learning community, together with a detailed analysis report generated based on these evidence sources to illustrate the inference process and final prediction.

N Prompt Templates for Summaries Generation

N.1 Prompt Template for Students' Natural Language Summaries Generation

Prompt templates for students' natural language summaries generation as shown in Fig. 11.

N.2 Prompt Template for Communities' Natural Language Summaries Generation

Prompt templates for communities' natural language summaries generation as shown in Fig. 12.

O Prompt Templates for KT Prediction and Analysis Report Generation

Prompt templates for KT prediction and analysis report generation as shown in Fig. 13 and Fig. 14.

System Message:
You are a rigorous educational data analysis assistant. Given historical statistics for a single student in JSON format before attempting the current target question, write a concise English natural-language summary.
Key rules:
1) Strictly use ONLY the facts provided in the JSON. Do NOT fabricate any new numbers, details, or conclusions.
2) If some fields are missing or marked as NA, ignore them and do not guess their values.
3) Output exactly ONE English paragraph of about 150 English characters (multiple sentences allowed). Do NOT use bullet lists, numbered lists, or headings.
4) Focus on the student's overall level and behavior patterns, rather than enumerating every skill ID one by one.
5) The summary will be used as context for a downstream knowledge tracing prediction.

Emphasize predictive structural cues such as ability level, overall accuracy, performance on the target skill and (if available) the target question, performance differences across difficulty and discrimination categories, and which skills are relatively strong / medium / weak in aggregate.

User Prompt Template:
Below are the historical statistics for a student before attempting the current target question in the test set (in JSON format). The fields are defined as follows:

· user_id: Student ID
· theta_norm: Normalized ability score in [0,1]; larger means stronger ability
· ability_level: Ability level (low/medium/high)
· acc_overall: Overall accuracy over historical answered events
· target_question_id: ID of the current target question
· target_skill_id: Main knowledge concept ID of the target question
· target_skill_acc: Historical accuracy on the target skill (history only)
· target_question_acc: If the student has answered the same target question in history, this is the historical accuracy; if never seen, it is missing/NA
· difficulty_norm / discrimination_norm: Normalized difficulty / discrimination of the target question
· difficulty_level / discrimination_level: Difficulty / discrimination level (low/medium/high)
· acc_by_diff_cat: Accuracy and number of events in difficulty categories (low/medium/high)
· acc_by_disc_cat: Accuracy and number of events in discrimination categories (low/medium/high)
· acc_by_skill: Accuracy and number of events for each skill (full set; do not enumerate one-by-one)

Based on these data, please write a ``student historical summary'' in English. Requirements:
· Global profile: describe the student's overall ability using theta_norm and ability_level, and report acc_overall.
· Target mastery: describe mastery of target_skill_id using target_skill_acc, and coarsely categorize it as strong / medium / weak (e.g., >0.6 is strong, around 0.5 is medium, <0.4 is weak). If target_question_acc exists, mention it briefly; if missing, ignore it.
· Difficulty/discrimination behavior: summarize performance differences across difficulty levels and discrimination levels in natural language, rather than listing raw numbers.
· Skill distribution: using acc_by_skill, summarize in aggregate which skills are relatively strong/average/weak, without enumerating every skill ID.
· Do not provide teaching suggestions; only describe observed learning state and behavior patterns useful for predicting the next response.
· Output exactly one English paragraph (no lists or bullet points).
· Keep the summary concise: output about 150 characters in a single English paragraph.
· Return the result strictly in JSON format: {``user_id'': <string>, ``summary'': <string>}.

=== STUDENT JSON START ===

{STUDENT_JSON}

=== STUDENT JSON END ===

Figure 11: Prompt template for students' natural language summaries generation.

System Message:
You are a rigorous educational data analysis assistant. Given statistical information of a learning community in JSON format, you must write a concise natural-language summary in English.
Key rules:
1) Strictly use ONLY the facts contained in the provided JSON. Do NOT fabricate any new numbers or conclusions.
2) If some fields are missing or marked as NA, simply ignore them and do not guess their values.
3) Output a single English paragraph of about 150 English characters. Multiple sentences are allowed, but do not use bullet lists, numbered lists, or headings.
4) The summary should highlight the overall level and behavioral patterns of the community, rather than enumerating every knowledge concept ID one by one.
5) The text will later be used as context for knowledge tracing prediction.

Please emphasize structural information that is useful for prediction, such as: overall ability level of the community, overall accuracy, differences in performance on easy/medium/hard questions and on questions with different discrimination levels, and which knowledge concepts are relatively strong or weak.

User Prompt Template:
Below is the statistical information of a learning community in JSON format. The fields have the following approximate meanings:

· cluster_id: ID of the community
· n_users: number of students in the community
· n_events: number of answer events after eventization
· cluster_theta_mean: mean IRT ability value
· cluster_theta_mean_norm: mean ability value after normalization to [0,1]
· cluster_ability_level: ability level (e.g., low/medium/high, indicating the overall ability is low/medium/high)
· cluster_acc_overall: overall accuracy of the community
· acc_by_diff_cat: accuracy and event count for each difficulty bucket (low/medium/high)
· acc_by_disc_cat: accuracy and event count for each discrimination bucket (low/medium/high)
· acc_by_skill: accuracy and event count for each knowledge concept (includes ALL concepts, no truncation)

Please read ALL the statistics carefully and write ONE English paragraph summarizing the community.
Requirements:
· First give an overall characterization of this community, including: number of students, overall ability level (explain the ability value and ability level in natural language; you do NOT need to mention the original raw IRT value; directly say the normalized ability is X and belongs to high/medium/low level), overall accuracy, and the main differences in performance on easy / medium / hard questions and on questions with different discrimination levels. You do NOT need to mention the total number of historical answer events.
· Use the full information in acc_by_skill to summarize in a high-level way which knowledge concepts are generally strong, which are average, and which are weak in this community. For example, you may roughly regard accuracy >0.60 as strong, around 0.50 as average, and <0.40 as weak. The goal is to capture the typical cognitive characteristics and learning patterns of the students in this community, not to list every concept ID explicitly.
· Do NOT give teaching suggestions; only describe phenomena and patterns. For example: ``the accuracy on hard questions is clearly lower than on easy questions'' or ``for some high-difficulty concepts, the error rate is relatively high''.
· Output exactly ONE paragraph of natural-language English text. Do NOT use bullet lists, numbered lists, or any headings.
· The paragraph should be about 150 English characters.
· Finally, return your answer in JSON format as: {``cluster_id'': <int>, ``summary'': <string>}, where summary is that single English paragraph.

=== COMMUNITY JSON START ===

{COMMUNITY_JSON}

=== COMMUNITY JSON END ===

Figure 12: Prompt template for communities' natural language summaries generation.

System Message:
You are an expert in knowledge tracing and educational data mining.
You will receive an integrated description that includes:
(1) the target student's profile and answering history on the TARGET knowledge concept set,
(2) the upcoming question and its difficulty and discrimination parameters,
(3) the assigned learning community (micro-environment) and its overall performance on the TARGET knowledge concept set and the upcoming question,
(4) a small set of similar students within the same community and their performance on the TARGET knowledge concept set and the upcoming question.

Based on the provided information, you must predict the probability that the student will answer the upcoming question correctly.
The probability MUST denote the likelihood of answering CORRECTLY:
0.00 means certainly wrong, and 1.00 means certainly correct.
Values greater than 0.5 indicate CORRECT; values less than 0.5 indicate WRONG.

You MUST follow the output format specified by the user.
Do not output anything other than the requested JSON.
If the user requests an explanation, append a short English explanation after the JSON and keep it within 4--6 sentences.

User Prompt Template:
The value range of student ability, question difficulty, and question discrimination is [0,1].

Information of the TARGET STUDENT:
· Student ID: U143.
· Student ability (normalized): 0.62 (higher means stronger ability).
· Overall accuracy on past records (before the upcoming question): 0.58.
· Accuracy on the TARGET KC set:
· Accuracy on TARGET KC (KC_12): 0.60.
· Accuracy on TARGET KC (KC_27): 0.45.

· Natural-language summary of the TARGET STUDENT (generated from statistics): The student shows medium-to-high normalized ability (0.62) with overall accuracy 0.58. Mastery is relatively stronger on KC_12 (0.60) but weaker on KC_27 (0.45), suggesting uneven concept-level performance. Errors occur more frequently when questions involve the harder concept, indicating potential vulnerability under higher difficulty .....

Information of the UPCOMING QUESTION:
· Question ID: Q8041.
· TARGET KC set: (KC_12, KC_27).
· Difficulty (normalized): 0.68 (higher means more difficult).
· Discrimination (normalized): 0.57 (higher means stronger discrimination).
· Difficulty level: high.
· Discrimination level: medium.

TARGET STUDENT's answering history on the TARGET KC set (earliest to latest):
· Question IDs: {Q1023, Q3310, Q1440, Q2197, Q5076, Q6102, Q7021, Q7705}.
· KC IDs per question (parentheses indicate multi-KC): {(KC_12), (KC_12,KC_27), (KC_27), (KC_12), (KC_27), (KC_12,KC_27), (KC_12), (KC_27)}.
· Difficulty (normalized): {0.42, 0.55, 0.63, 0.48, 0.70, 0.66, 0.52, 0.61}.
· Discrimination (normalized): {0.46, 0.53, 0.58, 0.49, 0.60, 0.57, 0.51, 0.55}.
· Answering results: {correct, wrong, wrong, correct, wrong, correct, correct, correct}.

Figure 13: Prompt templates for KT prediction and analysis report generation (1).

Information of the ASSIGNED LEARNING COMMUNITY (micro-environment):
· Community (cluster) ID: 17.
· Community ability mean (normalized): 0.55.
· Community ability level: medium.
· Community accuracy (overall): 0.56.
· Community accuracy on the TARGET KC set:
· Community accuracy on TARGET KC (KC_12): 0.61.
· Community accuracy on TARGET KC (KC_27): 0.48.
· Community accuracy on the upcoming question Q8041: 0.52.
· Natural-language summary of this COMMUNITY (generated from statistics): This community shows medium overall ability (0.55) with overall accuracy 0.56. Performance is relatively better on KC_12 (0.61) than on KC_27 (0.48), and accuracy tends to drop on higher-difficulty questions, indicating a consistent difficulty-sensitive pattern across members ... ...

Information of similar students within the SAME COMMUNITY:
We provide Top-K similar students, where K=3.
· Similar student 1:
· Similar student ID: S052.
· Ability (normalized): 0.64.
· Accuracy on TARGET KC (KC_12): 0.62; Accuracy on TARGET KC (KC_27): 0.50.
· Accuracy on the upcoming question Q8041 (historical): 0.55.

· Similar student 2:
· Similar student ID: S311.
· Ability (normalized): 0.59.
· Accuracy on TARGET KC (KC_12): 0.58; Accuracy on TARGET KC (KC_27): 0.46.
· Accuracy on the upcoming question Q8041 (historical): 0.50.

· Similar student 3:
· Similar student ID: S877.
· Ability (normalized): 0.57.
· Accuracy on TARGET KC (KC_12): 0.60; Accuracy on TARGET KC (KC_27): 0.44.
· Accuracy on the upcoming question Q8041 (historical): 0.49.

· Aggregate statistics of similar students:
· Average accuracy on TARGET KC (KC_12): 0.60.
· Average accuracy on TARGET KC (KC_27): 0.47.
· Average accuracy on the upcoming question Q8041: 0.51.

Guidelines for judging:
· Consider the target student's answering history on the TARGET KC set.
· Consider the student's overall accuracy and KC-specific accuracy on past records.
· Consider the IRT difficulty and discrimination of past questions and the upcoming question.
· Consider the community's overall level and its performance on the TARGET KC set and the upcoming question.
· Consider how similar students in this community perform on the TARGET KC set and the upcoming question.

Question: Will the learner answer question Q8041 correctly?
The probability means the chance of **CORRECT**.
· 0.00 is certainly wrong; 1.00 is certainly correct.
· Probability >0.5 means **CORRECT**; probability <0.5 means **WRONG**.

Respond ONLY with the following JSON:
{``prediction": ``CORRECT" or ``WRONG", ``probability": 0.00~1.00}
After the JSON, write a short English explanation in 4--6 sentences.
Your explanation must explicitly mention:
(1) the community the student belongs to and its overall characteristics,
(2) the community's performance on the TARGET KC set and the upcoming question,
(3) the performance of similar students on the TARGET KC set and the upcoming question,
(4) how the target student compares with the community and similar students,
(5) your final reasoning for the prediction.

Figure 14: Prompt templates for KT prediction and analysis report generation (2).