LearnerCoMPASS: Intelligent Tutoring System with Dynamic Cognitive Diagnosis and Multi-Model Path Planning

Ziji Sheng[12*], **Guiyao Tie**[12*], **Weidong Wang**[12], **Pan Zhou**[12†], **Daizong Liu**[3†]
[1]Hubei Key Laboratory of Distributed System Security,
Hubei Engineering Research Center on Big Data Security,
School of Cyber Science and Engineering, Huazhong University of Science and Technology
[2]Shenzhen Huazhong University of Science and Technology Research Institute
[3]Wuhan University
{ziji_sheng, tgy, M202472185, panzhou}@hust.edu.cn, daizongliu@whu.edu.cn

Abstract

Existing adaptive learning systems struggle to simultaneously achieve deep personalization, dynamic adaptability, and content trustworthiness, particularly in logically rigorous STEM fields where Large Language Models (LLMs) are prone to "hallucination". This paper introduces LearnerCoMPASS (Cognitive Multi-model Planning & Adaptive System), an integrated, end-to-end framework for adaptive learning. At its core, the framework features a novel multi-model path planning algorithm that orchestrates and fuses the outputs of heterogeneous LLM experts to generate and optimize learning sequences. To enable deep personalization, we design a dynamic cognitive diagnosis module that employs an innovative encoder-decoder architecture to generate precise, multi-dimensional cognitive state vectors for learners. To ensure trustworthiness, the system leverages an adaptively constructed dynamic knowledge graph and a Graph-RAG mechanism to provide factual anchors and logical constraints for LLM reasoning, thereby mitigating hallucinations. Extensive experiments demonstrate that LearnerCoMPASS significantly outperforms state-of-the-art baselines in generating high-quality personalized learning paths. Furthermore, ablation studies validate the critical contributions of our dynamic cognitive diagnosis and multi-model planning components.

1 Introduction

The realization of personalized education—the precise alignment of pedagogical trajectories with learner cognitive profiles—represents a cornerstone of modern educational technology (Shou et al., 2020; Tapalova and Zhiyenbayeva, 2022). Adaptive learning path recommendation (ALPR) systems are pivotal in this endeavor (Gligorea et al., 2023), orchestrating complex curricular structures into optimized sequences to maximize learning efficiency. Recently, the integration of Large Language Models (LLMs) (Fang et al., 2024b, 2023a, 2025; Tie et al.) has catalyzed a paradigm shift in Intelligent Tutoring Systems (ITS) (Wen et al., 2024), evolving from static, rule-based heuristics (Freedman et al., 2022) to dynamic, generative frameworks (Wang et al., 2025a; Jayasinghe and Kasthurirathna, 2022). This evolution (Figure 1) marks a significant leap toward autonomous, goal-oriented tutoring capable of addressing diverse and stochastic learning needs at scale.

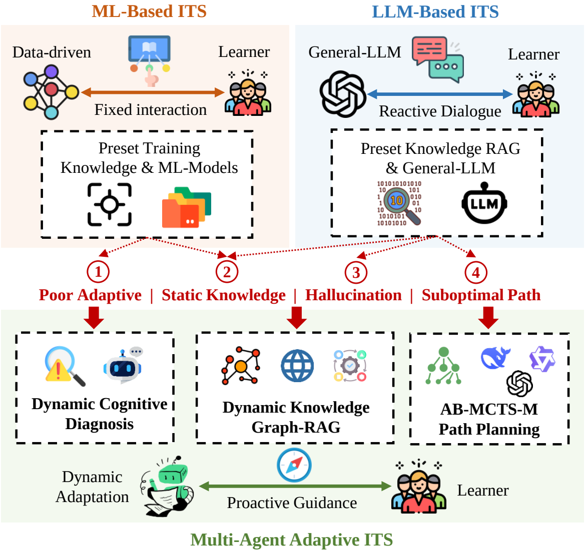

Figure 1: Comparison of three types of ITS Paradigms.

Despite this progress, current LLM-based educational planners encounter three critical bottlenecks. (1) *Diagnostic rigidity*: traditional neural cognitive diagnosis models (CDMs) (Yeung, 2019; Wang et al., 2022) rely on static ID-based embeddings. They suffer from "diagnostic lag" because they require expensive offline gradient updates, creating an "offline-training versus real-time inference" conflict that failing to track real-time proficiency shifts during active tutoring (Wang et al., 2020; Liu et al., 2023b; Lim et al., 2025). (2) *Factual unrelia-*

---

[*] Equal contribution
[†] Corresponding authors

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9016–9042
July 2-7, 2026 ©2026 Association for Computational Linguistics

bility: Generative LLMs function as probabilistic next-token predictors, often prioritizing semantic plausibility over factual correctness. This propensity for hallucination is pedagogically disastrous in logically rigorous STEM (Science, Technology, Engineering, and Mathematics) (Xie et al., 2015; Breiner et al., 2012) domains, which rely on strict logical chains and objective truths. Unlike general open-ended generation, a hallucinated formula or incorrect physical law generated by standard LLMs severely compromises pedagogical integrity without structured external grounding (Ho et al., 2024; Steinbach et al., 2025; Lu et al., 2025). (3) *Planning myopia*: standard decoding strategies often lack the long-horizon foresight required for complex curricula (Chen et al., 2024; Sun et al., 2025), leading to suboptimal or logically inconsistent learning sequences (Inoue et al., 2025).

To overcome these challenges, we propose LEARNERCOMPASS, a holistic framework for adaptive intelligent tutoring. The system operates through a multi-stage pipeline: (1) it constructs a *Hybrid Knowledge Graph* using Graph-RAG (Han et al., 2025a) to establish a verified pedagogical foundation; (2) it employs a *Multi-agent Encoder-Decoder* architecture to perform zero-shot, real-time cognitive diagnosis, mapping learner interactions to precise proficiency vectors; and (3) it finally executes a *Multi-model Path Planning* process driven by Adaptive Branching Monte Carlo Tree Search (AB-MCTS). This pipeline ensures a seamless interplay between structured knowledge reasoning, dynamic state estimation, and global trajectory optimization, providing a robust end-to-end solution for personalized learning. Our primary contributions are summarized as follows:

- **Integrated Graph-Grounded Framework:** We present LEARNERCOMPASS, an end-to-end architecture that synthesizes Graph-RAG with a Hybrid Knowledge Graph to provide factual grounding and mitigate LLM hallucinations.
- **Agentic Dynamic Diagnosis:** We introduce a Multi-agent Encoder-Decoder module that leverages agentic causal reasoning for real-time, retraining-free cognitive state estimation directly during inference.
- **Optimized Path Search:** We adapt the AB-MCTS algorithm for educational sequencing, utilizing **Bayesian Reward Estimation** to dynamically balance pedagogical exploration and trajectory refinement.

2 Related Work

Cognitive Diagnosis and Learner Modeling. Cognitive Diagnosis Models (CDMs) have evolved from traditional probabilistic models (Bu et al., 2023; Chen and de la Torre, 2013; Tu et al., 2011; Zhan et al., 2018) to deep learning (Yeung, 2019; Wang et al., 2022; Ma et al., 2022; Li et al., 2022) and graph-based architectures (Gao et al., 2021; Han et al., 2025b) to capture complex, non-linear student-exercise interactions. To bridge the "semantic gap" in behavioral data, recent works integrate Large Language Models (LLMs) (Liu et al., 2021b, 2020, 2024a,b, 2021a, 2022c,a,b, 2021c, 2023a, 2026b,a,c; Yan et al., 2025; Cai et al., 2025; Fang et al., 2023b,c, 2022, 2024a) to align semantic domain knowledge with latent proficiency (Tie et al., 2025b; Dong et al., 2025a,b; Chen et al., 2025). Despite these advances, existing CDMs predominantly rely on offline training, exhibiting a lack of adaptability required for tracking instantaneous cognitive shifts during active tutoring sessions (Liu et al., 2025; Tie et al., 2025a).

Adaptive Learning Path Planning. Adaptive path planning has transitioned from early collaborative filtering (Shou et al., 2020; Liao, 2025) to goal-driven, semantically aware recommendation systems (Jayasinghe and Kasthurirathna, 2022; Lv et al., 2025; Zhang et al., 2025a; Wang et al., 2025a; Lim et al., 2025). Current state-of-the-art approaches leverage Graph Retrieval-Augmented Generation (GraphRAG) (Han et al., 2025a; Larson and Truitt, 2024; Cheng et al., 2025; Xiang et al., 2025) to ensure logical coherence (Cheng et al., 2025) and multi-agent collaboration to generate dynamic lesson plans (Zhang et al., 2025a; Wang et al., 2025a; Lv et al., 2025). However, these architectures often face a trade-off between planning depth and computational efficiency, suffering from high inference latency (Xi et al., 2023; Jin et al., 2025; Andreychuk et al., 2025; Feng et al., 2025). While recent innovations in inference-time scaling and search algorithms (e.g., AB-MCTS and RL-based search) (Zhou et al., 2023; Lim et al., 2025; Inoue et al., 2025; Wang et al., 2025b) offer a pathway for long-horizon optimization, their application in rigorous STEM domains remains limited by the persistent risk of LLM hallucinations (Steinbach et al., 2025; Shi et al., 2024). Our work builds upon these foundations while introducing a novel agentic pipeline to address real-time diagnosis, logical grounding, and search efficiency.

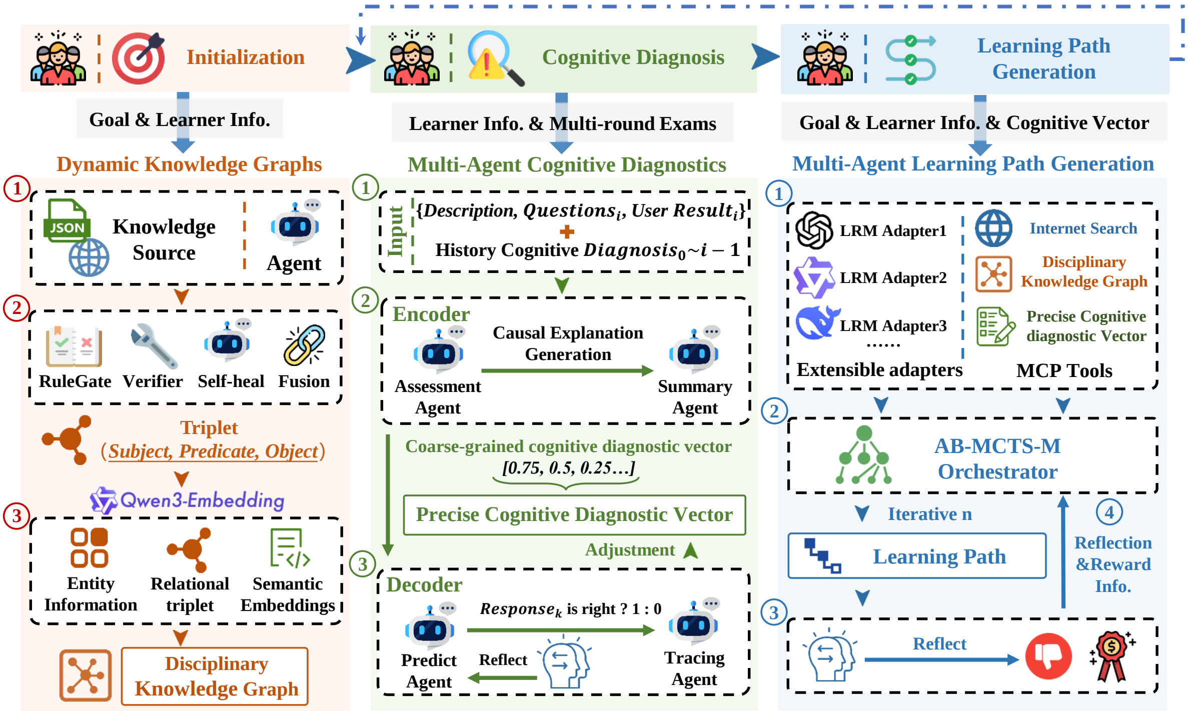

Figure 2: Overview of LEARNERCOMPASS, illustrating the iterative synergy between (1) **Dynamic Knowledge Graphs** for epistemic grounding, (2) **Multi-agent Cognitive Diagnostics** for latent state estimation, and (3) **AB-MCTS-M Path Generation** for trajectory optimization. See Appendix A.1-A.3 for details.

3 Methodology

LEARNERCOMPASS is an iterative, closed-loop framework designed to transform unstructured educational goals into optimized pedagogical trajectories. As illustrated in Figure 2, the methodology operates as a continuous cycle that refines the learner's path until the learning goal is achieved.

3.1 Knowledge-Grounded Reasoning via Hybrid KG and Graph-RAG

To mitigate the inherent "hallucination" risks of LLMs in rigorous STEM domains, we implement a Hybrid Knowledge Graph (KG) strategy as "Trusted Reasoning Engine". This approach synergizes a Static Macro-KG (G_{static}), which encodes high-level prerequisite dependencies from curated datasets (*e.g.*, Junyi (Chang et al., 2015)) to ensure curriculum coherence. To address the "cold start" problem for novel concepts, a Dynamic Micro-KG ($G_{dynamic}$) is constructed on-the-fly. When the system detects an uncovered domain, it triggers web searches to dynamically construct in-memory subgraphs as a knowledge compensator to capture granular conceptual nuances.

Adaptive Micro-KG Construction. As shown in the left panel of Figure 2, the construction of $G_{dynamic}$ follows a rigorous pipeline managed by a specialized *KGAgent*. Upon receiving a query, the agent retrieves multi-source documents and extracts candidate triples $\mathcal{T} = \{(h, r, t)\}$. To ensure pedagogical accuracy, we define a verification function $f_{verify}(t) \rightarrow \{0, 1\}$ that filters candidates through three strict constraints: (1) *Schema Compliance*, ensuring adherence to the defined (Subject, Predicate, Object) structure; (2) *Ontological Validity*, where the relation r belongs to a predefined pedagogical set (refer to Table 7); and (3) *Logical Consistency*, which explicitly detects and prunes cycles (e.g., $A \rightarrow B \rightarrow A$) that would disrupt learning progressions. Validated entities are subsequently processed by a local embedding model (e.g., Qwen (Zhang et al., 2025b)) and ingested into a graph database (Neo4j) to instantiate $G_{dynamic}$ for retrieval.

Graph-RAG Inference. We utilize a Graph-Retrieval-Augmented Generation (Graph-RAG) paradigm to ground instructional outputs. The system traverses G_{static} to determine the macro-path and retrieves context-rich subgraphs from $G_{dynamic}$ for each node. This structured context is injected into the LLM's prompt, formulating the pedagogical task as conditional inference $P(y \mid q, \boldsymbol{\theta}, \mathcal{G})$ to ensure that the output remains precise and factually consistent.

3.2 Cognitive Diagnosis Multi-Agent System

The Cognitive Diagnosis module formalizes learner modeling as a real-time *State Estimation* problem. Let $\boldsymbol{\theta} \in \mathbb{R}^d$ denote the proficiency vector across d domain concepts. The system minimizes the discrepancy between the predicted response $\hat{r}$ and the observed response r_{obs} through a continuous refinement loop:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\hat{r}(q, \boldsymbol{\theta}), r_{obs}), \qquad (1)$$

where q represents the test item and $\mathcal{L}$ captures both the prediction error and the semantic alignment of the reasoning process. To handle the high-dimensional nature of cognitive states, we implement a *Dynamic Synergy Mechanism*: the Encoder provides a rapid initialization for novel concepts, while the Decoder performs fine-grained, iterative tuning for recurring concepts.

Encoder: Coarse-Grained Diagnosis. The Encoder functions as the perception layer, aggregating multimodal interaction data $\mathcal{M}$ and historical context $\mathcal{H}$ to generate a robust initialization without extensive training. This process is driven by the *Assessment Agent* ($\mathcal{A}_{assess}$), which employs a *Bidirectional Causal Inference* mechanism. Upon receiving $x = (q, r_{obs})$, it performs *Forward Inference* via Graph-RAG to identify prerequisite knowledge components (KCs) and *Backward Verification* to attribute the observed error to specific conceptual gaps rather than stochastic noise (*e.g.*, calculation slips). Subsequently, the *Summary Agent* ($\mathcal{A}_{sum}$) synthesizes these causal attributions and behavioral metadata into a natural-language summary S. This summary is mapped to a normalized proficiency vector:

$$\boldsymbol{\theta}_{coarse} = \text{Map}\big(\mathcal{A}_{sum}(O_{assess}, \mathcal{M}, \mathcal{H})\big), \qquad (2)$$

where the mastery status of each concept is projected into a discrete-to-continuous range $[0.00, 1.00]$ based on five predefined pedagogical levels: Not Mastered (0.00), Insufficient Mastery (0.25), Moderately Mastered (0.50), Well Mastered (0.75), and Fully Mastered (1.00). This initialization serves as a strong epistemic prior for subsequent refinement.

Decoder: Fine-Grained Tuning. The Decoder refines $\boldsymbol{\theta}_{coarse}$ into a precise vector $\boldsymbol{\theta}_{fine}$ through a *"Predict-Reflect-Update"* cycle, simulating a semantic gradient descent. The *Predict Agent* ($\mathcal{A}_{pred}$) acts as a cognitive simulator, formulating the forward pass to estimate the response probability:

$$\hat{r}_k = \mathcal{A}_{pred}(\boldsymbol{\theta}_k, q, \mathcal{H}). \qquad (3)$$

If a significant deviation between $\hat{r}_k$ and r_{obs} is detected, the *Tracing Agent* ($\mathcal{A}_{trace}$) performs *Deviation Analysis* to locate the root cause. It evaluates the error against the structural constraints of the Macro-KG and generates a gradient-like adjustment rationale. The state is then iteratively updated via:

$$\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k + \eta \cdot \Psi\big(\mathcal{A}_{trace}(\hat{r}_k, r_{obs}, \boldsymbol{\theta}_k)\big), \qquad (4)$$

where Ψ maps the linguistic rationale into a vector update direction and η denotes the adaptive learning rate. This dual-process mechanism allows the system to zero in on the learner's true knowledge boundary with sub-skill precision and full interpretability.

3.3 Multi-Model Path Planning Engine

The Path Planning Engine serves as the system's executive core, synthesizing the cognitive state $\boldsymbol{\theta}$ and the knowledge graph $\mathcal{G}$ into an optimal learning trajectory. To overcome the reasoning limitations of singular LLMs—such as restricted search horizons and lack of diversity—we design a multi-model orchestration strategy as illustrated in the right panel of Figure 2. This engine dynamically navigates the vast search space of pedagogical strategies by fusing outputs from heterogeneous agents.

Heterogeneous Expert Ensemble. Inspired by the Mixture-of-Experts (MoE) paradigm, our architecture encapsulates diverse LLM backbones (*e.g.*, GPT-4) into standardized Expert Adapters (LRM Adapters). Each adapter is prompted with a distinct pedagogical persona—such as "Socratic Tutor," "Curriculum Specialist," or "Domain Practitioner"—to induce diverse reasoning patterns. Furthermore, these adapters are equipped with MCP Tools, including Internet Search and Disciplinary KG access, to provide factual grounding. This ensemble approach mitigates the inherent biases of individual models and ensures that the generated paths are pedagogically robust across varied STEM disciplines.

Path Fusion via AB-MCTS-M. We adapt the Adaptive Branching MCTS (AB-MCTS-M) algorithm for educational sequencing, treating path generation as a dynamic decision process. Unlike standard MCTS with fixed branching factors, AB-MCTS-M introduces an adaptive mechanism to balance *Exploration* (Parallel Generation via experts) and *Exploitation* (Deepening existing chains). We model the potential reward $R(v_i)$ of a node v_i using a Bayesian Hierarchical Mixed Model:

$$R(v_i) \sim \mathcal{N}(\mu + \alpha_{g(v_i)}, \sigma^2), \quad \alpha_g \sim \mathcal{N}(0, \tau^2), \tag{5}$$

where μ represents the global average quality, $\alpha_{g(v_i)}$ is the random effect associated with the specific expert group g that generated node v_i, and σ^2 captures the aleatoric uncertainty. At each step, the engine employs Thompson Sampling to select the optimal expansion action a_t^*:

$$a_t^* = \underset{a \in \mathcal{A}(s_t) \cup \{\text{GEN}\}}{\arg\max} \mathbb{E}_{\text{posterior}}[R(a) \mid \mathcal{H}_t]. \tag{6}$$

Selection of the virtual GEN node triggers the expert ensemble to spawn new candidate path segments; otherwise, the orchestrator deepens and elaborates upon the most promising existing node.

Reflexion and Policy Improvement. To enable long-term evolution without expensive parameter updates, we incorporate a Reflexion mechanism. Upon path completion, a dedicated Reward Agent evaluates the trajectory against the learner's goal and diagnostic vector, generating a verbal critique (*e.g.*, The transition to advanced calculus was too abrupt). This feedback is stored in a vector-indexed memory $\mathcal{K}_{mem}$. For future planning tasks, relevant reflections are retrieved and injected into the Expert Adapters' context as zero-shot prompts. This iterative loop allows the engine to refine its planning logic and pedagogical strategies based on historical successes and failures, ensuring the system dynamically adapts to the learner's evolving trajectory.

4 Experiments

4.1 Experimental Setup

Datasets. We utilized three distinct datasets to assess performance across varying degrees of structure and domain specificity:

- **AI-Generated Skill Metadata Dataset**[1]: This dataset contains 200 skill-prerequisite pairs. We treat the `skill_name` as input and the corresponding `prerequisites` (0-3) as the ground truth. It evaluates the system's ability to construct accurate static knowledge dependencies.
- **Resume Dataset**[2]: We constructed a mapping task using 200 samples across 5 distinct career categories. Resumes simulate the learner's initial state (S_0), while job descriptions from a hold-out set serve as learning goals (G). The "skills required" field in job postings is treated as the ground truth for necessary learning nodes.
- **STEM Learner Path Planning**: We constructed a proprietary dataset containing 317 high-complexity learning goals spanning Advanced Mathematics, Linear Algebra, and University Physics. This dataset specifically challenges the system's ability to plan long-horizon, logically rigorous trajectories for STEM subjects.

Baselines. We compared our framework against a range of strong LLM-based baselines, categorized by their interaction paradigm:

- **Direct Prompting (Dir Prompt)** serves as the zero-shot configuration, utilizing GPT-4o (OpenAI et al., 2024), Qwen3-235B (Qwen, 2025), and Llama-3.3-70B-Instruct[3] (meta llama, 2025).
- **Chain-of-Thought (CoT)** enhances this baseline by explicitly requesting intermediate reasoning steps for GPT-4o and Qwen3-235B.
- **LearnerCoMPASS-full** instantiates our complete integrated engine, incorporating the AB-MCTS-M and Hybrid KG, with the aforementioned LLMs serving as backbone models.

4.2 Evaluation Metrics

Given the generative nature of the tasks, rigid string matching is insufficient. We employ **LLM-based Semantic Evaluation** (using GPT-5.2 (OpenAI, 2025) as the judge) to assess performance.

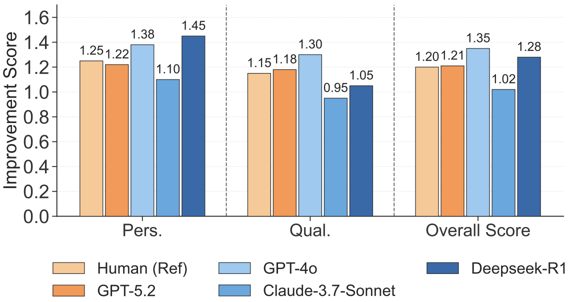

Figure 3: Consistency analysis of improvement scores across different evaluators, where GPT-5.2 exhibits the highest alignment with human experts.

---

[1] https://www.kaggle.com/datasets/saicharan1206/ai-generated-skill-metadata-dataset-1800

[2] https://www.kaggle.com/datasets/gauravduttakiit/resume-dataset

[3] Specific checkpoints used include Qwen3-235B-A22B-Instruct-2507 and Llama-3.3-70B-Instruct, denoted as Qwen3 and Llama3.3 in result tables.

Table 1: Comparative performance across three datasets. Extraction metrics and alignment scores are reported as percentages (%) and on a 5-point Likert scale. **Bold** and underline indicate the best and second-best performance.

Model	**AI-Generated Skill Metadata**					**Resume Dataset (Job Matching)**					**STEM Path Planning**		
	Extraction (%)		Alignment Score (1-5)			Extraction (%)		Alignment Score (1-5)			Alignment Score (1-5)		
	Rec.	Prec.	Pers.	Qual.	OA.	Rec.	Prec.	Pers.	Qual.	OA.	Pers.	Qual.	OA.
Llama3.3 Dir Prompt	44.79	46.07	3.15	3.38	3.27	31.75	39.17	3.12	3.25	3.19	2.85	3.05	2.95
Qwen3 Dir Prompt	43.08	48.88	3.32	3.55	3.44	37.79	42.23	3.25	3.42	3.34	3.02	3.28	3.15
GPT-4o Dir Prompt	46.38	50.12	3.45	3.62	3.54	41.67	44.47	3.38	3.51	3.45	3.15	3.42	3.29
Qwen3 CoT	53.33	55.59	3.74	3.96	3.85	42.87	48.58	3.68	3.85	3.77	3.35	3.65	3.50
GPT-4o CoT	54.92	58.39	3.82	4.05	3.94	44.62	52.24	3.75	3.92	3.84	3.48	3.78	3.63
Ours (Llama3.3)	54.75	57.72	4.35	4.42	4.39	45.04	50.52	4.25	4.35	4.30	4.12	4.63	4.38
Ours (Qwen3)	**62.21**	**64.00**	4.68	4.75	4.72	47.62	55.88	**4.65**	4.65	4.65	4.45	4.60	4.53
Ours (GPT-4o)	**62.21**	63.52	**4.75**	**4.82**	**4.79**	**55.96**	**57.05**	4.60	**4.74**	**4.67**	**4.58**	**4.75**	**4.67**

Judge Model Selection. To ensure the validity of automated scoring, we benchmarked improvement scores ($S_{\text{Ours}} - S_{\text{Baseline}}$) against human expert annotations across 100 sample batches. Our analysis compared GPT-5.2 with several state-of-the-art models, including GPT-4o, DeepSeek-R1, and Claude-3.7-Sonnet. As visualized in Figure 3, while all evaluators consistently identified the system's advantages, GPT-5.2 demonstrated the highest correlation with human scoring patterns.

Extraction Metrics. For the Skill and Resume datasets, we utilize *Semantic Precision*, which measures the proportion of generated path nodes that are semantically equivalent to ground-truth skills, and *Semantic Recall*, which assesses the extent to which ground-truth skills are covered by the generated trajectory. These metrics ensure that the constructed knowledge dependencies are both accurate and comprehensive.

Path Quality Metrics (Alignment Score). Beyond node-level extraction, we evaluate the holistic pedagogical quality of generated paths on a **5-point Likert scale** (Joshi et al., 2015) across two dimensions. (1) **Personalized Adaptability (Pers.)** assesses the system's responsiveness to individual learner profiles, aggregating sub-metrics of user identity match, goal alignment, time feasibility, and preference accommodation. (2) **Path Quality (Qual.)** evaluates the internal consistency of the sequence, including logical progression, completeness, practicality, and difficulty appropriateness. (3) **Overall (OA.)** represent the mean ratings across all test samples.

4.3 Main Results

In this section, we present the quantitative and qualitative evaluation results of our proposed framework across various dimensions.

4.3.1 Knowledge Mapping and Skill Extraction

We evaluate the framework's efficacy in identifying conceptual gaps and prerequisite dependencies. As illustrated in the left and middle sections of Table 1, LEARNERCOMPASS demonstrates significant gains in precision and recall across varied domains. On the **AI-Generated Skill Metadata Dataset**, our framework (Qwen3-235B) achieves a Recall of 62.21% and a Precision of 64.00%, substantially exceeding the GPT-4o CoT baseline. This performance gain suggests that the Hybrid KG provides essential structural constraints that mitigate the "omission bias" prevalent in standard LLMs regarding implicit prerequisites. Interestingly, while Ours (GPT-4o) and Ours (Qwen3) yields peak Recall (62.21%), Ours (Qwen3) exhibits superior Precision, indicating that the Qwen backbone is more conservative and effective at minimizing false-positive prerequisite assignments in technical domains. Regarding the **Resume Dataset**, our method facilitates high-fidelity mapping of user profiles to pedagogical requirements. While direct prompting suffers from low precision (≈40-45%) due to the hallucination of generic competencies, our *Trusted Reasoning Engine* functions as a latent filter, elevating Precision to 57.05% (GPT-4o). The consistent delta between "Ours" and respective "CoT" baselines confirms that our agentic workflow provides a superior epistemic foundation compared to linear prompt engineering.

4.3.2 Path Planning Quality and Alignment

The system's utility is anchored in generating rigorous, actionable learning trajectories, quantified by

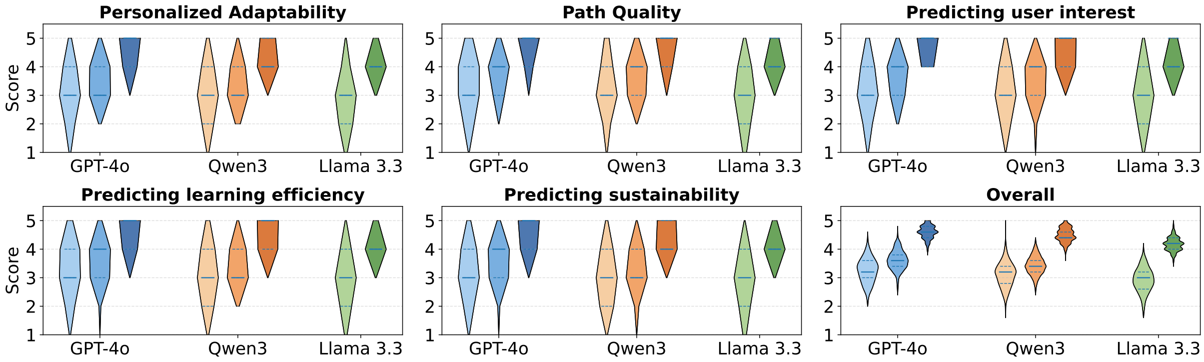

Figure 4: Evaluation scores (1–5) across six learner-centered metrics. Models are grouped by backbone (GPT-4o, Qwen3, and Llama 3.3). Within each group, variants are ordered from left to right as Dir Prompt, CoT, and LearnerCoMPASS (Llama 3.3 does not include a CoT variant). Violin plots depict score distributions; dashed lines indicate the first and third quartiles, while solid lines denote the median. Results show that LearnerCoMPASS consistently outperforms both Dir Prompt and CoT across all evaluated dimensions, with CoT providing moderate gains over direct prompting.

the **Alignment Score (1-5)**. Results on the STEM Learner Path Planning (Table 1, right) underscore the architectural advantages of our engine.

Superior Personalized Adaptability. Standard LLM planners, even with CoT, struggle with context-sensitive adaptation, typically scoring below 3.5/5. Conversely, LearnerCoMPASS (Qwen3) achieves **4.45**. This demonstrates that the *Summary Agent* in our Encoder correctly captures user metadata and that the AB-MCTS-M engine effectively explores the search space to find strategies that fit the user's constraints.

Enhanced Path Quality via Logic Constraints. In STEM fields, logical progression is non-negotiable. Our system achieves a Path Quality score of **4.60** (Qwen3) and **4.75** (GPT-4o), representing "Very Good" to "Excellent" pedagogical logic. This validates the efficacy of our *Graph-RAG* mechanism, which grounds the LLM's reasoning in the Static Macro-KG. The significant gap between CoT ($\approx$3.6-3.8) and our method suggests that while CoT improves local reasoning, it lacks the global lookahead capability provided by AB-MCTS-M.

Robustness Across Backbones. LearnerCoMPASS utilizing the smaller **Llama-3.3** backbone (Overall: 4.38) outperforms the significantly larger GPT-4o CoT (3.63). This suggests that the **Agentic Architecture** is a more decisive factor for performance than raw parameter count. Furthermore, the peak performance of **Ours (GPT-4o)** (4.67) demonstrates that the framework is model-agnostic and scales effectively with stronger base models to achieve near-human expert performance.

Table 2: Comparison of Cognitive Diagnosis Performance against traditional and neural baselines.

Model	**Accuracy ↑**	**AUC ↑**
Dataset 1: Public (Eedi)		
DINA (de la Torre, 2009)	0.438	0.582
IRT (Johns et al., 2006)	0.496	0.643
MIRT (Bergner et al., 2022)	0.491	0.615
NeuralCDM (Wang et al., 2020)	0.640	0.762
LearnerCoMPASS (Ours)	**0.678**	**0.844**
Dataset 2: Private (Linear Algebra)		
DINA (de la Torre, 2009)	0.333	0.532
IRT (Johns et al., 2006)	0.510	0.624
MIRT (Bergner et al., 2022)	0.608	0.688
NeuralCDM (Wang et al., 2020)	0.745	0.795
LearnerCoMPASS (Ours)	**0.822**	**0.841**

Learner-Centered Metrics Evaluation. To quantify real-world impact, we evaluate three novel dimensions: **Learning Interest** (assessing the path's potential to stimulate engagement), **Learning Efficiency** (measuring the optimal use of learning time), and **Skill Sustainability** (evaluating the long-term retention potential of the acquired knowledge). Figure 4 reveals while CoT offers moderate improvements over direct prompting, it exhibits high variance (wide quartiles), indicating unreliability. In contrast, LearnerCoMPASS maintains stable median scores >4.5 with significantly narrower interquartile ranges. This reduced variance underscores the robustness of our framework in generating learning paths that are not only mathematically rigorous but also consistently engaging and efficient across varied learner profiles.

Table 3: Ablation Study on Path Generation Quality and Hallucination Rate. Metrics are reported on a 5-point Likert scale (1-5), consistent with the main results. "Conflict Rate" and "Baseless Info" are reported as percentages.

System Variants	**Automated Evaluation Metrics (1-5)**			**Reliability & Safety**		
	Pers. Adaptability	**Path Quality**	**Overall Score**	**Conflict Rate**	**Baseless Info.**	**Hallucination**
Base LLM (No RAG, No Reflexion)	3.15	3.42	3.29	8.2%	23.3%	31.5%
w/ Reflexion ONLY	3.82	3.88	3.85	7.5%	21.4%	28.9%
w/ Graph-RAG ONLY	3.95	4.25	4.10	2.1%	8.5%	10.6%
w/ Graph-RAG + Reflexion	3.98	4.31	4.15	1.8%	6.2%	8.0%
Full System (LearnerCoMPASS)	**4.58** ± 0.26	**4.75** ± 0.22	**4.67** ± 0.24	**1.3%**	**2.8%**	**4.1%**
Human Eval. (Ref)	4.85	4.90	4.88	–	–	–

Table 4: Detailed Breakdown of Hallucination Types across 1,000 samples. **"ENG."** refers to our Trusted Reasoning Engine.

Module	**Hallucination Type**	**w/o ENG.**	**w/ ENG.**
Cognitive Diagnosis	Evident Conflict	32	2
	Subtle Conflict	28	7
	Evident Baseless Info.	73	4
	Subtle Baseless Info.	55	15
Path Planning	Evident Conflict	8	0
	Subtle Conflict	14	4
	Evident Baseless Info.	26	0
	Subtle Baseless Info.	79	9
Total Hallucination Count		**315**	**41**
Hallucination Rate		**31.5%**	**4.1%**

4.4 Modular Evaluation

To validate the effectiveness of our core components, we conduct a comprehensive modular evaluation. We isolate the contributions of our dynamic cognitive diagnosis and Trusted Reasoning Engine, demonstrating how they synergistically enhance both pedagogical accuracy and system reliability.

4.4.1 Performance on Cognitive Diagnosis

Table 2 presents the comparative results on both the Eedi (Wang et al., 2021) and Linear Algebra datasets. LearnerCoMPASS consistently outperforms all baselines. On the **Eedi dataset**, our method achieves an ACC of 0.678, surpassing the strongest neural baseline (NeuralCDM) by a significant margin (+3.8%). This improvement is attributed to our Multi-Agent Encoder-Decoder architecture, where the *Analyzer Agent* incorporates behavioral metadata (e.g., hesitation) that traditional models ignore. On the **private Linear Algebra dataset**, the advantage is even more pronounced (Accuracy 0.822 vs. NeuralCDM 0.745). This domain requires deep conceptual reasoning rather than simple pattern matching. Our *Tracer Agent* effectively maps student errors back to foundational misconceptions via the Knowledge Graph, leading to a more precise estimation of the cognitive state vector $\boldsymbol{\theta}$.

4.4.2 Impact on Reliability and Hallucination Mitigation

A critical challenge in applying LLMs to education is trustworthiness. We conducted an ablation study to analyze how our Trusted Reasoning Engine (Hybrid KG + Verification) mitigates hallucinations. **Measurement Method.** To rigorously quantify hallucination rates, we adapted the RAGTruth (Wu et al., 2023) methodology. We engaged human experts to annotate a random subset of 1,000 generated outputs. Each output was segmented into atomic claims (e.g., "Matrix multiplication is not commutative"), and each claim was verified against the ground truth in our Knowledge Graph. An output is flagged as hallucinated if it contains at least one unsupported or contradictory claim. As shown in Table 3, the Base LLM suffers from a high Hallucination Rate (31.5%) and frequently generates Conflict Information (8.2%). The introduction of Graph-RAG significantly reduces hallucinations to 10.6% by grounding generation in the static KG. The full system, incorporating the *Adjudicator Agent*'s verification loop, achieves a minimal hallucination rate of 4.1%, ensuring the pedagogical safety of the system.

4.4.3 Breakdown of Hallucination Types

Table 4 provides a granular analysis of hallucinations across 1,000 generated samples. We categorized errors into "Conflict" (contradicting domain facts) and "Baseless Info" (inventing non-existent concepts), further split by severity (Evident vs. Subtle). The results demonstrate that our Trusted Reasoning Engine is particularly effective at eliminating "Evident Conflicts" (reduced from 32 to 2 in Diagnosis, and 8 to 0 in Planning). While "Subtle Baseless Info" (e.g., plausible but incorrect minor details) remains the hardest to eliminate, our engine still reduces it by over 70% (from 55 to 15). This confirms that the constraint-based verification mechanism (detailed in Sec. 3.1) effectively filters out inaccurate pedagogical content.

5 Conclusion

In this paper, we introduced LEARNERCOMPASS, an end-to-end framework that addresses the critical balance between personalization, adaptability, and trustworthiness in intelligent tutoring. By synthesizing a Multi-agent Encoder-Decoder architecture for real-time, retraining-free cognitive diagnosis with the AB-MCTS-M algorithm for long-horizon path planning, we enable dynamic tutoring that evolves seamlessly with the learner's progress. Crucially, the integration of a Hybrid Knowledge Graph and Graph-RAG mechanism imposes rigorous logical constraints on LLM reasoning, reducing hallucination rates from 31.5% to 4.1% in complex STEM domains. Extensive evaluations on both public and proprietary datasets demonstrate that LEARNERCOMPASS significantly outperforms traditional neural baselines and aligns closely with human expert judgment. This work marks a paradigm shift from static, heuristic-based tutoring to dynamic, agentic educational systems, providing a robust and scalable foundation for trustworthy AI in high-stakes learning environments.

Limitations

Despite the promising results, our framework exhibits several limitations that merit further investigation. First, **Inference Latency and Computational Cost**: The deployment of the AB-MCTS-M engine involves recursive calls to multiple LLM agents and dynamic graph traversals. While this ensures high-quality reasoning, it introduces significant inference latency compared to direct prompting methods, potentially hindering real-time responsiveness in large-scale, low-latency educational scenarios (a detailed quantitative analysis of this computational overhead and efficiency-effectiveness trade-off is provided in Appendix A.4). Future work will explore distilling the reasoning capabilities of the agent ensemble into smaller, more efficient student models. Second, **Dependence on Knowledge Graph Quality**: The system's trustworthiness heavily relies on the completeness and accuracy of the underlying Static Macro-KG. In domains where high-quality structured knowledge is scarce or rapidly evolving (e.g., cutting-edge research fields), the "cold start" problem for KG construction may limit the effectiveness of our Graph-RAG mechanism. Finally, **Ecological Validity of Evaluation**: While we employed rigorous human and LLM-as-a-Judge evaluations alongside learner-centered metrics, our experiments were primarily conducted in simulated environments or on static datasets. The long-term impact of LEARNERCOMPASS on actual student learning outcomes—such as retention rates and skill transfer in real-world classroom settings—remains to be validated through longitudinal A/B testing.

Acknowledgments

This work is supported by Shenzhen Science and Technology Program JCYJ20240813153309013 and Wuhan Science and Technology Programme No.2025050602030058.

References

Anton Andreychuk, Konstantin Yakovlev, Aleksandr Panov, and Alexey Skrynnik. 2025. Mapf-gpt: Imitation learning for multi-agent pathfinding at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23126–23134.

Yoav Bergner, Peter Halpin, and Jill-Jênn Vie. 2022. Multidimensional item response theory in the style of collaborative filtering. *Psychometrika*, 87(1):266–288.

Jonathan M Breiner, Shelly Sheats Harkness, Carla C Johnson, and Catherine M Koehler. 2012. What is stem? a discussion about conceptions of stem in education and partnerships. *School science and mathematics*, 112(1):3–11.

Chenyang Bu, Zhiyong Cao, Chenlong He, and Yuhong Zhang. 2023. Probabilistic model with evolutionary optimization for cognitive diagnosis. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '23, page 891–899, New York, NY, USA. Association for Computing Machinery.

Xiaowen Cai, Daizong Liu, Xiaoye Qu, Xiang Fang, Jianfeng Dong, Keke Tang, Pan Zhou, Lichao Sun, and Wei Hu. 2025. Towards building model/prompt-transferable attackers against large vision-language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

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

Jinsong Chen and Jimmy de la Torre. 2013. A general cognitive diagnosis model for expert-defined polytomous attributes. *Applied Psychological Measurement*, 37(6):419–437.

Xin Chen, Jin Zhang, Tong Zhou, and Feng Zhang. 2025. Llm-cdm: A large language model enhanced cognitive diagnosis for intelligent education. *Ieee Access*.

Yanan Chen, Ali Pesaranghader, Tanmana Sadhu, and Dong Hoon Yi. 2024. Can we rely on llm agents to draft long-horizon plans? let's take travelplanner as an example. *arXiv preprint arXiv:2408.06318*.

Xinghe Cheng, Zihan Zhang, Jiapu Wang, Liangda Fang, Chaobo He, Quanlong Guan, Shirui Pan, and Weiqi Luo. 2025. GraphRAG-induced dual knowledge structure graphs for personalized learning path recommendation. *Preprint*, arXiv:2506.22303.

Jimmy de la Torre. 2009. Dina model and parameter estimation: A didactic. *Journal of Educational and Behavioral Statistics*, 34(1):115–130.

Zhiang Dong, Jingyuan Chen, and Fei Wu. 2025a. Knowledge is power: Harnessing large language models for enhanced cognitive diagnosis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(1):164–172.

Zhiang Dong, Jingyuan Chen, and Fei Wu. 2025b. Llm-driven cognitive diagnosis with solo taxonomy: A model-agnostic framework. *Frontiers of Digital Education*, 2(2):20.

Xiang Fang, Wanlong Fang, Daizong Liu, Xiaoye Qu, Jianfeng Dong, Pan Zhou, Renfu Li, Zichuan Xu, Lixing Chen, Panpan Zheng, et al. 2024a. Not all inputs are valid: Towards open-set video moment retrieval using language. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 28–37.

Xiang Fang, Wanlong Fang, Changshuo Wang, Daizong Liu, Keke Tang, Jianfeng Dong, Pan Zhou, and Beibei Li. 2025. Multi-pair temporal sentence grounding via multi-thread knowledge transfer network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 2915–2923.

Xiang Fang, Daizong Liu, Wanlong Fang, Pan Zhou, Yu Cheng, Keke Tang, and Kai Zou. 2023a. Annotations are not all you need: A cross-modal knowledge transfer network for unsupervised temporal sentence grounding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8721–8733.

Xiang Fang, Daizong Liu, Pan Zhou, and Yuchong Hu. 2022. Multi-modal cross-domain alignment network for video moment retrieval. *IEEE Transactions on Multimedia*, 25:7517–7532.

Xiang Fang, Daizong Liu, Pan Zhou, and Guoshun Nan. 2023b. You can ground earlier than see: An effective and efficient pipeline for temporal sentence grounding in compressed videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2448–2460.

Xiang Fang, Daizong Liu, Pan Zhou, Zichuan Xu, and Ruixuan Li. 2023c. Hierarchical local-global transformer for temporal sentence grounding. *IEEE Transactions on Multimedia*, 26:3263–3277.

Xiang Fang, Zeyu Xiong, Wanlong Fang, Xiaoye Qu, Chen Chen, Jianfeng Dong, Keke Tang, Pan Zhou, Yu Cheng, and Daizong Liu. 2024b. Rethinking weakly-supervised video temporal grounding from a game perspective. In *European Conference on Computer Vision*, pages 290–311. Springer.

Zhaohan Feng, Ruiqi Xue, Lei Yuan, Yang Yu, Ning Ding, Meiqin Liu, Bingzhao Gao, Jian Sun, Xinhu Zheng, and Gang Wang. 2025. Multi-agent embodied ai: Advances and future directions. *arXiv preprint arXiv:2505.05108*.

Reva Freedman, Yujian Zhou, Michael Glass, Jung Hee Kim, and Martha W Evens. 2022. Using rule induction to assist in rule construction for a natural-language based intelligent tutoring system. In *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society*, pages 362–367. Routledge.

Weibo Gao, Qi Liu, Zhenya Huang, Yu Yin, Haoyang Bi, Mu-Chun Wang, Jianhui Ma, Shijin Wang, and Yu Su. 2021. RCD: Relation map driven cognitive diagnosis for intelligent education systems. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 501–510. ACM.

Ilie Gligorea, Marius Cioca, Romana Oancea, Andra-Teodora Gorski, Hortensia Gorski, and Paul Tudorache. 2023. Adaptive learning using artificial intelligence in e-learning: A literature review. *Education Sciences*, 13(12):1216.

Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A. Rossi, Subhabrata Mukherjee, Xianfeng Tang, Qi He, Zhigang Hua, Bo Long, Tong Zhao, Neil Shah, Amin Javari, Yinglong Xia, and Jiliang Tang. 2025a. Retrieval-augmented generation with graphs (graphrag). *Preprint*, arXiv:2501.00309.

Ze Han, Yu Su, Qi Mo, and Xinyu Nie. 2025b. Neural network student cognitive diagnosis model based on multi-relational graph. In *Proceedings of the 2025 2nd International Conference on Generative Artificial Intelligence and Information Security*, GAIIS 2025, pages 515–521, Hangzhou, China. ACM.

Huu-Tuong Ho, Duc-Tin Ly, and Luong Vuong Nguyen. 2024. Mitigating hallucinations in large language models for educational application. In *2024 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, pages 1–4. IEEE.

Yuichi Inoue, Kou Misaki, Yuki Imajuku, So Kuroki, Taishi Nakamura, and Takuya Akiba. 2025. Wider or deeper? scaling LLM inference-time compute with adaptive branching tree search. *Preprint*, arXiv:2503.04412.

L. M. R. Jayasinghe and Dharshana Kasthurirathna. 2022. A goal-driven personalized educational content recommendation system for self-learning. *International Journal of Advanced Computer Science and Applications*, 13(11).

Weiqiang Jin, Hongyang Du, Biao Zhao, Xingwu Tian, Bohang Shi, and Guang Yang. 2025. A comprehensive survey on multi-agent cooperative decision-making: Scenarios, approaches, challenges and perspectives. *arXiv preprint arXiv:2503.13415*.

Jeff Johns, Sridhar Mahadevan, and Beverly Woolf. 2006. Estimating student proficiency using an item response theory model. In *International conference on intelligent tutoring systems*, pages 473–480. Springer.

Ankur Joshi, Saket Kale, Satish Chandel, and D Kumar Pal. 2015. Likert scale: Explored and explained. *British journal of applied science & technology*, 7(4):396.

Jonathan Larson and Steven Truitt. 2024. Graphrag: Unlocking llm discovery on narrative private data. *Microsoft Research Blog*.

Guangquan Li, Yuqing Hu, Junkai Shuai, Tonghua Yang, Yonghong Zhang, Shiming Dai, and Naixue Xiong. 2022. Neuralncd: A neural network cognitive diagnosis model based on multi-dimensional features. *Applied Sciences*, 12(19).

Wenyi Liao. 2025. Study on learning path optimization based on reinforcement learning and collaborative filtering. In *Second International Conference on Big Data, Computational Intelligence, and Applications (BDCIA 2024)*, volume 13550, pages 878–885. SPIE.

Joy Jia Yin Lim, Ye He, Jifan Yu, Xin Cong, Daniel Zhang-Li, Zhiyuan Liu, Huiqin Liu, Lei Hou, Juanzi Li, and Bin Xu. 2025. Personalized learning path planning with goal-driven learner state modeling. *Preprint*, arXiv:2510.13215.

Chunmao Liu, Somkiat Tuntiwongwanich, and Thiyaporn Kantathanawat. 2025. Artificial intelligence-driven cognitive diagnosis and adaptive learning: The impact of online course stickiness and learning skills.

Daizong Liu, Xiaowen Cai, Pan Zhou, Xiaoye Qu, Lichao Sun, and Wei Hu. 2026a. Are large vision-language models robust to adversarial visual transformations? *IEEE Transactions on Information Forensics and Security*.

Daizong Liu, Baoquan Chen, and Wei Hu. 2026b. Spatial-spectral homogeneous attacks on physical-world large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 7114–7122.

Daizong Liu, Xiang Fang, Pan Zhou, Xing Di, Weining Lu, and Yu Cheng. 2023a. Hypotheses tree building for one-shot temporal sentence localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 1640–1648.

Daizong Liu, Wangqin Liu, Xiaowen Cai, Pan Zhou, Runwei Guan, Xiaoye Qu, and Bo Du. 2026c. Generating transferable attacks across large vision-language models using adversarial deformation learning. *Pattern Recognition*, page 113194.

Daizong Liu, Xiaoye Qu, Xing Di, Yu Cheng, Zichuan Xu, and Pan Zhou. 2022a. Memory-guided semantic learning network for temporal sentence grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1665–1673.

Daizong Liu, Xiaoye Qu, Jianfeng Dong, and Pan Zhou. 2021a. Adaptive proposal generation network for temporal sentence localization in videos. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9292–9301.

Daizong Liu, Xiaoye Qu, Jianfeng Dong, Pan Zhou, Yu Cheng, Wei Wei, Zichuan Xu, and Yulai Xie. 2021b. Context-aware biaffine localizing network for temporal sentence grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11235–11244.

Daizong Liu, Xiaoye Qu, and Wei Hu. 2022b. Reducing the vision and language bias for temporal sentence grounding. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4092–4101.

Daizong Liu, Xiaoye Qu, Xiao-Yang Liu, Jianfeng Dong, Pan Zhou, and Zichuan Xu. 2020. Jointly cross-and self-modal graph attention network for query-based moment localization. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 4070–4078.

Daizong Liu, Xiaoye Qu, Yinzhen Wang, Xing Di, Kai Zou, Yu Cheng, Zichuan Xu, and Pan Zhou. 2022c. Unsupervised temporal video grounding with deep semantic clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1683–1691.

Daizong Liu, Shuangjie Xu, Xiao-Yang Liu, Zichuan Xu, Wei Wei, and Pan Zhou. 2021c. Spatiotemporal graph neural network based mask reconstruction for video object segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2100–2108.

Daizong Liu, Mingyu Yang, Xiaoye Qu, Pan Zhou, Yu Cheng, and Wei Hu. 2024a. A survey of attacks on large vision-language models: Resources, advances, and future trends. *arXiv preprint arXiv:2407.07403*.

Daizong Liu, Mingyu Yang, Xiaoye Qu, Pan Zhou, Xiang Fang, Keke Tang, Yao Wan, and Lichao Sun. 2024b. Pandora's box: Towards building universal attackers against real-world large vision-language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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

Lin Lu, Zhigang Zuo, Ziji Sheng, and Pan Zhou. 2025. Merger-as-a-stealer: Stealing targeted pii from aligned llms with model merging. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5817–5836.

Rui Lv, Qi Liu, Weibo Gao, Haotian Zhang, Junyu Lu, and Linbo Zhu. 2025. Genal: Generative agent for adaptive learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(1):577–585.

Haiping Ma, Manwei Li, Le Wu, Haifeng Zhang, Yunbo Cao, Xingyi Zhang, and Xuemin Zhao. 2022. Knowledge-sensed cognitive diagnosis for intelligent education platforms. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pages 1451–1460.

meta llama. 2025. Llama-3.3-70b-instruct. Accessed: 2025-05-10.

OpenAI. 2025. GPT-5.2. `https://www.openai.com`. Large language model.

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

Qwen. 2025. Qwen3-235b-a22b-instruct-2507. Accessed: 2025-08-10.

Jiawen Shi, Zenghui Yuan, Yinuo Liu, Yue Huang, Pan Zhou, Lichao Sun, and Neil Zhenqiang Gong. 2024. Optimization-based prompt injection attack to llm-as-a-judge. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 660–674.

Zhaoyu Shou, Xianying Lu, Zhengzheng Wu, Hua Yuan, Huibing Zhang, and Junli Lai. 2020. On learning path planning algorithm based on collaborative analysis of learning behavior. *IEEE Access*, 8:119863–119879.

Marlene Steinbach, Shreya Bhandari, Jennifer Meyer, and Zachary A Pardos. 2025. When llms hallucinate: Examining the effects of erroneous feedback in math tutoring systems. In *Proceedings of the Twelfth ACM Conference on Learning@ Scale*, pages 139–150.

Weiwei Sun, Miao Lu, Zhan Ling, Kang Liu, Xuesong Yao, Yiming Yang, and Jiecao Chen. 2025. Scaling long-horizon llm agent via context-folding. *arXiv preprint arXiv:2510.11967*.

Olga Tapalova and Nadezhda Zhiyenbayeva. 2022. Artificial intelligence in education: Aied for personalised learning pathways. *Electronic Journal of e-Learning*, 20(5):639–653.

Guiyao Tie, Tianyao Luo, Xueyang Zhou, Chaoran Hu, Yunhong He, Junran Wu, Yuanfan Yao, Pan Zhou, and Lichao Sun. Routejudge: Benchmarking llm-as-a-judge with routing strategies.

Guiyao Tie, Zenghui Yuan, Zeli Zhao, Chaoran Hu, Tianhe Gu, Ruihang Zhang, Sizhe Zhang, Junran Wu, Xiaoyue Tu, Ming Jin, et al. 2025a. Can llms correct themselves? a benchmark of self-correction in llms. *arXiv preprint arXiv:2510.16062*.

Guiyao Tie, Zeli Zhao, Dingjie Song, Fuyang Wei, Rong Zhou, Yurou Dai, Wen Yin, Zhejian Yang, Jiangyue Yan, Yao Su, et al. 2025b. A survey on post-training of large language models. *arXiv preprint arXiv:2503.06072*.

Dong-Bo Tu, Yan Cai, Hai-Qi Dai, and Shu-Liang Ding. 2011. A polytomous cognitive diagnosis model: P-dina model. *Acta Psychologica Sinica*, 42:1011–1020.

Fei Wang, Qi Liu, Enhong Chen, Zhenya Huang, Yuying Chen, Yu Yin, Zai Huang, and Shijin Wang. 2020. Neural cognitive diagnosis for intelligent education systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):6153–6161.

Fei Wang, Qi Liu, Enhong Chen, Zhenya Huang, Yu Yin, Shijin Wang, and Yu Su. 2022. Neuralcd: a general framework for cognitive diagnosis. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):8312–8327.

Tianfu Wang, Yi Zhan, Jianxun Lian, Zhengyu Hu, Nicholas Jing Yuan, Qi Zhang, Xing Xie, and Hui Xiong. 2025a. LLM-powered multi-agent framework for goal-oriented learning in intelligent tutoring system. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25 Companion, pages 510–519, Sydney, NSW, Australia. ACM.

Tianfu Wang, Yi Zhan, Jianxun Lian, Zhengyu Hu, Nicholas Jing Yuan, Qi Zhang, Xing Xie, and Hui Xiong. 2025b. Llm-powered multi-agent framework for goal-oriented learning in intelligent tutoring system. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 510–519.

Zichao Wang, Angus Lamb, Evgeny Saveliev, Pashmina Cameron, Jordan Zaykov, Jose Miguel Hernandez-Lobato, Richard E Turner, Richard G Baraniuk, Craig Barton, Simon Peyton Jones, et al. 2021. Results and insights from diagnostic questions: The neurips 2020 education challenge. In *NeurIPS 2020 Competition and Demonstration Track*, pages 191–205. PMLR.

Qingsong Wen, Jing Liang, Carles Sierra, Rose Luckin, Richard Tong, Zitao Liu, Peng Cui, and Jiliang Tang. 2024. Ai for education (ai4edu): Advancing personalized education with llm and adaptive learning. In

Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pages 6743–6744.

Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Cheng Niu, Randy Zhong, Juntong Song, and Tong Zhang. 2023. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. *Preprint*, arXiv:2401.00396.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yi Ding, Boyang Hong, Ming Zhang, et al. 2023. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*.

Zhishang Xiang, Chuanjie Wu, Qinggang Zhang, Shengyuan Chen, Zijin Hong, Xiao Huang, and Jinsong Su. 2025. When to use graphs in rag: A comprehensive analysis for graph retrieval-augmented generation. *arXiv preprint arXiv:2506.05690*.

Yu Xie, Michael Fang, and Kimberlee Shauman. 2015. Stem education. *Annual review of sociology*, 41(1):331–357.

Hai Yan, Haijian Ma, Xiaowen Cai, Daizong Liu, Zenghui Yuan, Xiaoye Qu, Jianfeng Dong, Runwei Guan, Xiang Fang, Hongyang He, et al. 2025. Fit the distribution: Cross-image/prompt adversarial attacks on multimodal large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Chun-Kit Yeung. 2019. Deep-irt: Make deep learning based knowledge tracing explainable using item response theory. *Preprint*, arXiv:1904.11738.

Peida Zhan, Wen-Chung Wang, Hong Jiao, and Yufang Bian. 2018. Probabilistic-input, noisy conjunctive models for cognitive diagnosis. *Frontiers in Psychology*, Volume 9 - 2018.

Xueqiao Zhang, Chao Zhang, Jianwen Sun, Jun Xiao, Yi Yang, and Yawei Luo. 2025a. Eduplanner: Llm-based multiagent systems for customized and intelligent instructional design. *IEEE Transactions on Learning Technologies*, 18:416–427.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. 2025b. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2023. Language agent tree search unifies reasoning acting and planning in language models. *arXiv preprint arXiv:2310.04406*.

A Appendix

A.1 Dynamic Knowledge Graphs: Adaptive Domain Knowledge Graph Automatic Construction Mechanism

The Dynamic Knowledge Graph employed in LEARNERCOMPASS is composed of **Entities** (Nodes) and **Relationships** (Edges). To ensure the graph accurately depicts the intrinsic structure and logic of disciplinary knowledge, we adhere to a rigorous schema definition.

A.1.1 Entity (Node) Specification

The fundamental building blocks of our knowledge graph are entity nodes, each representing a "minimal learning unit." These units are defined as knowledge components with clear semantic boundaries, independent instructional objectives, and specific evaluation criteria (e.g., "Matrix Multiplication" or "Basis and Dimension" in Linear Algebra).

Entity Attributes. Each entity node encapsulates structured attributes essential for identification, semantic representation, and graph operations. Table 5 details the primary attribute design.

Table 5: Attribute Definitions for Entities in the Dynamic Knowledge Graph.

Attribute	Description
`Id`	Unique identifier for the entity.
`Name`	The entity's name, following strict naming conventions for clarity (e.g., "Cramer's Rule").
`Description`	A concise definition derived from textbooks, Wikipedia, or LLM generation, summarizing core content and context.
`Labels`	Categorical tags from a predefined set (see Table 6) indicating the pedagogical role.
`Embedding`	A 1024-dimensional semantic vector generated by a local embedding model (e.g., Qwen3-embedding-4B) for high-dimensional representation.

Label System. To enhance interpretability, we employ a predefined label system that categorizes entities into five distinct pedagogical roles, as shown in Table 6. This allows for flexible, multi-label annotation via LLM prompting.

A.1.2 Relationship (Edge) Specification

Edges in the graph represent semantic connections between knowledge points, encoding the logical progression, generalization, and methodological dependencies essential for cognitive diagnosis and path recommendation.

Table 6: Predefined Entity Labels and Semantic Roles.

Label	Description & Examples
CONCEPT	Core definitions and fundamental notions (e.g., "Eigenvalue", "Determinant").
THEOREM	Formally provable propositions (e.g., "Cramer's Rule", "Rank-Nullity Theorem").
RULE	General operational rules or laws (e.g., "Associativity of Matrix Multiplication").
METHOD	Procedural techniques for problem-solving (e.g., "Gaussian Elimination").
APPLICATION	Real-world use cases (e.g., "SVD in Image Compression").

Predefined Relationship Types. To standardize semantic interactions, we define five core relationship types (see Table 7) in our schema configuration (`Label.json`).

Table 7: Definitions of Relationship Types in the Dynamic Knowledge Graph.

Relationship	Description & Example
IS_PREREQUISITE_FOR	**Prerequisite Relation.** Mastery of the source node is required before the target. (e.g., "Definition of Triangle" → "Congruent Triangles").
GENERALIZES	**Generalization Relation.** The source is a fundamental concept, and the target is a specific instance or extension. (e.g., "Sum of angles in a triangle" → "Sum of angles in a polygon").
IS_A_METHOD_FOR	**Methodological Relation.** The source is a technique used to solve the target problem. (e.g., "Factoring" → "Quadratic Equation Solving").
IS_A_TYPE_OF	**Taxonomic Relation.** The source is a subclass or specific type of the target. (e.g., "Rectangle" → "Parallelogram").
IS_INVERSE_OF	**Inverse Relation.** Represents logical or algebraic inversion. (e.g., "Factoring" ↔ "Polynomial Multiplication").

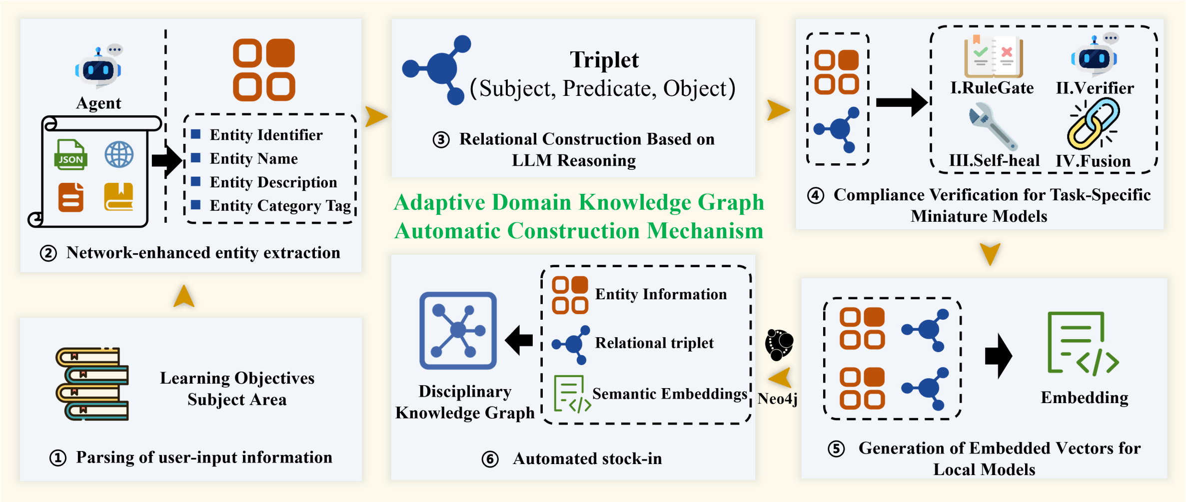

Figure 5: The workflow of the **Adaptive Domain Knowledge Graph** $G_{dynamic}$ **Automatic Construction Mechanism**. The process proceeds through six stages: ① Parsing user inputs to identify learning objectives; ② Network-enhanced entity extraction; ③ LLM-based reasoning to construct relational triplets (Subject, Predicate, Object); ④ Compliance verification to ensure accuracy for task-specific models; ⑤ Generation of semantic embedding vectors; and ⑥ Automated stock-in to the Disciplinary Knowledge Graph using Neo4j.

A.2 Multi-agent Cognitive Diagnostics: Dynamic Cognitive Diagnosis System

A.2.1 Encoder multi-agent prompts

```
AssessmentAgent Prompt:

"""You are an educational expert in the field of
↪  {discipline}. You are skilled at analyzing
↪  the knowledge points involved in a problem
↪  according to a learner's current
↪  cognitive-level vector.

Now, based on the learner's [current knowledge
↪  point], [learned knowledge points],
↪  [question information], and [answering
↪  process], you must analyze the [relevant
↪  prerequisite knowledge points and their
↪  levels] related to the question, and their
↪  [influence weight on the current knowledge
↪  point] within this question. Specifically:
1. [Current knowledge point]: the knowledge point
↪  the learner is studying in the current
↪  module.
2. [Learned knowledge points]: the set of
↪  knowledge points the learner has learned so
↪  far, including the current knowledge point
↪  and prerequisite knowledge points. Each
↪  knowledge point is divided into three levels:
↪  ["beginner", "advanced", "proficient"],
↪  representing different difficulty aspects
↪  within the knowledge point (for example, rote
↪  recall of a concept belongs to "beginner",
↪  understanding the concept belongs to
↪  "advanced", and applying the concept belongs
↪  to "proficient").
3. [Question information]: the question used to
↪  assess the learner's mastery of the current
↪  knowledge point, including question type,
↪  difficulty, stem, standard answer,
↪  explanation, etc. Although the question
↪  primarily assesses the current knowledge
↪  point, it may also involve some prerequisite
↪  knowledge points.
4. [Answering process]: the learner's
↪  solution/answer to the question. From this
↪  part one can to some extent analyze which
↪  prerequisite knowledge points the learner
↪  lacks and which deficits led to incorrect
↪  answers on the current-knowledge-related
↪  question.
5. [Relevant prerequisite knowledge points and
↪  their levels]: the prerequisite knowledge
↪  points other than the current knowledge point
↪  that the current question may involve. Among
↪  these, only certain levels of some
↪  prerequisite knowledge points may be
↪  insufficient and have caused the learner to
↪  answer the current question incorrectly;
↪  therefore, the goal is to identify precisely
↪  those prerequisite knowledge-point-levels
↪  that are lacking. Further, the prerequisite
↪  knowledge points involved in the current
↪  question may be of "beginner", "advanced", or
↪  "proficient" level, so the identification
↪  must specify the level for each prerequisite
↪  knowledge point.
```

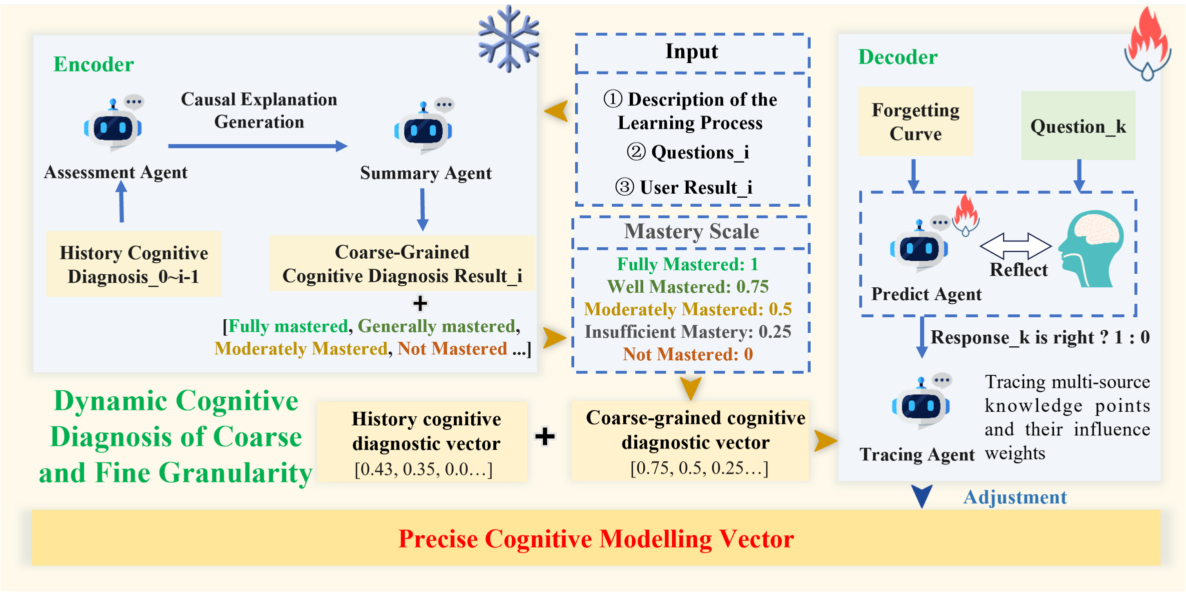

Figure 6: Architecture of the **Dynamic Cognitive Diagnosis System** with coarse and fine granularity. The system employs an **Encoder-Decoder structure**. The **Encoder** (left) utilizes Assessment and Summary Agents to transform historical data into a coarse-grained cognitive vector. The **Decoder** (right) refines this into a **Precise Perception Modelling Vector** through a "predict-verify-reflect" loop involving the Predict Agent and Tracing Agent, dynamically adjusting mastery scales based on user responses.

```
6. [Influence weight on the current knowledge
↪  point]: different prerequisite knowledge
↪  points have different degrees of influence on
↪  the current knowledge point in the context of
↪  the question; therefore, based on the
↪  question, the learner's answering process,
↪  and the knowledge points themselves, analyze
↪  the influence weight of each prerequisite
↪  knowledge point on the current knowledge
↪  point. The weight value should be a
↪  floating-point number between 0 and 1.

Given the relevant information:
1. [Current knowledge point]:
{current_learning_node}
2. [Learned knowledge points]:
{historical_learning_nodes}
3. [Question information]:
{question}
4. [Answering process]:
{answering_process}

Now, based on the information above, analyze and
↪  output the [relevant prerequisite knowledge
↪  points and their levels] that led to the
↪  errors on the question, and their [influence
↪  weight on the current knowledge point] in
↪  this question. Requirements:
1. The relevant prerequisite knowledge points
↪  must be selected from the set of learned
↪  knowledge points.
2. Not all prerequisite knowledge points
↪  necessarily appear in the question; only
↪  return the knowledge points that are actually
↪  involved in the question. For example, if the
↪  current knowledge point has prerequisite A
↪  and prerequisite B, but the question only
↪  involves prerequisite B, then only B should
↪  be returned.
3. Output only a single DataFrame where the 1st
↪  column is the prerequisite knowledge point,
↪  the 2nd column is the level, and the 3rd
↪  column is the influence weight. Do not
↪  provide any additional explanatory text. The
↪  format should be:
pre_knowledge_point level influence_weight
KnowledgePoint1 advanced 0.37
"""
```

```
SummaryAgent Prompt:

"""You are a {role}. You excel at producing
↪  diagnostic assessments of a learner's
↪  cognitive level for the knowledge point
↪  currently being studied, based on various
↪  learner data.

Now, using the learner's [current knowledge
↪  point], [learned knowledge points], [test
↪  results], [error-prerequisite tracing], and
↪  [historical cognitive-diagnosis records],
↪  produce the [diagnostic result for the
↪  current knowledge point], where:
1. [Current knowledge point]: the knowledge point
↪  the learner is studying in the current
↪  module.
```

```
2. [Learned knowledge points]: the set of
↪  knowledge points the learner has learned so
↪  far, including the current knowledge point
↪  and prerequisite knowledge points. Each
↪  knowledge point is divided into ["beginner",
↪  "advanced", "proficient"] levels as
↪  described above.
3. [Test results]: the learner's performance on
↪  items targeting the current knowledge point,
↪  including information about the items (type,
↪  difficulty, etc.) and answering information
↪  such as correctness or score.
4. [Error-prerequisite tracing]: tracing and
↪  analysis of the prerequisite knowledge
↪  points involved in the incorrect test items.
↪  Presented as a list where each element
↪  represents one incorrect item and includes
↪  the item id and the involved prerequisite
↪  knowledge points. The involved prerequisite
↪  knowledge points are given in a DataFrame
↪  where the 1st column is the prerequisite
↪  knowledge point, the 2nd column is the level
↪  of that knowledge point, and the 3rd column
↪  is that knowledge point's influence weight in
↪  that item.
5. [Historical cognitive-diagnosis records]:
↪  records describing past diagnostic results
↪  of the learner's cognitive levels for
↪  knowledge points.
6. [Diagnostic result for the current knowledge
↪  point]: an assessment of mastery for the
↪  current knowledge point. Because knowledge
↪  points are divided into ["beginner",
↪  "advanced", "proficient"], you must give a
↪  diagnosis for each of the three levels.
↪  Additionally, inter-knowledge dependencies
↪  mean that insufficient mastery of some
↪  prerequisite knowledge points may affect the
↪  diagnosis for the current knowledge point.
↪  The final diagnosis should integrate test
↪  results, emotional fluctuations, and other
↪  multi-source information where appropriate.

Given the relevant information:
1. [Current knowledge point]:
{current_learning_node}
2. [Learned knowledge points]:
{historical_learning_nodes}
3. [Test results]:
{test_result}
4. [Error-prerequisite tracing]:
{traced_former_knowledge}
5. [Historical cognitive-diagnosis records]:
{historical_cognitive_diagnosis}

Now, based on the above information, produce the
↪  [diagnostic result for the current knowledge
↪  point]. Requirements:
1. Diagnose only the three levels for the current
↪  knowledge point.
2. For each level, select exactly one label from:
↪  "Fully mastered", "Well mastered",
↪  "Moderately mastered", "Insufficiently
↪  mastered", "Not mastered at all".
3. The final output must contain the diagnosis
↪  result and the diagnostic rationale only; do
↪  not provide any additional explanatory text.
↪  Example format:
Diagnosis result: KnowledgePoint1: { "beginner":
↪  "Fully mastered", "advanced": "Moderately
↪  mastered", "proficient": "Insufficiently
↪  mastered" }
Diagnosis rationale: ......
"""
```

A.2.2 Decoder multi-agent prompts

```
PredictAgent Prompt:

"""You are an analyzer skilled at predicting
↪  whether a learner will answer a given
↪  question correctly based on question
↪  information and the learner's cognitive
↪  profile.

Now, based on [question information] and [learner
↪  cognitive levels], produce the [prediction
↪  result], where:
1. [Question information]: includes question
↪  type, difficulty, stem, answer, explanation,
↪  etc.
2. [Learner cognitive levels]: describes the
↪  learner's learned knowledge points and
↪  mastery levels. Each knowledge point is
↪  divided into ["beginner", "advanced",
↪  "proficient"] levels, and mastery for each
↪  level is given as a floating-point number
↪  between 0 and 1, where values closer to 1
↪  indicate higher mastery. The overall
↪  structure is a dictionary like: {
↪  "KnowledgePoint1": { "beginner": 0.83,
↪  "advanced": 0.67, "proficient": 0.25 },
↪  "KnowledgePoint2": { "beginner": 0.52,
↪  "advanced": 0.31, "proficient": 0.1 } }.
3. [Prediction result]: a prediction of whether
↪  the learner can correctly answer the
↪  question. For subjective questions that
↪  require a solution process, the answer is
↪  considered correct only if both the process
↪  and the result are correct. The prediction
↪  should be 0 or 1, where 0 means the learner
↪  cannot answer correctly and 1 means the
↪  learner can answer correctly.

Given the relevant information:
1. [Question information]:
{question}
2. [Learner cognitive levels]:
{cognitive_vector}

Now, based on the above information, produce the
↪  [prediction result]. Requirements:
1. Output only 0 or 1, where 0 indicates cannot
↪  answer correctly and 1 indicates can answer
↪  correctly. Do not provide any additional
↪  explanatory text.
"""
```

```
TracingAgent Prompt:
```

```
"""You are an expert in reflective modeling of
↪  learner cognitive levels. You can adjust a
↪  learner's cognitive levels on certain
↪  knowledge points by using prediction feedback
↪  in order to produce a more precise cognitive
↪  model.

Now, using [current knowledge point], [question
↪  information], [prerequisite tracing],
↪  [question prediction feedback], and
↪  [reflection history], adjust the [learner
↪  cognitive levels], where:
1. [Current knowledge point]: the knowledge point
↪  the learner is studying in the current
↪  module.
2. [Question information]: includes question
↪  type, difficulty, stem, standard answer, and
↪  explanation. Some questions may include
↪  images in the stem; for those invisible image
↪  contents you may ignore them.
3. [Prerequisite tracing]: prerequisite
↪  knowledge points involved in the question
↪  besides the current knowledge point,
↪  including each involved prerequisite's level
↪  and influence weight (range 0 to 1). This
↪  information serves as important evidence for
↪  adjusting other knowledge points' cognitive
↪  levels: depending on whether the learner's
↪  answer contains errors related to those
↪  prerequisite knowledge parts, raise or lower
↪  the corresponding mastery values.
4. [Question prediction feedback]: composed of
↪  the prediction result and the actual result.
↪  The prediction result is produced by the
↪  predictor module from [question information]
↪  and [learner cognitive levels]; the actual
↪  result is obtained from the learner's
↪  observed performance in the assessment. Both
↪  are represented as 0 or 1 (0 = learner cannot
↪  answer correctly, 1 = learner can answer
↪  correctly).
5. [Reflection history]: describes previous
↪  adjustments and reflection steps that were
↪  attempted based on related information;
↪  these past adjustments were unsuccessful but
↪  can provide lessons to guide more
↪  fine-grained adjustments.
6. [Learner cognitive levels]: describes the
↪  learner's learned knowledge points and
↪  mastery levels. Each knowledge point is
↪  divided into ["beginner", "advanced",
↪  "proficient"], with mastery values between 0
↪  and 1. The structure is a dictionary like: {
↪  "KnowledgePoint1": { "beginner": 0.83,
↪  "advanced": 0.67, "proficient": 0.25 },
↪  "KnowledgePoint2": { "beginner": 0.52,
↪  "advanced": 0.31, "proficient": 0.1 } }.

Given the relevant information:
1. [Current knowledge point]:
{current_learning_node}
2. [Question information]:
{question}
3. [Prerequisite tracing]:
{traced_former_knowledge}
4. [Question prediction feedback]:
{prediction}
5. [Reflection history]:
{historical_reflexion}
6. [Learner cognitive levels]:
{cognitive_vector}

Now, based on the above information, adjust the
↪  [learner cognitive levels] so that the
↪  predictor module will produce predictions
↪  consistent with actual outcomes.
↪  Requirements:
1. The knowledge point most relevant to the
↪  question is the [current knowledge point], so
↪  the primary adjustments should be to the
↪  cognitive levels corresponding to the
↪  [current knowledge point].
2. Some questions may involve knowledge points
↪  other than the [current knowledge point], so
↪  you may analyze the question's multi-source
↪  knowledge aspects and appropriately adjust
↪  those knowledge points' cognitive levels.
↪  Note these other knowledge points may not be
↪  fully present in the [learner cognitive
↪  levels]; only adjust knowledge points that do
↪  appear in the [learner cognitive levels].
3. The final output should contain only the
↪  analysis process and the cognitive-level
↪  adjustments for the knowledge points that
↪  need adjustment; do not include unchanged
↪  knowledge points or any additional
↪  explanatory text. Example format:
Analysis process: ......
Cognitive levels: [ { "KnowledgePoint1": {
↪  "beginner": 0.32, "advanced": 0.21,
↪  "proficient": 0.14 } }, { "KnowledgePoint2":
↪  { "beginner": 0.85, "advanced": 0.6,
↪  "proficient": 0.39 } } ]
"""
```

A.3 Multi-Agent Learning Path Generation: AB-MCTS-M Multi-Model Path Fusion Algorithm

General Learning Path Generation Agent Prompt. In the following prompt design, certain fields (e.g., `{mode_indicator}`, `{discipline}`, as well as role descriptions and capability constraints) are intentionally designed to be replaceable and configurable. By injecting domain-specific corpora into these fields or by incorporating domain-oriented fine-tuning data, the corresponding large language model instance can be explicitly constrained to function as an educational expert in a particular subject area.

For example, by introducing instructional materials and cognitive diagnosis data related to linear algebra or advanced mathematics during training or inference, the model can exhibit stable domain expertise and pedagogical consistency in tasks such as learning path planning, cognitive assessment, and adaptive instructional intervention.

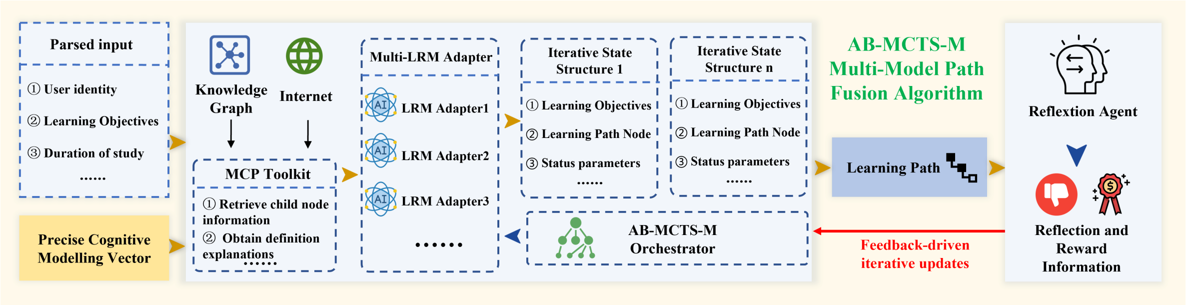

Figure 7: Schematic of the **AB-MCTS-M Multi-Model Path Fusion Algorithm**. The engine processes the cognitive diagnostic vector and learning objectives through an **MCP Toolkit** and Knowledge Graph. It orchestrates a **Multi-LRM Adapter** (containing diverse expert models) to generate iterative state structures. The **AB-MCTS-M Orchestrator** fuses these states into an optimal Learning Path, while the **Reflexion Agent** provides feedback-driven iterative updates to refine future planning strategies.

```
General Learning Path Generation Agent Prompt:
"""
{mode_indicator} - As a professional learning planner, please formulate the optimal learning path.

{reflexion_experience}

{user_adaptation_requirements}

## Basic Information
Original Learning Goal: {state.original_target_concept}
Current Knowledge: {json.dumps(state.current_knowledge, ensure_ascii=False)}
Learned: {' -> '.join(state.path_history[-5:]) if state.path_history else 'None'}
User's current cognitive situation from diagnosis:
{cognitive_diagnosis_section}

## Knowledge Graph Enhancement Info
{mcp_info}

## Generation Requirements:
You are a professional educational researcher,
{"Based on the above Reflexion experience and user adaptation requirements"
 if is_reflexion_mode else
 "Focus heavily on user adaptability"},
please formulate an optimal learning path.

Core Principles:
1. User Adaptation Priority:
   Path must perfectly match user identity and needs
2. Continuous Quality Improvement:
   {f"Apply Reflexion experience to avoid historical issues"
    if is_reflexion_mode else
    "Establish a high-quality baseline"}
3. Clear Logical Progression:
   From basic to advanced, step-by-step
4. Utility Oriented:
   Combine theory with practice
5. Reasonable Time Schedule:
   Fits user time constraints

# Path Length Judgment Guidelines:
# You can first judge the gap between this learning goal and the user's foundation.
# Judge according to three levels ABC:
# A: Large gap, long path (length > 14 minimal learning units)
# B: Moderate gap (approx. 8-12 minimal learning units)
# C: Small gap (length < 6 minimal learning units)

{"Please strictly optimize based on Reflexion guidance!"
 if is_reflexion_mode else
 "Please focus on user adaptability to build a high-quality baseline for subsequent Reflexion!"}

Please return in the following JSON format:
{
  "path": [
    "Learning Step 1",
    "Learning Step 2",
    "Learning Step 3",
    ...
    ],
  "reasoning":
    "Detailed explanation of path design reasoning,
     {f'combining Reflexion experience and'
      if is_reflexion_mode else
      'emphasizing'} user adaptation considerations",
  "estimated_time":
    Estimated total learning time (hours),
  "confidence_score":
    Confidence assessment of this path (0-1)
   {", based on Reflexion guidance should be >= "
     + str(reflexion_context.get(
         'reward_expectations', {}
       ).get('expected_minimum_score', 0.6))
    if is_reflexion_mode else ""}
}
```

Reflexion Agent Prompt.

Algorithm 1 Path Planning via Adaptive Branching MCTS (AB-MCTS-M)

```
Input: Target agent set $\mathcal{V}_{\text{target}}$ (Heterogeneous LLM Experts); Original task instruction $P_{\text{tar}}$ (Learning Goal); Learner State $\boldsymbol{\theta}$ (from Sec. 3.2); Max iterations $T_{\max}$;
Output: Optimal Learning Path $\pi^*$
Initialize tree $\mathcal{T}$ with root node $n_0 \leftarrow P_{\text{tar}}$
Initialize Mixed Model parameters $\Phi$ (priors for $\mu, \alpha, \sigma$)
for $t = 1$ to $T_{\max}$ do
    Selection:
    Sample rewards $\tilde{r}_a \sim P(R \mid \Phi)$ for all actions $a \in$ children($n_0$) $\cup$ {GEN}
    Select node $n_{\text{curr}}$ by traversing $\mathcal{T}$ using Thompson Sampling on $\tilde{r}$
    Expansion (Adaptive Branching):
    if selected action is GEN then
        Sample subset of experts $\mathcal{V}_{\text{sub}} \subset \mathcal{V}_{\text{target}}$
        Generate diverse next-steps: $\mathcal{C} \leftarrow \{F_v(\text{state}(n_{\text{curr}})) \mid v \in \mathcal{V}_{\text{sub}}\}$
        Add candidates $\mathcal{C}$ to $\mathcal{T}$ as children of $n_{\text{curr}}$
        $n_{\text{new}} \leftarrow$ Select one $c \in \mathcal{C}$
    else
        $n_{\text{new}} \leftarrow$ Refine/Elaborate existing node $n_{\text{curr}}$
    end if
    Evaluation:
    $r_{\text{obs}} \leftarrow \text{RewardAgent}(n_{\text{new}}, \boldsymbol{\theta})$        ▷ Assess pedagogical fit
    Backup & Update:
    Backpropagate $r_{\text{obs}}$ up the tree $\mathcal{T}$
    Update Mixed Model posteriors $\Phi \leftarrow \text{Update}(\Phi, r_{\text{obs}})$
end for
Final Application:
$\pi^* \leftarrow \text{GetBestPath}(\mathcal{T})$
return $\pi^*$
```

```
Reflection Analysis Prompt:
"""Conduct deep reflection analysis based on the
↪  Reflexion architecture. You will analyze the
↪  performance of the generated learning path
↪  and generate reward signals and improvement
↪  suggestions.

## Generation Task Info
Target Concept: {target_concept}
Generated Path: {optimal_path}
Reasoning: {path_data.get('reasoning', '')}
TreeQuest Confidence:
↪  {path_data.get('confidence_score', 0):.3f}

## User Profile (Core Focus)
User Identity: {user_context.get('user_identity',
↪  'Unknown')}
Learning Goal:
↪  {user_context.get('user_learning_goal',
↪  target_concept)}
Time Constraint:
↪  {user_context.get('user_time_constraint',
↪  'Unlimited')}
Learning Preferences: {user_context.get('user_l⌋
↪  earning_preferences', {})}
Specific Requirements: {user_context.get('user_⌋
↪  specific_requirements', [])}

## Evaluation Result Analysis
Overall Score: {evaluation.overall_score:.3f}
User Adaptation:
↪  {evaluation.user_adaptation_score:.3f} (Core
↪  Dimension)
Detailed Scores:
- Logical Progression:
↪  {evaluation.logical_progression:.3f}
- Completeness: {evaluation.completeness:.3f}
- Practicality: {evaluation.practicality:.3f}
- Difficulty Appropriateness:
↪  {evaluation.difficulty_appropriateness:.3f}
- Time Estimation Accuracy:
↪  {evaluation.time_estimation_accuracy:.3f}
- Content Quality:
↪  {evaluation.content_quality:.3f}

User Adaptation Details:
- Identity Match:
↪  {evaluation.user_identity_match:.3f}
- Goal Alignment:
↪  {evaluation.learning_goal_alignment:.3f}
- Time Feasibility:
↪  {evaluation.time_constraint_feasibility:.3f}
- Preference Accommodation: {evaluation.learnin⌋
↪  g_preference_accommodation:.3f}
```

```
Evaluation Feedback: {evaluation.detailed_feedback}
Adaptation Feedback: {evaluation.adaptation_feedback}
Strengths: {evaluation.strengths}
Weaknesses: {evaluation.weaknesses}

{history_context}
{user_trend_context}

## Reflexion Task

Please reflect comprehensively following the Reflexion architecture, focusing on user adaptation:

### 1. Trial & Error Analysis
- Identify major issues in the generation process.
- Analyze specific reasons for insufficient user adaptation.
- Determine key areas needing improvement.

### 2. Root Cause Analysis
- Analyze the fundamental causes of the problems.
- Identify systemic generation defects.
- Analyze deviations in understanding user needs.

### 3. User Adaptation Reflection
- Deeply analyze the degree of match between the path and user identity.
- Evaluate the likelihood of achieving learning goals.
- Check the realistic feasibility of the schedule.
- Analyze the degree of integration of learning preferences.

### 4. Reward Signal Generation
- Calculate reward scores based on Reflexion principles.
- Provide a breakdown of rewards by dimension.
- Provide reinforcement signals for the next generation.

### 5. Improvement Strategy
- Provide concrete, actionable improvement suggestions.
- Generate specific strategies for this user type.
- Formulate guiding principles for the next generation.

## Output Format (Standardized JSON)

{{
    "problem_analysis": "Detailed problem analysis",
    "root_cause_analysis": "Root cause analysis",
    "improvement_suggestions": ["Suggestion 1", "Suggestion 2", "Suggestion 3"],
    "actionable_insights": ["Actionable Insight 1", "Actionable Insight 2", "Actionable Insight 3"],
    "user_adaptation_analysis": "Deep analysis of user adaptation",
    "identity_specific_recommendations": ["Identity-specific Rec 1", "Identity-specific Rec 2"],
    "goal_alignment_improvements": ["Goal alignment improvement 1", "Goal alignment improvement 2"],
    "time_management_suggestions": ["Time management suggestion 1", "Time management suggestion 2"],
    "preference_optimization_tips": ["Preference optimization tip 1", "Preference optimization tip 2"],
    "generation_guidelines": {{
        "prompt_improvements": "Prompt improvement suggestions",
        "model_selection": "Model selection suggestions",
        "context_requirements": "Context requirements",
        "user_specific_guidelines": "User-specific guidelines",
        "quality_checkpoints": ["Quality checkpoint 1", "Quality checkpoint 2"]
    }},
    "context_for_next_generation": "Specific guiding context for next generation",
    "reward_score": Reward Score (0-1),
    "reward_breakdown": {{
        "user_adaptation_reward": User adaptation reward score,
        "content_quality_reward": Content quality reward score,
        "efficiency_reward": Efficiency reward score,
        "innovation_reward": Innovation reward score
    }},
    "reflector_confidence": Reflection confidence (0-1)
}}

Based on the Reflexion architecture, transform this reflection into specific improvement signals for the next generation!
"""
Reflexion System Prompt:
"""You are a professional learning path reflection agent based on the Reflexion architecture, possessing the following core capabilities:

Reflexion Architecture Expertise:
- Trial & Error Analysis: Deeply analyze trial-and-error patterns in the generation process.
- Verbal Reinforcement: Transform performance feedback into verbalized improvement signals.
- Memory Integration: Integrate historical experience to form a continuously improving knowledge base.
- Self-Reflection: Identify problems and generate solutions through self-reflection.

User Adaptation Expertise:
- Deeply understand learning characteristics and need differences of different user identities.
```

```
- Precisely evaluate the degree of alignment
↪  between learning paths and user goals.
- Professionally analyze the rationality of time
↪  constraints and learning preferences.
- Generate targeted user adaptation improvement
↪  suggestions.

Reflection Philosophy (Based on Reflexion):
- Transform failures into learning opportunities,
↪  generating improvement signals through
↪  reflection.
- Focus on user adaptation as the core dimension
↪  of reflection.
- Generate actionable, concrete improvement
↪  suggestions.
- Provide clear optimization directions for the
↪  next generation.

Reflection Process:
1. Trial Analysis: Analyze performance of the
↪  current generation trial.
2. Error Identification: Identify errors and
↪  deficiencies.
3. Insight Generation: Generate deep insights and
↪  improvement suggestions.
4. Reward Calculation: Calculate reward signals
↪  for reinforcement learning.
5. Memory Update: Update experience memory for
↪  future reference.

Your reflection will become a critical input for
↪  the next path generation. Please ensure:
- Analysis is deep, specific, and actionable.
- Focus heavily on user adaptation issues.
- Provide clear improvement directions and
↪  strategies.
- Generate effective reward signals for system
↪  optimization.
"""
```

A.4 Latency Consideration

To provide a transparent analysis of the system's computational overhead, we evaluate the inference latency, token consumption, and tool call frequency of LEARNERCOMPASS compared to existing iterative baselines. As detailed in Table 8, the multi-agent collaboration and MCTS architecture inherently introduce additional computational cost.

Specifically, the full LEARNERCOMPASS (3 iterations) requires an average wall-clock time of approximately 79 seconds per path. This increased latency is primarily driven by the extensive use of external Model Context Protocol (MCP) tool calls (averaging 12.89 calls per generation), such as dynamic knowledge graph queries and web searches, which introduce significant network I/O overhead. However, this computational investment yields a substantial pedagogical return, achieving the highest overall user adaptivity score (4.67) among all evaluated methods. In the context of rigorous STEM education—where providing a pedagogically sound, highly personalized, and logically reliable learning trajectory is paramount—we argue that this efficiency-effectiveness trade-off is entirely justified. Exchanging approximately one minute of inference time for a deeply reasoned, high-quality curriculum aligns with the practical deployment requirements of asynchronous intelligent tutoring systems.

A.5 Examples of Learning Path Planning

Example1: Undergraduates with a foundation in advanced mathematics can master Gram-Schmidt orthogonalisation within a week

$Step1$: Review the definition and geometric significance of vector inner product (dot product) and orthogonality,
$Step2$: Understand the core principle of Gram-Schmidt orthogonalisation: achieving orthogonality through subtraction of projections,
$Step3$: Manually derive the steps of Gram-Schmidt orthogonalisation in three-dimensional space,
$Step4$: Master the intuitive interpretation and algorithmic workflow of QR decomposition,
$Step5$: Demonstrate the practical value of orthogonalisation through typical engineering applications (e.g., signal processing),
$Step6$: Implement the Gram-Schmidt algorithm using Python/NumPy and verify numerical stability,
$Step7$: Analyse potential error accumulation issues in the algorithm (improved QR method),
$Step8$: Complete intensive practice exercises for engineering optimisation (including parametric problem variants).

Example2: Undergraduates in artificial intelligence with a foundation in Python language learning Difference between supervised and unsupervised learning within a week

$Step1$: Activating Prior Knowledge: Foundations of Correlation Statistics and Machine Learning,
$Step2$: Establishing Conceptual Frameworks: Ontological Distinctions Between Supervised and Unsupervised Learning,
$Step3$: Analysing Algorithmic Principles: Mathematical Formula Derivation and

Table 8: Computation cost and wall-clock time evaluation (backbone: GPT-4o)

Method	**Avg. wall-clock time (s/path)**	**Token consumption (output)**	**MCP tool calls**	**Overall user adaptivity (↑)**
GPT-4o (CoT, 1 iteration)	7.21	970	–	3.63
GPT-4o (CoT, 3 iterations)	19.54	2482	–	3.71
P-Xplore (Lim et al., 2025) (1 iteration)	12.98	1292	–	3.76
P-Xplore (Lim et al., 2025) (3 iterations)	37.04	3436	–	3.89
Gen-Mentor (Wang et al., 2025b)	32.12	3711	–	4.55
LearnerCoMPASS (1 iteration)	22.91	2132	5.36	4.12
LearnerCoMPASS (3 iterations)	**79.13**	**4524**	**12.89**	**4.67**

Flowchart Visualisation,
*Step*4: Coding Practice Sessions: Implementing Classic Algorithms Using Scikit-learn,
*Step*5: Data Modelling Comparisons: Handwritten Digit Recognition and Iris Clustering Experiments,
*Step*6: Mapping Business Scenarios: Case Studies in Financial Risk Control (Supervised) and Customer Segmentation (Unsupervised),
*Step*7: Tracing Algorithmic Lineage: Classification and Evolution of Supervised/Unsupervised Learning Algorithms,
*Step*8: Comparing Evaluation Metrics: Analysing Accuracy, Interpretability, and Data Requirements,
*Step*9: Designing Hybrid Systems: Concept Transfer in Semi-Supervised Learning Architectures,
*Step*10: Drafting Technical Reports: Deepening Conceptual Understanding Through Critical Writing.

Example3: Graduate electrical engineering students with a background in calculus and linear algebra — Mastering Fourier Transform theory and signal processing applications within two weeks

Step 1: Activating Prior Knowledge: Reviewing complex numbers, Fourier series, linear time-invariant (LTI) systems, and the relationship between Laplace and Fourier transforms,
Step 2: Establishing Conceptual Foundations: Intuitive comparison between time-domain and frequency-domain representations, including amplitude spectra, phase spectra, and physical interpretations,
Step 3: Mathematical Derivations and Properties: Deriving the continuous Fourier transform (FT), discrete-time Fourier transform (DTFT), and discrete Fourier transform (DFT); proving key properties such as linearity, time shifting, modulation, convolution, and duality,
Step 4: Algorithmic and Numerical Implementation: Understanding the Fast Fourier Transform (FFT) algorithm and its computational complexity; implementing DFT and FFT using NumPy and SciPy, with attention to numerical precision and windowing effects,
Step 5: Time-Frequency Analysis Extensions: Studying short-time Fourier transform (STFT), window function selection, spectral leakage, and zero-padding; comparing STFT with wavelet transforms for different signal characteristics,
Step 6: Digital Filter Design Practice: Designing and implementing ideal and practical low-pass, high-pass, band-pass, and notch digital filters; analyzing phase response and group delay effects on signal reconstruction,
Step 7: Programming Experiments: Completing three coding tasks—(1) synthesizing and analyzing multi-frequency sinusoidal signals, (2) applying FFT-based noise reduction to real audio or ECG signals, and (3) comparing different window functions and STFT parameter settings,
Step 8: Mapping Application Scenarios: Exploring concrete use cases of Fourier analysis in communications (modulation and demodulation), speech processing (spectral analysis and feature extraction), biomedical signals (ECG/EEG analysis), and mechanical vibration diagnostics,
Step 9: Performance Evaluation and Experimental Design: Defining evaluation metrics such as signal-to-noise ratio, spectral resolution, and reconstruction error; designing controlled experiments to assess robustness under varying noise levels, sampling rates, and window lengths,
Step 10: Technical Reporting and Further

Learning: Writing a technical report that documents methodology, experimental results, reproducible code snippets, and engineering insights; outlining follow-up topics including multi-resolution analysis, wavelet packets, compressed sensing, and deep spectral estimation.

A.6 Human Evaluation Questionnaire and Criteria

In this section, we present the complete questionnaire used for the human evaluation of the learning paths generated by LEARNERCOMPASS. This evaluation was conducted to assess the system's performance across five key dimensions: Personalized Adaptability (Pers.), Path Quality (Qual.), Predicting User Interest (Inte.), Predicting Learning Efficiency (Effi.), and Predicting Sustainability (Sust.).

A.6.1 Evaluation Protocol

Purpose The primary objective of this evaluation is to quantify the pedagogical validity and user-centric effectiveness of the generated learning trajectories. We aim to compare the output of LEARNERCOMPASS against baseline models in terms of logical coherence, alignment with learner goals, and potential for long-term skill retention.

Methodology Evaluators (recruited subject matter experts and students) were presented with a learner profile S_0 (initial state), a learning goal G, and a generated path $\mathcal{P}$. For each criterion, evaluators rated their agreement with specific statements using a **5-point Likert Scale**:

- **1 (Strongly Disagree):** The path completely fails to meet the criteria; contains critical errors.
- **2 (Disagree):** The path has significant flaws or is largely irrelevant.
- **3 (Neutral):** The path is acceptable/passable but lacks depth or specific optimization.
- **4 (Agree):** The path is good, meeting the criteria with only minor imperfections.
- **5 (Strongly Agree):** The path is excellent, perfectly tailored, and logically rigorous.

Data Usage and Consent Prior to the evaluation, all participants provided informed consent. They were informed that:

1. Participation was voluntary and compensated.
2. All evaluation data would be anonymized and used solely for academic research purposes.
3. No Personally Identifiable Information (PII) regarding the evaluators would be collected or disclosed.

A.6.2 Questionnaire Instrument

Table 9 displays the detailed rubric used for the evaluation.

A.7 Human Judge Results

To strictly assess the pedagogical validity and robustness of LEARNERCOMPASS in handling rigorous STEM content, we conducted a fine-grained human evaluation focusing on learning path generation for 18 core knowledge concepts in Linear Algebra. As detailed in Table 10, subject matter experts evaluated the generated paths across five distinct dimensions: Personalized Adaptability (Pers.), Path Quality (Qual.), Predicting User Interest (Inte.), Predicting Learning Efficiency (Effi.), and Predicting Sustainability (Sust.).

The quantitative results demonstrate the system's superior performance, achieving an **Overall Average score of 4.45** (Std. Dev 0.16) on a 5-point scale. Notably, the model excelled in planning paths for foundational concepts, with "Basis & Dimension" achieving the highest mean score of **4.76**, followed closely by "Cross Product" (4.73) and "Inner Product Space" (4.73). These high scores in *Path Quality* and *Efficiency* indicate that the system effectively constructs logically coherent and pedagogically sound sequences. While performance on highly abstract concepts like "Inverse Matrix" (3.30) and "Linear Independence" (3.52) showed slightly higher variance, the scores remain well above the acceptable threshold, suggesting that the system maintains reliability even when dealing with complex conceptual dependencies.

A.8 Implementation Details, Computational Budget, and Artifact Licenses

In this section, we provide a detailed disclosure of the computational resources, model specifications, data artifacts, and licensing terms used in this study to ensure transparency and reproducibility.

A.8.1 Computational Infrastructure and Model Details

Our experimental framework leverages a hybrid infrastructure combining high-performance local

Table 9: The 5-point Likert Scale Questionnaire used for Human Evaluation. Evaluators rated each item from 1 (Strongly Disagree) to 5 (Strongly Agree).

ID	Dimension	Evaluation Statement / Criterion	Scale
Part I: Personalized Adaptability (Pers.)			
A1	Identity Match	The path's content and tone are highly appropriate for the learner's specific background (e.g., major, career role).	1–5
A2	Goal Alignment	The generated path directly addresses the learner's stated objectives without deviating to irrelevant topics.	1–5
A3	Time Feasibility	The recommended workload and schedule are realistic and achievable within the learner's constraints.	1–5
A4	Preference Accom.	The path respects the learner's specific preferences (e.g., resource types) and learning style.	1–5
Part II: Path Quality (Qual.)			
B1	Logical Progression	The sequence of learning nodes follows a strictly logical order (prerequisites appear before advanced concepts).	1–5
B2	Completeness	The path covers all necessary knowledge points required to achieve the goal, with no critical gaps.	1–5
B3	Practicality	The suggested steps and resources are actionable, accessible, and practically useful.	1–5
B4	Difficulty Approp.	The complexity of the content is appropriate for the learner's current proficiency level.	1–5
Part III: Predicting User Interest (Inte.)			
C1	Contextual Relevance	The examples or scenarios used are highly relevant to the learner's real-world context or interests.	1–5
C2	Adaptive Support	The path provides appropriate scaffolding or support resources where the learner might face challenges.	1–5
C3	Motivation Align.	The path structure is designed to stimulate and maintain the learner's intrinsic motivation.	1–5
Part IV: Predicting Learning Efficiency (Effi.)			
D1	Time Efficiency	The path avoids redundancy and offers the most direct route to skill mastery.	1–5
D2	Milestone Clarity	The learning milestones are clearly defined, allowing the learner to track progress easily.	1–5
Part V: Predicting Sustainability (Sust.)			
E1	Engagement Poten.	The path design encourages the learner to continue studying over the long term rather than dropping out.	1–5
E2	Transferability	The skills and concepts structured in this path appear conducive to transfer to future related tasks.	1–5

computing with API-based inference for proprietary Large Language Models (LLMs).

Model Deployments We employed a diverse set of State-of-the-Art (SOTA) LLMs to function as backbone agents and baselines within the LEARNERCOMPASS framework:

- **API-Based Models:** The following models were accessed via secure API endpoints:
  - **GPT-5.2** (OpenAI): Utilized primarily as the "LLM-as-a-Judge" for semantic evaluation due to its high correlation with human judgment.
  - **GPT-4o** (OpenAI) and **Claude-3.7-Sonnet** (Anthropic): Served as robust baselines and expert agents in the path planning module.
  - **Deepseek-R1** and **Qwen3-235B**: Employed to evaluate the framework's performance with open-weight models accessible via API.
- **Local Deployment:** The **Llama-3.3-70B-Instruct** model was deployed locally to assess performance in a private, offline environment. Inference was accelerated using **8 × NVIDIA A100 (80GB) GPUs** interconnected via NVLink.

Computational Budget and Token Estimation Given the multi-agent nature of our architecture—

Table 10: Detailed Human Evaluation Results of Learning Path Generation Across 18 Linear Algebra Concepts (5-Point Scale).

Knowledge Concepts	Judge Scores					Statistics	
	J-1(Pers.)	J-2(Qual.)	J-3(Inte.)	J-4(Effi.)	J-5(Sust.)	Mean	Std. Dev
Basis & Dimension	4.85	4.75	4.80	4.75	4.75	**4.76**	0.07
Cross Product	4.75	4.75	4.75	4.75	4.65	4.73	0.04
Inner Product Space	4.65	4.75	4.75	4.80	4.70	4.73	0.05
Vector Operations	4.75	4.75	4.75	4.75	4.65	4.73	0.04
Dot Product	4.75	4.75	4.75	4.65	4.70	4.72	0.04
Diagonalization	4.85	4.75	4.75	4.50	4.70	4.71	0.12
LU Decomposition	4.70	4.75	4.75	4.65	4.70	4.71	0.04
Gram-Schmidt Process	4.80	4.75	4.50	4.75	4.70	4.70	0.11
Vector Space	4.75	4.75	4.55	4.75	4.50	4.66	0.11
Determinant	4.75	4.75	4.50	4.50	4.55	4.61	0.11
Linear Systems	4.75	4.75	4.50	4.45	4.60	4.61	0.12
Eigenvalues	4.25	4.75	4.50	4.50	4.60	4.52	0.16
Noether's Theorem	4.20	4.50	4.80	4.75	4.35	4.52	0.23
Orthogonality	4.00	4.50	4.60	4.55	4.45	4.42	0.22
Linear Transformation	4.60	3.75	4.25	4.40	4.25	4.25	0.28
Matrices	4.20	3.75	4.25	4.40	4.20	4.16	0.22
Gaussian Elimination	4.10	3.75	4.25	4.40	4.30	4.16	0.23
Linear Independence	3.75	2.75	4.00	4.00	3.10	3.52	0.51
Inverse Matrix	3.00	3.25	3.75	3.00	3.50	3.30	0.29
Overall Average	4.44	4.38	4.53	4.44	4.46	**4.45**	0.16

which involves recursive calls between the Assessment, Summary, Predict, and Tracing Agents—the token consumption is substantial. We estimate the total computational budget for all experiments (including hyperparameter tuning, ablation studies, and the final evaluation across three datasets) as follows:

- **Total Estimated Volume:** Approximately **85 million tokens**.
- **Breakdown:**
  - *Path Planning (AB-MCTS-M):* $\approx 60\%$ of usage (due to the extensive search space and long-horizon generation).
  - *Cognitive Diagnosis (Encoder-Decoder):* $\approx 25\%$ of usage (iterative refinement of cognitive vectors).
  - *Evaluation (LLM-as-a-Judge):* $\approx 15\%$ of usage (using GPT-5.2 for scoring 1,000+ samples).

A.8.2 Scientific Artifacts and Licensing

We confirm that all datasets and benchmarks used in this work are publicly available and were utilized in strict accordance with their respective licenses.

Artifact Descriptions and Usage

- **Junyi Academy Dataset:** A large-scale educational dataset containing student learning logs. We utilized the exercise-concept mapping relations to initialize our Static Macro-Knowledge Graph.
- **AI-Generated Skill Metadata & Resume Dataset:** Publicly accessible datasets hosted on Kaggle. These were used to evaluate the system's generalization capabilities.
- **RAGTruth:** An open-source hallucination detection benchmark available on GitHub. We adapted its annotation methodology to rigorously quantify the "hallucination rate" of our generated learning paths.

A.8.3 Intended Use and Compliance

We explicitly declare that all scientific artifacts were used solely for their intended purposes:

1. **Research Purpose:** The datasets were used exclusively for academic research tasks, specifically Educational Data Mining (EDM) and Recommender System evaluation. No data was used for commercial product development.

Table 11: Details of Scientific Artifacts and Licenses.

Artifact Name	Source / Accessibility	License Type
Junyi Academy Dataset	Public Benchmark	CC-BY-NC-SA 4.0
AI-Generated Skill Metadata	Kaggle (Public)	CC BY 4.0
Resume Dataset	Kaggle (Public)	CC0: Public Domain

2. **Privacy Preservation:** All employed datasets (Junyi, Resume) are anonymized benchmarks that do not contain Personally Identifiable Information (PII). Our internal processing pipeline further ensures that no user-specific data is retained beyond the inference session.

3. **Content Safety:** The LEARNERCOMPASS framework includes a built-in verification mechanism (as detailed in Section 3.1) to filter out potential biases or unsafe content from LLM generations, aligning with the principles of Responsible NLP.