One LLM Does Not Simulate All Students: Ability-Aware Student Simulation via Cognitive Diagnosis Guided LLM Assignment

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have become integral to personalized education systems, particularly in the realm of student behavior simulation. By predicting fine-grained learning behaviors, these simulations enable intelligent systems to provide tailored instructional support. However, most existing methods rely on a single high-capacity LLM to represent an entire population of diverse learners. In this work, we demonstrate that this "one-size-fits-all" approach induces a systematic *ability-dependent bias*, where high-capacity models tend to overestimate low-ability students while lower-capacity models underestimate high-ability ones. To mitigate this distortion, we propose an **ability-aware student simulation framework** that dynamically matches students with appropriate LLM backbones through cognitive alignment. We leverage Neural Cognitive Diagnosis (NeuralCD) to extract multidimensional cognitive profiles for both human students and LLM agents within a shared skill space, subsequently pairing each student with the most cognitively representative model. Extensive experiments demonstrate that our approach substantially reduces simulation bias and consistently outperforms single-model baselines across the entire proficiency spectrum. Our findings suggest that faithful behavior simulation necessitates the **alignment of model capacity with student ability**, establishing cognitive diagnosis as a principled mechanism for model assignment in educational AI.

1 Introduction

Personalized learning is widely recognized as an effective paradigm for improving learning outcomes by adapting instructional content and strategies to individual students (Bernacki et al., 2021; Shemshack and Spector, 2020; Jian, 2023). With the rapid advancement of large language models (LLMs), recent studies have increasingly incorporated LLMs into personalized learning systems

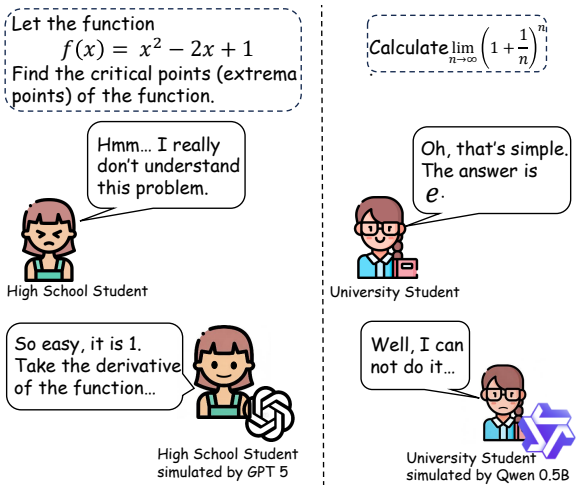

Figure 1: Different ways to simulate students. The left shows high-capacity LLMs simulating low cognitive ability students, while the right shows low-capacity LLMs simulating high cognitive ability students.

(Zhang et al., 2025; Wen et al., 2024; Neumann et al., 2024). Among these applications, *student behavior simulation* has attracted growing attention (Gao et al., 2025; Lu and Wang, 2024; Xu et al., 2024), where LLMs act as agents to predict students' future learning behaviors from historical interactions.

Despite promising empirical results, existing LLM-based student simulation methods suffer from a fundamental limitation. Most prior work relies on a *single LLM*—typically a high-capacity model such as ChatGPT (Xu et al., 2025; Abbasiantaeb et al., 2024)—to simulate all students. This approach assumes that a strong LLM can faithfully represent learners across the entire ability spectrum, which is inherently problematic. Due to their pre-trained knowledge, LLMs exhibit an uncontrollable *competence prior* that can leak into simulations and undermine cognitive fidelity. As illustrated in Figure 1, when simulating a struggling high school student solving a basic math problem, a powerful model such as GPT-5 tends to produce

correct answers, even in cases where the student's historical interaction data suggest that the relevant concept has not been mastered. Conversely, low-capacity LLMs frequently lack the domain knowledge required to simulate advanced learners. This mismatch leads to systematic ability bias, overestimating low-ability students while underestimating high-ability ones, thereby distorting simulated learning behaviors.

Given the diverse landscape of LLMs, ranging from lightweight to large-scale systems (Bai et al., 2023; Achiam et al., 2023; Liu et al., 2024; Touvron et al., 2023; Team et al., 2023), a natural alternative is to leverage a heterogeneous pool of LLMs to represent students with varying cognitive abilities. This raises a fundamental question: **How can a student's latent cognitive state be systematically aligned with the most representative LLM backbone to ensure a faithful simulation?** While research on *LLM assignment* (Ding et al., 2024; Xia et al., 2024; Feng et al., 2024) is growing, it primarily focuses on task-driven optimization, aiming to find a cost-effective model combination that maximizes task success within a budget (Panda et al., 2025; Song et al., 2025; Ashury Tahan et al., 2024). This objective is misaligned with educational simulation, where the goal is not *utility-optimal performance* but *cognitive ability alignment*. A faithful simulation must reflect a student's specific cognitive constraints, persistent misconceptions, and actual skill mastery, even when it leads to incorrect answers. Therefore, the selection criterion must shift from maximizing task success to minimizing the cognitive ability gap between the LLM agent and the human learner.

To address this challenge, we propose an **ability-aware student simulation framework** grounded in **Cognitive Diagnosis (CD)** theory (Templin and Henson, 2006; Leighton and Gierl, 2007). Rooted in psychometrics, CD models infer learners' latent cognitive states by mapping observable behaviors to fine-grained skill mastery. Building on this foundation, we employ *Neural Cognitive Diagnosis (NeuralCD)* (Wang et al., 2022), a representative deep learning-based CD model, to estimate students' cognitive ability profiles from historical interaction data, while simultaneously deriving skill-level performance profiles for candidate LLMs. By matching students and LLMs based on their similarity, we assign each student the LLM that best reflects their cognitive ability, rather than their task-solving potential.

Contributions Our main contributions are summarized as follows:

- **Systematic Bias Analysis**: We empirically demonstrate that using a single LLM to simulate diverse students induces systematic ability bias, failing to capture the heterogeneity of learners.
- **Ability-Aware Assignment Framework**: We propose a novel framework that leverages NeuralCD to facilitate student–LLM matching, moving beyond task-driven selection toward *cognitive-ability alignment*.
- **Improved Simulation Fidelity**: Extensive experiments show that our approach consistently outperforms single-LLM baselines, achieving higher simulation fidelity.

2 Related Work

2.1 LLM-based Agent Simulation in Education

The simulation of LLM-based agents is gradually gaining momentum (Park et al., 2023; Man et al., 2025; Yang et al., 2025). In the educational domain, these simulations have proliferated to encompass diverse scenarios such as learning process simulation (Mannekote et al., 2025; Gao et al., 2025; Xu et al., 2024)and pedagogical interactions (Zheng et al., 2025; Lv et al., 2025). To achieve human-like behavior, most existing agent-based simulations adopt a *modular architecture* paradigm (Chu et al., 2025; Bhowmik et al., 2024). Researchers typically employ a single, high-ability LLM as the central brain and augment it with specialized components, such as memory modules, planning modules and reflection components for self-correction (Zheng et al., 2025; Arana et al., 2025). However, these structural augmentations do not mitigate the inherent competence prior of the underlying LLMs, often leading to simulated behaviors that remain unaligned with the actual cognitive state of students.

In contrast, our work shifts the focus from structural modularity to **ability-aware alignment**. Instead of relying on a single model with complex external modules, we dynamically select an appropriate LLM for each student based on their diagnosed cognitive profile. This approach ensures that the simulation is grounded in the intrinsic capacity of the agent itself, thereby facilitating a more authentic reflection of student-specific behaviors and learning outcomes.

2.2 Cognitive Diagnosis Models

Cognitive Diagnosis (CD) models aim to infer the latent cognitive states of learners, particularly within the context of educational assessment. Classical paradigms including DINA (De La Torre, 2009), IRT (Lord, 2012), and MIRT (Reckase, 2009) estimate student abilities through probabilistic frameworks. While theoretically well-founded, these methods often rely on predefined parametric functions such as 1PL and 2PL models (DeMars, 2010). This reliance constrains their flexibility and scalability when processing complex or large-scale educational datasets.To address these limitations, deep learning-based CD models, such as NeuralCD (Wang et al., 2022) and RCD (Gao et al., 2021), utilize neural networks to capture complex student-exercise interactions from large-scale response data. Building on this paradigm, we employ NeuralCD to map both human students and LLM agents into a unified latent space by treating LLMs as "artificial learners". This shared cognitive dimensionality allows us to directly quantify the mastery gap between students and models, facilitating precise ability-aware alignment.

2.3 Large Language Model Assignment

Research on LLM model assignment has gained momentum as a strategy to balance inference costs with task performance (Mei et al., 2025; Song et al., 2025; Ding et al., 2024). These methods typically function as routers that assign queries to appropriate models based on task difficulty. Such approaches are primarily task-driven (Dai et al., 2024; Zhao et al., 2024; Ashury Tahan et al., 2024), prioritizing the minimization of computational overhead or latency while maintaining a specific threshold of solution quality (Hu et al., 2024; Jitkrittum et al., 2025).

Our work differs fundamentally from this paradigm by shifting the focus from task-oriented routing to profile-oriented alignment for simulation. Rather than optimizing for efficiency, we prioritize behavioral fidelity by matching the intrinsic capabilities of an LLM with the diagnosed cognitive profiles of the target learners.

3 Method

3.1 Task Formulation

Let $\mathcal{S}$ denote the set of students, $\mathcal{E}$ the set of exercises, and $\mathcal{K}$ the set of knowledge concepts. For each student $s \in \mathcal{S}$, the historical interaction sequence is denoted as $I_s = \{(e_1, y_{s,e_1}), \ldots, (e_n, y_{s,e_n})\}$, where each exercise e_i is a triplet consisting of textual content $e_{i,\text{text}}$, associated concepts $e_{i,\text{concept}} \subseteq \mathcal{K}$, and the ground-truth answer $e_{i,\text{ans}}$. The response $y_{s,e_i} \in \{0, 1\}$ indicates whether the student's answer was correct.

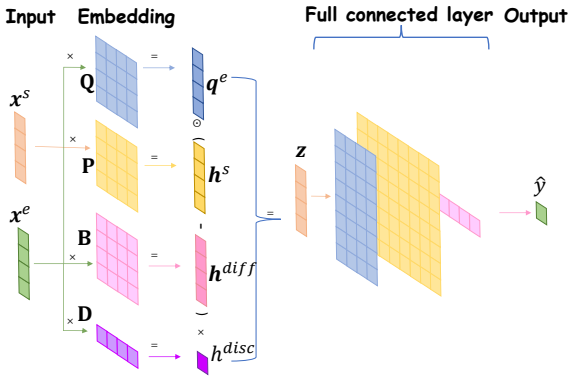

Figure 2: The architecture of NeuralCD.

In this work, we redefine the student simulation task as a cognitive-aligned model assignment problem. Beyond the student data, we introduce a heterogeneous pool of Large Language Models $\mathcal{M} = \{m_1, m_2, \ldots, m_k\}$ with varying capacities. Our objective is to develop a framework that consists of two key components:

Cognitive Profiling: A diagnostic function $f_D : I_s \to \boldsymbol{\alpha}_s$ that maps a student's history into a latent cognitive profile $\boldsymbol{\alpha}_s \in [0, 1]^{|\mathcal{K}|}$, representing their mastery levels across all concepts.

Ability-Aware Assignment: An alignment function $f_A : (\boldsymbol{\alpha}_s, \mathcal{M}) \to m^*$ that selects the optimal LLM $m^* \in \mathcal{M}$ whose intrinsic capability profile most closely mirrors the student's cognitive state $\boldsymbol{\alpha}_s$.

The final goal is to utilize the assigned model m^* to simulate the student's future behaviors. A successful simulation ensures that the response $\hat{y}_{s,e_{new}}$ generated by m^* is not only accurate in terms of prediction but also reflects the underlying cognitive constraints and misconceptions of student s.

3.2 Neural Cognitive Diagnosis

To project both human students and LLM agents into a unified mastery space, we employ Neural Cognitive Diagnosis (NeuralCD) (Wang et al., 2022) as our underlying diagnostic engine. This model allows us to transform observable response data into latent, multidimensional ability profiles. The NeuralCD model architecture is show in Figure 2.

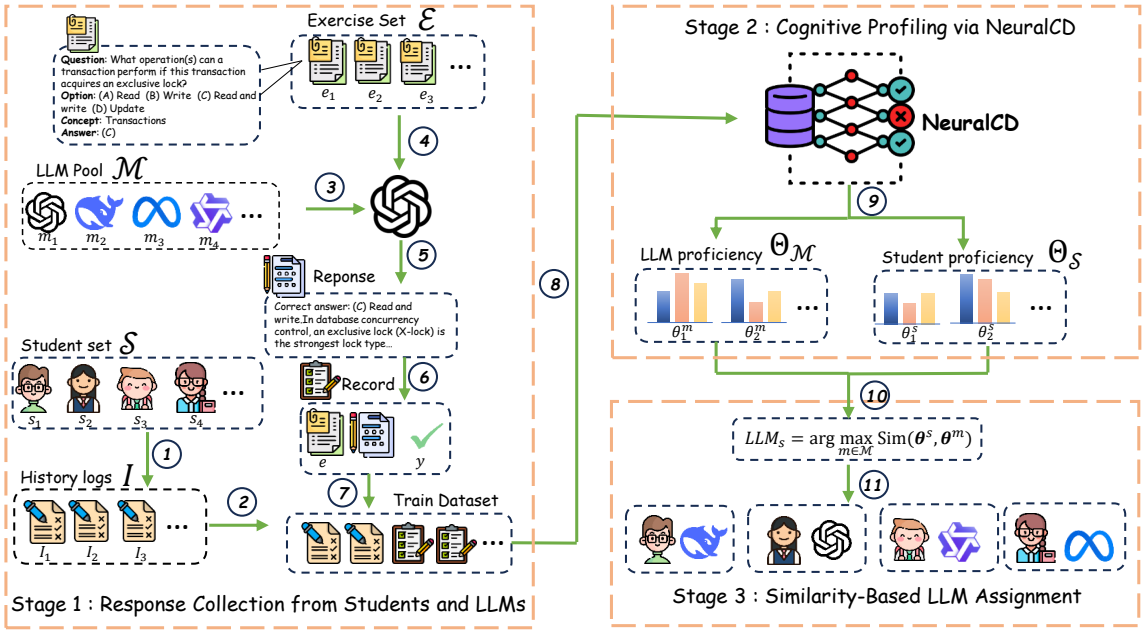

Figure 3: The overview of adaptive LLM assignment for student simulation.

Each student s is encoded as a binary *one-hot* vector $\boldsymbol{x}^s \in \{0,1\}^{1\times|\mathcal{S}|}$, where exactly one entry is set to 1 to indicate the student's identity. Based on this representation, NeuralCD learns a student-specific knowledge concept proficiency vector $\boldsymbol{h}^s \in (0,1)^{1\times|\mathcal{K}|}$ via:

$$\boldsymbol{h}^s = \sigma\left(\boldsymbol{x}^s \mathbf{P}\right) \tag{1}$$

where $\sigma(\cdot)$ denotes the sigmoid function and $\mathbf{P} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{K}|}$ is a trainable parameter matrix.

Similarly, each exercise $e \in \mathcal{E}$ is represented by a binary *one-hot* vector $\boldsymbol{x}^e \in \{0,1\}^{1\times|\mathcal{E}|}$. We further adopt a predefined Q-matrix $\mathbf{Q} \in \{0,1\}^{|\mathcal{E}|\times|\mathcal{K}|}$ to encode the associations between exercises and knowledge concepts, where $\mathbf{Q}_{ij} = 1$ indicates that exercise i involves the j-th knowledge concept. Using the Q-matrix, the concept-level representation of exercise e is computed as:

$$\boldsymbol{q}^e = \boldsymbol{x}^e \mathbf{Q} \tag{2}$$

To characterize exercise properties, NeuralCD estimates both *concept-level difficulty* and *exercise discrimination*. Specifically, the difficulty vector $\boldsymbol{h}^{diff} \in (0,1)^{1\times|\mathcal{K}|}$ and the discrimination scalar $h^{disc} \in (0,1)$ are computed as:

$$\boldsymbol{h}^{diff} = \sigma\left(\boldsymbol{x}^e \mathbf{B}\right), \quad h^{disc} = \sigma\left(\boldsymbol{x}^e \mathbf{D}\right) \tag{3}$$

where $\mathbf{B} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{K}|}$ and $\mathbf{D} \in \mathbb{R}^{|\mathcal{E}|\times 1}$ are trainable parameter matrices.

Following MIRT (Chalmers, 2012), NeuralCD integrates student proficiency and exercise characteristics to construct the input to the prediction network:

$$\boldsymbol{z} = \boldsymbol{q}^e \odot \left(\boldsymbol{h}^s - \boldsymbol{h}^{diff}\right) \times h^{disc} \tag{4}$$

where $\odot$ denotes element-wise product.

The resulting vector $\boldsymbol{z}$ is then fed into a multi-layer fully connected neural network:

$$\boldsymbol{f}_1 = \phi\left(\mathbf{W}_1 \boldsymbol{z}^T + \boldsymbol{b}_1\right) \tag{5}$$
$$\boldsymbol{f}_2 = \phi\left(\mathbf{W}_2 \boldsymbol{f}_1 + \boldsymbol{b}_2\right) \tag{6}$$
$$\hat{y} = \phi\left(\mathbf{W}_3 \boldsymbol{f}_2 + b_3\right) \tag{7}$$

where $\phi(\cdot)$ denotes the activation function and $\hat{y}$ represents the predicted probability that student s correctly answers exercise e.

The model is trained by minimizing the binary cross-entropy loss:

$$\mathcal{L} = -\sum_i \left[r_i \log \hat{y}_i + (1 - r_i) \log(1 - \hat{y}_i)\right] \tag{8}$$

where $r_i \in \{0,1\}$ denotes the ground-truth response correctness.

After training, the learned parameter matrix $\mathbf{P}$ enables us to infer each student's knowledge concept proficiency vector $\boldsymbol{h}^s$ via Equation 1, which serves as the cognitive profile for subsequent LLM assignment and student behavior simulation.

3.3 Adaptive LLM Assignment for Student Simulation

The core component of our framework is an **adaptive LLM assignment module**, which assigns each student an appropriate LLM to serve as the simulation backbone. Instead of relying on a single LLM to simulate all students, our approach explicitly accounts for heterogeneity in student proficiency by matching students with LLMs of comparable cognitive profiles.

The assignment procedure consists of three sequential stages: (1) collecting response records from both students and LLMs, (2) estimating their knowledge concept proficiency vectors via NeuralCD, and (3) matching each student with the most suitable LLM based on proficiency similarity.

Stage 1: Response Collection from Students and Candidate LLMs. For students, we directly collect their historical response records. In addition, we define a candidate set of LLMs $\mathcal{M}$ to serve as potential simulation backbones. Since LLMs do not possess prior interaction histories, we prompt each LLM $m \in \mathcal{M}$ to answer all exercises in the dataset. We employ a chain-of-thought (CoT) prompting strategy (Wei et al., 2022) to elicit structured reasoning and record the final answer correctness. As a result, we obtain a train dataset that includes both real student responses and LLM-generated responses over the same set of exercises.

Stage 2: Cognitive Profiling via NeuralCD. Given the combined response records from students and candidate LLMs, we train a NeuralCD model to infer latent cognitive profiles. By treating LLMs as pseudo-students during training, NeuralCD provides a unified and interpretable representation of both human learners and LLMs under the same diagnostic framework. To maintain consistency with the parameterization of cognitive diagnosis models (Baker, 2001; Liu et al., 2023), we denote the student ability representation $\boldsymbol{h}^s$ produced by NeuralCD as $\boldsymbol{\theta}$. Specifically, NeuralCD produces a student proficiency set $\boldsymbol{\Theta}_{\mathcal{S}} = \{\boldsymbol{\theta}^{s_1}, \boldsymbol{\theta}^{s_2}, \ldots, \boldsymbol{\theta}^{s_{|\mathcal{S}|}}\}$ and an LLM proficiency set $\boldsymbol{\Theta}_{\mathcal{M}} = \{\boldsymbol{\theta}^{m_1}, \boldsymbol{\theta}^{m_2}, \ldots, \boldsymbol{\theta}^{m_{|\mathcal{M}|}}\}$.

Stage 3: Similarity-Based LLM Assignment. For each student s, we compare the student's proficiency vector with those of all candidate LLMs in $\mathcal{M}$ and select the most similar one LLM_s as the simulation backbone. We adopt a similarity-based criterion to select LLM_s:

$$\text{LLM}_s = \arg\max_{m \in \mathcal{M}} \text{Sim}\left(\boldsymbol{\theta}^s, \boldsymbol{\theta}^m\right) \tag{9}$$

where $\boldsymbol{\theta}^s$ and $\boldsymbol{\theta}^m$ denote the knowledge concept proficiency vectors of student s and LLM m, respectively, and $\text{Sim}(\cdot)$ represents a similarity function. In this work, we instantiate $\text{Sim}(\cdot)$ as cosine similarity.

This assignment strategy ensures that each student is simulated by an LLM whose cognitive proficiency profile is most compatible with the student's own abilities.

3.4 Student Simulation Agent

As the primary focus of this work is *adaptive LLM selection* rather than student agent design, we adopt an LLM-based student simulation framework from prior studies (Gao et al., 2025; Mannekote et al., 2025; Xu et al., 2024). Each student agent consists of a memory module, an action module, and a reflection module. Details of the agent framework are provided in Appendix A.

Memory Module. The memory module includes both short-term and long-term memory. Short-term memory stores the student's recent response records. Long-term memory contains the student's ability profile inferred via NeuralCD as well as reinforced short-term memories.If short-term records that occur more frequently than a predefined threshold are promoted into long-term memory.

Action Module. The action module models the student's exercise-solving behavior. Before answering an exercise, the agent retrieves relevant records from short-term memory and further extracts related long-term memories based on similarity. This process includes identifying the knowledge concepts involved in the exercise to assist problem solving. The agent then generates an answer and performs a self-assessment to judge whether the response is correct.

Reflection Module. The reflection module enables the student agent to reflect on its behavior by comparing the LLM-generated response with the ground-truth student response. Reflection is triggered only when the two are inconsistent. In such cases, the reflection outcome, together with the response record, is written into short-term memory.

By adopting a standard student agent architecture, our framework isolates the effect of adap-

tive LLM selection, ensuring that performance differences arise from the suitability of the selected LLMs rather than agent design choices.

4 Experiments

4.1 Dataset Description

We conduct experiments on the public DBE-KT22 dataset (Abdelrahman et al., 2022), which contains rich educational interaction records including exercise texts, associated knowledge concepts, and student responses. The dataset consists of 1,361 students, 212 exercises, and 98 knowledge concepts, with a total of 167,222 student–exercise interaction records. Such characteristics make DBE-KT22 suitable for evaluating student behavior simulation under heterogeneous cognitive profiles.

4.2 Experiement Set Up

NeuralCD Training Configuration. For each student, their interaction records are chronologically split, with the first 80% used to train the NeuralCD model. During training, all LLM-generated responses corresponding to these exercises are included in the training set, allowing the model to learn from both student behaviors and LLM outputs. The remaining 20% of student interactions are reserved as ground-truth for evaluating LLM simulation. This setup ensures that the evaluation is performed on unseen student behaviors, avoiding information leakage. Detailed training hyperparameters are reported in Appendix B.

LLM Inference Settings. For all LLMs, we set the temperature to 0 to ensure reproducibility, except for GPT-5-Mini [1], which does not support a configurable temperature and uses a fixed default temperature of 1.

Student Selection for LLM Simulation. Due to the computational and cost constraints associated with large-scale LLM-based simulation, we conduct experiments on a carefully selected subset of students. For every student s, we compute the mean of $\boldsymbol{\theta}^s$ as student's overall proficiency score. Then we partition students into three groups according to its empirical quantiles: students whose scores fall within the 0–33rd percentile are categorized as low-proficiency, those within the 33–66th percentile as medium-proficiency, and those within the 66–100th percentile as high-proficiency. From each group, we randomly sample 100 students, resulting in a total of 300 students for evaluation. The IDs of the selected students are reported in Appendix C for reproducibility.

[1] GPT-5-Mini (2025-08-07). The specific model types and configurations are detailed in Appendix D.

4.3 LLM Pool and Baselines

We construct an LLM pool consisting of 35 language models with diverse capabilities. Detailed model configurations are provided in Appendix D.

Specifically, we evaluate three representative settings: (1) a *single strong LLM* with high model capacity, (2) a *single weak LLM* with limited capacity, and (3) our proposed *NeuralCD-guided multi-LLM selection framework*. By comparing the bias patterns exhibited by these settings, we aim to verify both the limitations of single-LLM simulation and the effectiveness of adaptive LLM assignment.

4.4 Evaluation Metrics

We evaluate simulation quality by comparing LLM-generated responses with students' ground-truth answers. A naive accuracy metric, however, is insufficient due to the imbalance between correct and incorrect student responses.

Let $\mathcal{D}^s_{\text{correct}}$ and $\mathcal{D}^s_{\text{incorrect}}$ denote the sets of exercises that a student answered correctly and incorrectly, respectively. Similarly, for the student Agent (LLM simulation), let $\mathcal{D}^m_{\text{correct}}$ and $\mathcal{D}^m_{\text{incorrect}}$ denote the sets of exercises answered correctly and incorrectly by the Agent.

Based on these sets, the conditional accuracies are computed as

$$Acc^{+} = \frac{|\mathcal{D}^s_{\text{correct}} \cap \mathcal{D}^m_{\text{correct}}|}{|\mathcal{D}^s_{\text{correct}}|} \tag{10}$$

$$Acc^{-} = \frac{|\mathcal{D}^s_{\text{incorrect}} \cap \mathcal{D}^m_{\text{incorrect}}|}{|\mathcal{D}^s_{\text{incorrect}}|} \tag{11}$$

Here, Acc^{+} measures the proportion of exercises that the Agent correctly answers among the exercises that the student answered correctly, and Acc^{-} measures the proportion of exercises that the Agent incorrectly answers among the exercises that the student answered incorrectly.

To evaluate student simulation while accounting for bias between correct and incorrect responses, we adopt the ***Bias-Aware Accuracy*** *(BAA)*, inspired by Balanced Accuracy (Brodersen et al., 2010):

$$BAA = \frac{Acc^{+} + Acc^{-}}{2} \cdot \left(1 - \left|Acc^{+} - Acc^{-}\right|\right) \tag{12}$$

An effective student simulation framework should not only achieve high overall accuracy but also avoid systematic bias towards predicting either correct or incorrect responses. **The first term** of the metric captures the overall simulation accuracy across all response types, while **the second term** serves as an explicit penalty for imbalance between correct and incorrect predictions. When the accuracies of the two response types diverge, the penalty term decreases accordingly, reflecting degraded simulation quality.

5 Result

5.1 Bias Analysis of Single LLM Simulation

To better understand the intrinsic simulation bias of LLMs, we analyze how model capacity affects the accuracy of simulating correct and incorrect student responses. Specifically, we compare Acc^{+} and Acc^{-} across LLMs with different parameter scales [2]. We also conduct case study in E.

As shown in Figure 4, we observe a clear capacity-dependent trend. As model size increases—corresponding to stronger model capability— Acc^{+} consistently improves. In contrast, Acc^{-} steadily decreases. This indicates that LLM-based student simulation is strongly influenced by the intrinsic capability of the underlying model.

Intuitively, larger and more capable LLMs tend to overestimate students' mastery and are biased toward producing correct answers, making them less effective at reproducing incorrect student behaviors. Conversely, smaller LLMs are more likely to generate incorrect responses, resulting in higher Acc^{-} but lower Acc^{+}. Therefore, selecting either a weak or a strong LLM as a single simulation backbone inevitably introduces bias: weaker models better capture incorrect behaviors, while stronger models better capture correct behaviors.

We further compare LLM-based simulation results with the LLMs' ground-truth (we calculate it through Section 3.3). The ground truth is defined as the results of the LLM's direct answers to all non-deduplicated exercises involved in the student simulation. Specifically, the ground truth corresponds to the answering accuracy in Acc^{+} table and the answering error rate in Acc^{-} table. We observe that Acc^{+} does not exceed the model's own ground truth, while Acc^{-} is higher than the corresponding ground truth. This suggests that an LLM's intrinsic capability upper-bounds its simulation performance.

Overall, these findings imply that student simulation bias originates from the inherent capability constraints of LLMs. A single LLM cannot simultaneously and faithfully simulate students across diverse proficiency levels, as its own strengths and weaknesses are inevitably reflected in the simulated behaviors.

[2]Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct, Qwen2.5-14B-Instruct, LLaMA-3.2-1B-Instruct, LLaMA-3.2-3B-Instruct,LLaMA-3.1-8B-Instruct. The specific model types and configurations are detailed in Appendix D.

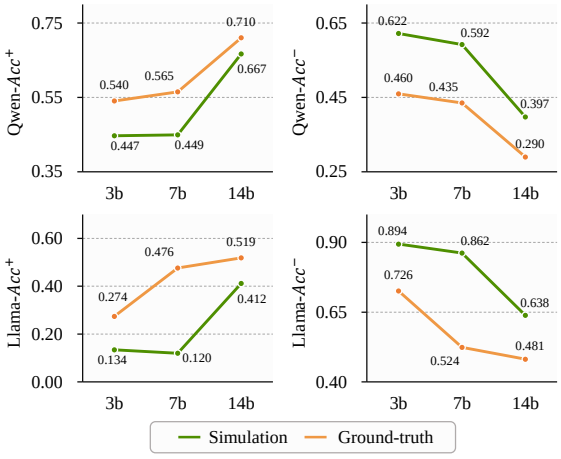

Figure 4: Acc^{+} and Acc^{-} w.r.t. ground truth across model scales for Qwen and LLaMA.

Table 1: Simulation performance across different student ability levels.

Model	BAA_{low}	BAA_{mid}	BAA_{high}	Overall
Weak LLMs				
Qwen-3B	0.427	0.417	0.463	0.441
Qwen-7B	0.476	0.440	0.439	0.446
Qwen-14B	0.360	0.398	0.394	0.389
LLaMA-1B	0.121	0.120	0.126	0.123
LLaMA-3B	0.131	0.124	0.126	0.127
LLaMA-8B	0.441	0.428	0.377	0.406
Strong LLMs				
DeepSeek	0.242	0.297	0.321	0.296
GPT-5-Mini	0.142	0.174	0.172	0.167
CD-based Method				
Ours	**0.523**	**0.516**	**0.475**	**0.505**

5.2 Effectiveness of Adaptive LLM Assignment

We further demonstrates the effectiveness of our proposed adaptive LLM assignment strategy. We report *BAA* scores for students with low, medium,

Table 2: Detailed Acc^{+} and Acc^{-} of representative LLMs across different student ability groups, where $\Delta = |Acc^{+} - Acc^{-}|$.

Model	Low Ability			Medium Ability			High Ability			Overall		
	Acc^{+}	Acc^{-}	Δ_{low}	Acc^{+}	Acc^{-}	Δ_{mid}	Acc^{+}	Acc^{-}	Δ_{high}	Acc^{+}	Acc^{-}	Δ
Weak LLMs												
Qwen-3B	0.435	0.641	0.206	0.423	0.631	0.208	0.468	0.602	0.134	0.447	0.622	0.175
Qwen-7B	0.479	0.571	0.092	0.444	0.602	0.158	0.442	0.598	0.156	0.449	0.592	0.143
Qwen-14B	0.727	0.378	0.349	0.648	0.405	0.243	0.661	0.402	0.259	0.667	0.397	0.270
LLaMA-1B	0.131	0.894	0.763	0.130	0.896	0.766	0.138	0.892	0.754	0.134	0.894	0.760
LLaMA-3B	0.119	0.846	0.727	0.121	0.870	0.749	0.119	0.863	0.744	0.120	0.862	0.742
LLaMA-8B	0.446	0.610	0.164	0.434	0.627	0.193	0.384	0.664	0.280	0.412	0.638	0.226
Strong LLMs												
GPT-5-Mini	0.871	0.149	0.722	0.821	0.172	0.649	0.834	0.175	0.659	0.835	0.168	0.667
DeepSeek	0.822	0.267	0.555	0.752	0.311	0.441	0.736	0.336	0.400	0.755	0.311	0.444
CD-based Method												
Ours	0.567	0.525	**0.042**	0.517	0.516	**0.001**	0.476	0.551	**0.075**	0.505	0.532	**0.027**

and high proficiency levels, as well as the overall performance. We compare three types of simulation settings: six weak-capacity LLMs (Qwen-3B, Qwen-7B, Qwen-14B, LLaMA-1B, LLaMA-3B, and LLaMA-8B), two strong-capacity LLMs (DeepSeek and GPT-5-Mini) [3], and our NeuralCD-guided adaptive LLM assignment method .

As illustrated in Table 1, our method achieves the superior *BAA* scores across all proficiency tiers as well as in the overall evaluation. This consistent performance gain demonstrates that adaptively matching students with cognitively aligned LLMs substantially enhances simulation fidelity, outperforming any single-model baseline across the entire learner spectrum.

We further observe a clear interaction between model capacity and student proficiency level. Most weak-capacity models perform relatively better when simulating low-proficiency students, but their *BAA* scores gradually decrease as the simulated student proficiency increases. This trend can be observed for models such as Qwen-7B and LLaMA-8B. In contrast, strong-capacity models demonstrate stronger performance when simulating high-proficiency students, with *BAA* scores increasing as student proficiency rises, as seen in DeepSeek and GPT-5-Mini.

Notably, **stronger model capacity does not necessarily translate into better simulation performance**. This observation is closely related to the simulation bias discussed in Section 5.1. As shown in Table 2, extremely strong models (e.g., GPT-5-Mini) tend to exhibit high Acc^{+} but low Acc^{-}, while very weak models (e.g., LLaMA-3B) show the opposite pattern. The large discrepancy between Acc^{+} and Acc^{-} for such models leads to suboptimal *BAA*, despite their strengths in one aspect of simulation.

These results underscore that no single LLM can serve as a universally optimal simulator across heterogeneous student populations. By leveraging cognitive diagnosis to adaptively assign LLMs based on student proficiency, our framework effectively mitigates the competence bias inherent in fixed-model paradigms, achieving superior simulation fidelity across the entire ability spectrum.

6 Conclusion

In this work, we identified the inherent biases of the single-LLM paradigm in student simulation and proposed an ability-aware framework grounded in NeuralCD. By dynamically matching human learners with appropriately capable LLM backbones, our method effectively mitigates competence-related biases and enhances simulation fidelity across diverse proficiency levels. This shift from performance-driven to alignment-oriented model selection establishes a principled foundation for more authentic educational simulations.

7 Limitations

Despite its effectiveness, our method has several limitations. First, due to cost constraints, the LLM

[3] GPT-5-Mini (2025-08-07), DeepSeek-V3.2. The specific model types and configurations are detailed in Appendix D.

pool used in this study does not exhaustively cover all existing open-source and closed-source models.

Second, our approach relies on students' historical response records to estimate their cognitive states via cognitive diagnosis models. As a result, it is not directly applicable to cold-start scenarios where no prior student interaction data is available. Addressing this limitation may require incorporating additional sources of information, such as demographic features.

Third, our evaluation primarily focuses on the accuracy and bias of simulated student responses to problem-solving tasks. We do not explicitly assess the behavioral plausibility or rationality of the simulated actions, nor do we consider other important dimensions of student behavior, such as learning strategy development, metacognitive processes, or peer collaboration. Extending the evaluation beyond answer correctness to cover richer and more realistic learning behaviors is an important direction for future work.

Finally, because student simulation itself is not the primary focus of this work, we adopt a commonly used student agent framework from prior studies to isolate the impact of LLM selection. However, these existing simulation architectures are not guaranteed to be optimal. As shown in Table 2, our method does not achieve the best overall accuracy. Improvements in student agent design may further enhance the overall performance of our framework. We leave the exploration of more advanced student simulation mechanisms to future work.

Notably, model fine-tuning (Touvron et al., 2023) is not considered as a baseline in our experiments. Fine-tuning essentially trains a new model to fit student abilities. In contrast, our approach focuses on *model assignment* rather than *model retraining*, allowing us to explicitly analyze and mitigate the inherent capacity-dependent biases of existing LLMs while preserving their original capabilities. This design choice also improves reproducibility and reduces training costs.

References

Zahra Abbasiantaeb, Yifei Yuan, Evangelos Kanoulas, and Mohammad Aliannejadi. 2024. Let the llms talk: Simulating human-to-human conversational qa via zero-shot llm-to-llm interactions. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 8–17.

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

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jasper Meynard Arana, Kristine Ann M Carandang, Ethan Robert Casin, Christian Alis, Daniel Stanley Tan, Erika Fille Legara, and Christopher Monterola. 2025. Foundations of peers: Assessing llm role performance in educational simulations. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 908–918.

Shir Ashury Tahan, Ariel Gera, Benjamin Sznajder, Leshem Choshen, Liat Ein-Dor, and Eyal Shnarch. 2024. Label-efficient model selection for text generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8384–8402, Bangkok, Thailand. Association for Computational Linguistics.

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

Frank B Baker. 2001. *The basics of item response theory*. ERIC.

Matthew L Bernacki, Meghan J Greene, and Nikki G Lobczowski. 2021. A systematic review of research on personalized learning: Personalized by whom, to what, how, and for what purpose (s)? *Educational Psychology Review*, 33(4):1675–1715.

Saptarshi Bhowmik, Luke West, Alex Barrett, Nuodi Zhang, Chih-Pu Dai, Zlatko Sokolikj, Sherry Southerland, Xin Yuan, and Fengfeng Ke. 2024. Evaluation of an llm-powered student agent for teacher training. In *European conference on technology enhanced learning*, pages 68–74. Springer.

Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. 2010. The balanced accuracy and its posterior distribution. In *2010 20th international conference on pattern recognition*, pages 3121–3124. IEEE.

R Philip Chalmers. 2012. mirt: A multidimensional item response theory package for the r environment. *Journal of statistical Software*, 48:1–29.

Zhendong Chu, Shen Wang, Jian Xie, Tinghui Zhu, Yibo Yan, Jinheng Ye, Aoxiao Zhong, Xuming Hu, Jing Liang, Philip S Yu, and 1 others. 2025. Llm agents for education: Advances and applications. *arXiv preprint arXiv:2503.11733*.

Xiangxiang Dai, Jin Li, Xutong Liu, Anqi Yu, and John Lui. 2024. Cost-effective online multi-llm selection with versatile reward models. *arXiv preprint arXiv:2405.16587*.

Jimmy De La Torre. 2009. Dina model and parameter estimation: A didactic. *Journal of educational and behavioral statistics*, 34(1):115–130.

Christine DeMars. 2010. *Item response theory*. Oxford University Press.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks VS Lakshmanan, and Ahmed Hassan Awadallah. 2024. Hybrid llm: Cost-efficient and quality-aware query routing. *arXiv preprint arXiv:2404.14618*.

Tao Feng, Yanzhen Shen, and Jiaxuan You. 2024. Graphrouter: A graph-based router for llm selections. *arXiv preprint arXiv:2410.03834*.

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.

Weibo Gao, Qi Liu, Linan Yue, Fangzhou Yao, Rui Lv, Zheng Zhang, Hao Wang, and Zhenya Huang. 2025. Agent4edu: Generating learner response data by generative agents for intelligent education systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 23923–23932.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. 2024. Routerbench: A benchmark for multi-llm routing system. *arXiv preprint arXiv:2403.12031*.

Maher Joe Khan Omar Jian. 2023. Personalized learning through ai. *Advances in Engineering Innovation*, 5:16–19.

Wittawat Jitkrittum, Harikrishna Narasimhan, Ankit Singh Rawat, Jeevesh Juneja, Congchao Wang, Zifeng Wang, Alec Go, Chen-Yu Lee, Pradeep Shenoy, Rina Panigrahy, and 1 others. 2025. Universal model routing for efficient llm inference. *arXiv preprint arXiv:2502.08773*.

Jacqueline Leighton and Mark Gierl. 2007. *Cognitive diagnostic assessment for education: Theory and applications*. Cambridge University Press.

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

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

Frederic M Lord. 2012. *Applications of item response theory to practical testing problems*. Routledge.

Xinyi Lu and Xu Wang. 2024. Generative students: Using llm-simulated student profiles to support question item evaluation. In *Proceedings of the Eleventh ACM Conference on Learning@ Scale*, pages 16–27.

Rui Lv, Qi Liu, Weibo Gao, Jiatong Li, Kai Zhang, and Shiwei Tong. 2025. Real: How can llms simulate the real teacher? retrieval-enhanced agent for adaptive learning. *Thinking*, 1:2.

Fanhang Man, Huandong Wang, Jianjie Fang, Zhaoyi Deng, Baining Zhao, Xinlei Chen, and Yong Li. 2025. Context-aware sentiment forecasting via LLM-based multi-perspective role-playing agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2687–2703, Vienna, Austria. Association for Computational Linguistics.

Amogh Mannekote, Adam Davies, Jina Kang, and Kristy Elizabeth Boyer. 2025. Can llms reliably simulate human learner actions? a simulation authoring framework for open-ended learning environments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 29044–29052.

Kai Mei, Wujiang Xu, Minghao Guo, Shuhang Lin, and Yongfeng Zhang. 2025. Omnirouter: Budget and performance controllable multi-llm routing. *ACM SIGKDD Explorations Newsletter*, 27(2):107–116.

Alexander Tobias Neumann, Yue Yin, Sulayman Sowe, Stefan Decker, and Matthias Jarke. 2024. An llm-driven chatbot in higher education for databases and information systems. *IEEE Transactions on Education*.

Pranoy Panda, Raghav Magazine, Chaitanya Devaguptapu, Sho Takemori, and Vishal Sharma. 2025. Adaptive llm routing under budget constraints. *arXiv preprint arXiv:2508.21141*.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Mark D Reckase. 2009. Historical background for multidimensional item response theory (mirt). In *Multidimensional item response theory*, pages 57–77. Springer.

Atikah Shemshack and Jonathan Michael Spector. 2020. A systematic literature review of personalized learning terms. *Smart learning environments*, 7(1):33.

Wei Song, Zhenya Huang, Cheng Cheng, Weibo Gao, Bihan Xu, GuanHao Zhao, Fei Wang, and Runze Wu. 2025. Irt-router: Effective and interpretable multi-llm routing via item response theory. *arXiv preprint arXiv:2506.01048*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Jonathan L Templin and Robert A Henson. 2006. Measurement of psychological disorders using cognitive diagnosis models. *Psychological methods*, 11(3):287.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

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.

Yu Xia, Fang Kong, Tong Yu, Liya Guo, Ryan A Rossi, Sungchul Kim, and Shuai Li. 2024. Which llm to play? convergence-aware online model selection with time-increasing bandits. In *Proceedings of the ACM Web Conference 2024*, pages 4059–4070.

Songlin Xu, Hao-Ning Wen, Hongyi Pan, Dallas Dominguez, Dongyin Hu, and Xinyu Zhang. 2025. Classroom simulacra: Building contextual student generative agents in online education for learning behavioral simulation. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–26.

Songlin Xu, Xinyu Zhang, and Lianhui Qin. 2024. Eduagent: Generative student agents in learning. *arXiv preprint arXiv:2404.07963*.

Yizhe Yang, Palakorn Achananuparp, Heyan Huang, Jing Jiang, Nicholas Gabriel Lim, Cameron Tan Shi Ern, Phey Ling Kit, Jenny Giam Xiuhui, John Pinto, and Ee-Peng Lim. 2025. Consistent client simulation for motivational interviewing-based counseling. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 20959–20998, Vienna, Austria. Association for Computational Linguistics.

Zheyuan Zhang, Daniel Zhang-Li, Jifan Yu, Linlu Gong, Jinchang Zhou, Zhanxin Hao, Jianxiao Jiang, Jie Cao, Huiqin Liu, Zhiyuan Liu, and 1 others. 2025. Simulating classroom education with llm-empowered agents. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10364–10379.

Zesen Zhao, Shuowei Jin, and Z Morley Mao. 2024. Eagle: Efficient training-free router for multi-llm inference. *arXiv preprint arXiv:2409.15518*.

Longwei Zheng, Fei Jiang, Xiaoqing Gu, Yuanyuan Li, Gong Wang, and Haomin Zhang. 2025. Teaching via llm-enhanced simulations: Authenticity and barriers to suspension of disbelief. *The Internet and Higher Education*, 65:100990.

A Student Agent Framework

In this appendix, we present two representative works that focus on the simulation of student behavior in prior studies. And then we provide a detailed description of the student simulation agent adopted in our framework.

The firsrt work is EduAgent (Xu et al., 2024). EduAgent adopts a modular design with two core space. **Memory Space**: it hierarchically stores physiological data (gaze trajectories, mouse operations), cognitive data (6 types of states like workload), and knowledge data (post-course test results), integrating student personas and course-related information. **Action Space**: Outputs gaze/motor behaviors mapped to Areas of Interest (AOIs), and personalized question-answering performance.

The second work is Agent4Edu (Gao et al., 2025). The LLM-powered generative agent in Agent4Edu integrates three specialized modules for personalized learning simulation: **Learner Profile Module**, initialized with real-world response data to capture explicit practice styles (e.g., activity, diversity) and implicit cognitive factors (e.g., problem-solving ability); **Memory Module**, designed based on human learning mechanisms to include factual memory (reinforced response records), short-term memory (recent practice details), and long-term memory (reinforced facts, LLM-generated summaries, and a

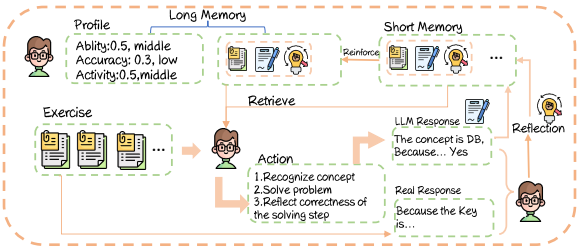

Figure 5: The Over View of Student Agent

forgetting curve) with retrieval, writing, and reflection capabilities; and **Action Module**, enabling human-like behaviors such as cognitive-driven exercise acceptance/rejection, exercise reading/understanding (with corrective reflection for mismatched knowledge concepts), and chain-of-thought analysis/solving (generating answers and correctness predictions, plus corrective reflection for inconsistencies).

Because our dataset only contains student–exercise interaction records and does not include fine-grained behavioral signals such as gaze trajectories or mouse operations, we adopt the second line of work as our student simulation framework.

As illustrated in Figure 5, the agent consists of three core modules: a memory module, a behavior module, and a reflection module. The agent simulates student learning by sequentially interacting with exercises in temporal order.

A.1 Memory Module

The memory module maintains both short-term memory and long-term memory.

Short-Term Memory. Short-term memory stores the student agent's most recent interaction records. Each memory entry is represented as $r = (e, ans, reflection)$, where e denotes the exercise, *ans* is the agent's response, and *reflection* records the agent's reflective analysis. Each exercise e includes both the exercise text and its associated knowledge concepts.

Short-term memory has a limited capacity. In our implementation, the maximum capacity is set to 5 recent interaction records. In addition, for each knowledge concept, we track its occurrence frequency within the short-term memory. If the occurrence count of a concept reaches a predefined threshold, the corresponding short-term memory entries are reinforced and transferred into long-term memory. We set the reinforcement threshold to 3 occurrences.

Long-Term Memory. Long-term memory stores more stable information about the student agent. It includes the student's cognitive profile (i.e., the knowledge concept proficiency vector), historical response accuracy, and response activity level. Response activity is defined as the ratio between the number of exercises attempted by the student and the total number of exercises. These attributes allow the agent to condition its behavior on its overall ability and engagement level. Long-term memory also contains reinforced memories transferred from short-term memory.

A.2 Action Module

The action module governs how the student agent answers an exercise. For a given exercise, the agent first identifies the knowledge concepts involved. Before answering, the agent retrieves all short-term memories as contextual input. It then searches long-term memory for records whose associated concepts overlap with those of the current exercise. If such records exist, they are collected as candidates, and the most recent three entries (in temporal order) are selected.

Based on the exercise content, extracted concepts, and retrieved memories, the agent generates a response to the exercise. After answering, the agent additionally judges whether it believes its own response is correct. The generated response and self-assessment together constitute the agent's answer *ans*.

A.3 Reflection Module

The reflection module is responsible for metacognitive correction. After the response is generated, the agent compares its self-assessment with the ground-truth outcome. If the agent correctly judges its response as correct, or correctly judges it as incorrect, no reflection is performed. Otherwise, the agent triggers a reflection process to analyze why the response or the self-assessment is incorrect.

The reflection result, together with the exercise and the generated response, is written into short-term memory as part of the interaction record. This mechanism allows the agent to adjust future behavior based on past inconsistencies.

A.4 Overall Workflow

The student agent simulates learning by interacting with exercises sequentially in chronological order. For each exercise, the agent first retrieves relevant memories, then generates a response through the

behavior module, performs reflection when necessary, and finally stores the resulting record in short-term memory. Through the interaction of memory, behavior, and reflection modules, the agent produces coherent and temporally grounded learning behavior simulation.

B Model Training Configuration for Cognitive Diagnosis

The NeuralCDM model was trained on an NVIDIA GeForce RTX 4090 GPU. We used a batch size of 32 and trained for 5 epochs with the Adam optimizer at a learning rate of 0.002. The prediction network consisted of two fully connected hidden layers with output dimensions of 512 and 256, respectively. Dropout regularization with a rate of 0.5 was applied to both fully connected layers.

C Student IDs Used in Experiments

Due to computational and cost constraints, we do not conduct LLM-based simulation on the full student population. Instead, we select a representative subset of students for evaluation.

Specifically, based on the cognitive diagnosis results obtained from NeuralCD, we stratify students into three ability groups: *low-ability*, *medium-ability*, and *high-ability*. From each group, we randomly sample 100 students, resulting in a total of 300 students used in the experiments.

To facilitate reproducibility and fair comparison in future work, we explicitly list the student identifiers corresponding to each ability group in this appendix. These identifiers uniquely determine the subset of students used for evaluation and allow exact reconstruction of the experimental setting.

Low-Ability Students. The student IDs corresponding to the low-ability group are listed as follows:

- {2, 5, 10, 11, 17, 19, 24, 30, 31, 39, 41, 43, 45, 60, 63, 66, 71, 72, 73, 74, 81, 86, 87, 90, 95, 98, 102, 105, 107, 108, 120, 122, 124, 126, 127, 131, 133, 135, 152, 156, 157, 172, 176, 185, 188, 198, 201, 205, 212, 218, 222, 223, 232, 236, 237, 241, 251, 255, 260, 265, 272, 275, 278, 279, 280, 281, 283, 287, 291, 293, 297, 299, 301, 302, 304, 307, 308, 311, 312, 313, 314, 315, 317, 323, 327, 329, 333, 334, 335, 336, 346, 347, 348, 350, 352, 354, 360, 369, 370, 375}

Medium-Ability Students. The student IDs corresponding to the medium-ability group are listed as follows:

- {12, 16, 21, 23, 25, 26, 29, 33, 35, 38, 40, 46, 49, 53, 54, 55, 59, 65, 67, 84, 92, 94, 99, 101, 111, 123, 132, 138, 140, 141, 144, 147, 154, 159, 161, 162, 163, 171, 177, 179, 180, 181, 183, 186, 187, 202, 206, 209, 210, 214, 216, 220, 221, 226, 228, 242, 245, 252, 253, 261, 262, 266, 267, 268, 274, 282, 288, 294, 295, 296, 303, 305, 309, 310, 316, 319, 322, 328, 331, 341, 342, 343, 359, 361, 363, 365, 366, 373, 374, 379, 380, 381, 383, 384, 388, 392, 394, 397, 399, 401}

High-Ability Students. The student IDs corresponding to the high-ability group are listed as follows:

- {1, 15, 20, 22, 37, 42, 50, 51, 56, 69, 75, 77, 79, 80, 93, 97, 100, 104, 109, 110, 115, 118, 119, 142, 150, 155, 164, 165, 169, 170, 173, 178, 190, 191, 197, 203, 233, 243, 246, 256, 259, 263, 269, 270, 271, 273, 276, 286, 289, 290, 298, 318, 321, 325, 338, 340, 344, 345, 349, 355, 364, 368, 371, 377, 390, 391, 398, 408, 426, 427, 428, 429, 430, 431, 432, 434, 438, 443, 444, 445, 447, 448, 451, 455, 458, 466, 468, 472, 479, 480, 481, 483, 486, 498, 500, 505, 506, 507, 508, 509}

In total, after the above selection process, we obtain 7,697 student–exercise interaction records, which are used in all subsequent experiments.

D LLMs Pool

We list all large language models included in the LLM pool used in our experiments, together with their corresponding access addresses. Each model is indexed for ease of reference and reproducibility.

1. **Qwen1.5-0.5B-Chat**: https://huggingface.co/Qwen/Qwen1.5-0.5B-Chat
2. **Qwen1.5-1.8B-Chat**: https://huggingface.co/Qwen/Qwen1.5-1.8B-Chat
3. **Qwen1.5-7B-Chat**: https://huggingface.co/Qwen/Qwen1.5-7B-Chat
4. **Qwen1.5-32B-Chat**: https://huggingface.co/Qwen/Qwen1.5-32B-Chat

5. **Qwen1.5-110B-Chat**: https://huggingface.co/Qwen/Qwen1.5-110B-Chat
6. **Qwen2-0.5B-Instruct**: https://huggingface.co/Qwen/Qwen2-0.5B-Instruct
7. **Qwen2-1.5B-Instruct**: https://huggingface.co/Qwen/Qwen2-1.5B-Instruct
8. **Qwen2-7B-Instruct**: https://huggingface.co/Qwen/Qwen2-7B-Instruct
9. **Qwen2-57B-A14B-Instruct**: https://huggingface.co/Qwen/Qwen2-57B-A14B-Instruct
10. **Qwen2-72B-Instruct**: https://huggingface.co/Qwen/Qwen2-72B-Instruct
11. **Qwen2.5-3B-Instruct**: https://huggingface.co/Qwen/Qwen2.5-3B-Instruct
12. **Qwen2.5-7B-Instruct**: https://huggingface.co/Qwen/Qwen2.5-7B-Instruct
13. **Qwen2.5-14B-Instruct**: https://huggingface.co/Qwen/Qwen2.5-14B-Instruct
14. **Qwen2.5-32B-Instruct**: https://huggingface.co/Qwen/Qwen2.5-32B-Instruct
15. **Qwen2.5-72B-Instruct**: https://huggingface.co/Qwen/Qwen2.5-72B-Instruct
16. **Qwen3-0.6B**: https://huggingface.co/Qwen/Qwen3-0.6B
17. **Qwen3-1.7B**: https://huggingface.co/Qwen/Qwen3-1.7B
18. **Qwen3-4B**: https://huggingface.co/Qwen/Qwen3-4B
19. **Qwen3-8B**: https://huggingface.co/Qwen/Qwen3-8B
20. **Qwen3-14B**: https://huggingface.co/Qwen/Qwen3-14B
21. **Qwen3-32B**: https://huggingface.co/Qwen/Qwen3-32B
22. **Qwen-Turbo (2025-07-15)**: https://bailian.console.aliyun.com
23. **Qwen-Plus (2025-09-11)**: https://bailian.console.aliyun.com
24. **Qwen-Max (2025-01-25)**: https://bailian.console.aliyun.com
25. **LLaMA-3.2-1B-Instruct**: https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct
26. **LLaMA-3.2-3B-Instruct**: https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct
27. **LLaMA-3.1-8B-Instruct**: https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct
28. **LLaMA-3.1-70B-Instruct**: https://huggingface.co/meta-llama/Llama-3.1-70B-Instruct
29. **GLM-4-9B-Chat**: https://huggingface.co/zai-org/glm-4-9b-chat
30. **GLM-4-32B-0414**: https://huggingface.co/zai-org/GLM-4-32B-0414
31. **GLM-4.5-Air**: https://huggingface.co/zai-org/GLM-4.5-Air
32. **GPT-3.5-Turbo (0125)**: https://platform.openai.com/docs/models/gpt-3.5-turbo
33. **GPT-4.1-Mini (2025-04-14)**: https://platform.openai.com/docs/models/gpt-4.1-mini
34. **GPT-5-Mini (2025-08-07)**: https://platform.openai.com/docs/models/gpt-5-mini
35. **DeepSeek-V3.2**: https://platform.deepseek.com

E Case Study

E.1 Case Study of Negative Bias Caused by Weak LLMs

We present a concrete example illustrating the negative bias introduced by a low-capacity LLM when simulating a student who answered correctly. The student id is 190, the exercise id is 181.

Exercise *If we wish to grant a privilege to a user and also allow the user to pass the privilege on to other users, which of the following should we append to the appropriate grant command?*
Options: (A) With grant privilege (B) Allow grant privilege (C) Allow grant option (D) With grant option **Correct Answer:** D
Knowledge Concepts: Discretionary access control (DAC), GRANT

Student Response The student answered correctly by selecting **(D) With grant option**.

Qwen2.5-7B Response (Weak Baseline) GRANT. The task requires understanding the SQL GRANT command and its limitations. The GRANT command cannot be used to grant privileges on a specified tuple directly. Therefore, the correct answer is (B) A specified tuple. Yes

Our Method (Ability-Aware Assignment) Response Discretionary access control (DAC), GRANT. Since the question involves granting privileges and passing those privileges on to other users, the correct answer should allow for the propagation of granted privileges. The most suitable option is (D) With grant option, as it enables the user to grant the same privilege to other users. Yes.

Analysis Although the student demonstrated correct understanding of the `GRANT` command with the *with grant option*, Qwen2.5-7B produced an incorrect response. This indicates that the intrinsic capacity of Qwen2.5-7B is insufficient to faithfully simulate students who have already mastered this concept.

In contrast, our ability-aware framework assigns an LLM whose diagnosed capability better aligns with the student's cognitive state. As a result, the simulated response correctly reflects the student's mastery of discretionary access control and privilege delegation.

E.2 Case Study of Positive Bias Caused by Strong LLMs

We present a representative example illustrating positive bias introduced by a high-capacity LLM, where the model overestimates a student's cognitive ability and produces a correct response even when the student answered incorrectly. The student id is 190, the exercise id is 177.

Exercise *Consider the following database schema:*

> Movie(title, production year, country, run time, major genre)
> primary key: {title, production year}
>
> Person(id, first name, last name, year born)
> primary key: {id}
>
> Director(id, title, production year)
> foreign keys: [title, production year] $\subseteq$ Movie[title, production year], [id] $\subseteq$ Person[id]
>
> Writer(id, title, production year, credits)
> foreign keys: [title, production year] $\subseteq$ Movie[title, production year], [id] $\subseteq$ Person[id]

Which of the following relational algebra expressions output the titles and production years of movies written by writers born after 1980?
(A): $\pi_{title,production\ year}\big(Movie \bowtie (Writer \bowtie \sigma_{year\ born>1980}(Person))\big)$
(B) $\pi_{title,production\ year}\big(Writer \bowtie \sigma_{year\ born>1980}(Person)\big)$
Options: (A) only A (B) only B (C) both A and B (D) neither A nor B
Correct Answer: C
Knowledge Concept: Relational Algebra

Student Response The student answered **incorrectly**.

GPT-5-mini Response (Strong Baseline) Relational Algebra. Join Writer with Person filtered by year born > 1980, then project title and production year. The Movie join in A is redundant because Writer has a foreign key to Movie, so both expressions yield the same result. Final answer: (C) both A and B. Yes.

Our Method (Ability-Aware Assignment) Response Relational Algebra. The solution involves

understanding how to join tables in relational algebra and apply selection conditions. Both alternatives involve joining the Writer table with the Person table to filter out writers born after 1980. However, alternative A also joins the result with the Movie table and projects the title and production year, which is necessary to get the required information. Alternative B only joins Writer with Person but does not project the movie details, making it incomplete for the required output. Therefore, the correct answer is A. Yes.

Analysis In this example, the student failed to correctly reason about equivalence in relational algebra and answered the question incorrectly. However, the high-capacity model GPT-5-mini produced the correct answer by leveraging its strong internal reasoning ability.

In contrast, our ability-aware framework assigns an LLM whose diagnosed capability better matches the student's cognitive state. As a result, the simulated response reproduces the student's incorrect reasoning pattern, providing a more faithful representation of the learner's actual mastery level.